

ARTIFICIAL GENERAL INTELLIGENCE 2008

Frontiers in Artificial Intelligence and Applications

FAIA covers all aspects of theoretical and applied artificial intelligence research in the form of monographs, doctoral dissertations, textbooks, handbooks and proceedings volumes. The FAIA series contains several sub-series, including “Information Modelling and Knowledge Bases” and “Knowledge-Based Intelligent Engineering Systems”. It also includes the biennial ECAI, the European Conference on Artificial Intelligence, proceedings volumes, and other ECCAI – the European Coordinating Committee on Artificial Intelligence – sponsored publications. An editorial panel of internationally well-known scholars is appointed to provide a high quality selection.

Series Editors:

J. Breuker, R. Dieng-Kuntz, N. Guarino, J.N. Kok, J. Liu, R. López de Mántaras,
R. Mizoguchi, M. Musen, S.K. Pal and N. Zhong

Volume 171

Recently published in this series

- Vol. 170. J.D. Velásquez and V. Palade, Adaptive Web Sites – A Knowledge Extraction from Web Data Approach
- Vol. 169. C. Branki et al. (Eds.), Techniques and Applications for Mobile Commerce – Proceedings of TAMoCo 2008
- Vol. 168. C. Riggelsen, Approximation Methods for Efficient Learning of Bayesian Networks
- Vol. 167. P. Buitelaar and P. Cimiano (Eds.), Ontology Learning and Population: Bridging the Gap between Text and Knowledge
- Vol. 166. H. Jaakkola, Y. Kiyoki and T. Tokuda (Eds.), Information Modelling and Knowledge Bases XIX
- Vol. 165. A.R. Lodder and L. Mommers (Eds.), Legal Knowledge and Information Systems – JURIX 2007: The Twentieth Annual Conference
- Vol. 164. J.C. Augusto and D. Shapiro (Eds.), Advances in Ambient Intelligence
- Vol. 163. C. Angulo and L. Godo (Eds.), Artificial Intelligence Research and Development
- Vol. 162. T. Hirashima et al. (Eds.), Supporting Learning Flow Through Integrative Technologies
- Vol. 161. H. Fujita and D. Pisanelli (Eds.), New Trends in Software Methodologies, Tools and Techniques – Proceedings of the sixth SoMeT_07
- Vol. 160. I. Maglogiannis et al. (Eds.), Emerging Artificial Intelligence Applications in Computer Engineering – Real World AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies
- Vol. 159. E. Tyugu, Algorithms and Architectures of Artificial Intelligence
- Vol. 158. R. Luckin et al. (Eds.), Artificial Intelligence in Education – Building Technology Rich Learning Contexts That Work

Artificial General Intelligence 2008

Proceedings of the First AGI Conference

Edited by

Pei Wang

Temple University

Ben Goertzel

Novamente LLC

and

Stan Franklin

University of Memphis

IOS
Press

Amsterdam • Berlin • Oxford • Tokyo • Washington, DC

© 2008 The authors and IOS Press.

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, without prior written permission from the publisher.

ISBN 978-1-58603-833-5

Library of Congress Control Number: 2008900954

Publisher

IOS Press

Nieuwe Hemweg 6B

1013 BG Amsterdam

Netherlands

fax: +31 20 687 0019

e-mail: order@iospress.nl

Distributor in the UK and Ireland

Gazelle Books Services Ltd.

White Cross Mills

Hightown

Lancaster LA1 4XS

United Kingdom

fax: +44 1524 63232

e-mail: sales@gazellebooks.co.uk

Distributor in the USA and Canada

IOS Press, Inc.

4502 Rachael Manor Drive

Fairfax, VA 22032

USA

fax: +1 703 323 3668

e-mail: iosbooks@iospress.com

LEGAL NOTICE

The publisher is not responsible for the use which might be made of the following information.

PRINTED IN THE NETHERLANDS

Preface

Pei WANG, Ben GOERTZEL and Stan FRANKLIN

The field of Artificial Intelligence (AI) was initially directly aimed at the construction of “thinking machines” – that is, computer systems with human-like general intelligence. But this task proved more difficult than expected. As the years passed, AI researchers gradually shifted focus to producing AI systems that intelligently approached specific tasks in relatively narrow domains.

In recent years, however, more and more AI researchers have recognized the necessity – and the feasibility – of returning to the original goal of the field. Increasingly, there is a call to focus less on highly specialized “narrow AI” problem solving systems, and more on confronting the difficult issues involved in creating “human-level intelligence,” and ultimately general intelligence that goes beyond the human level in various ways. “Artificial General Intelligence (AGI)”, as this renewed focus has come to be called, attempts to study and reproduce intelligence as a whole in a domain-independent way.

Encouraged by the recent success of several smaller-scale AGI-related meetings and special tracks at conferences, we took the initiative to organize the very first international conference on AGI. Our goal in doing so was to give researchers in the field an opportunity to present relevant research results and to exchange ideas on topics of common interest.

The response to AGI-08 was even stronger than we expected. We received many interesting papers addressing a wide range of AGI-relevant topics and exploring various kinds of theoretical and technical ideas. Given the complexity and difficulty of the problems the field is facing, we believe it is crucial to encourage the exchange of ideas and opinions with various degrees of maturity, ranging from position papers to descriptions of mature mathematical and cognitive AGI theories, and practical work with implemented AGI architectures.

In this collection, the AGI-08 papers are organized into three groups. The conference papers are divided into full-length papers (12 pages, with a few exceptions) and short position statements (5 pages). Also included are the papers presented in the post-conference workshop on the sociocultural, ethical and futurological implications of AGI.

We believe meetings like AGI-08 are important, not only because of their presentations and discussions, but also because of the potential they have to help catalyze the self-organization of a vital AGI research community. Together, we will continue to directly challenge one of the most difficult and essential problems in human history, the creation of human-level artificial general intelligence.

We thank the following referees for devoting their valuable time to reviewing the submitted papers: Sam Adams, James Anderson, Mike Anderson, Eric Baum, Mark Bickhard, Henry Brighton, Nick Cassimatis, Hernan Castro, L. Andrew Coward, Hugo de Garis, Włodzisław Duch, Richard Duro, David Friedlander, J. Storrs Hall, David

Hart, Marcus Hutter, Cliff Joslyn, Randal Koene, Moshe Looks, Bruce MacLennan, Don Perlis, Matthias Scheutz, Juergen Schmidhuber, Lokendra Shastri, Boris Velichkovsky, Karin Verspoor, Paul Vogt, and Mark Waser.

We would also like to thank the other members of the AGI-08 Organizing Committee, Sidney D'Mello, Bruce Klein, and Lee McCauley, for the thought and effort they expended in preparing the conference; and Bruce Klein and Natasha Vita-More for their help with the post-conference workshop which resulted in a number of the papers contributed to this volume.

Contents

Preface <i>Pei Wang, Ben Goertzel and Stan Franklin</i>	v
Full-Length Papers	
Participating in Cognition: The Interactive Search Optimization Algorithm <i>Nadav Abkasis, Israel Gottlieb and Eliraz Itzhaki</i>	3
Input Feedback Networks: Classification and Inference Based on Network Structure <i>Tsvi Achler and Eyal Amir</i>	15
Reasoning with Prioritized Data by Aggregation of Distance Functions <i>Ofer Arieli</i>	27
Distance-Based Non-Deterministic Semantics <i>Ofer Arieli and Anna Zamansky</i>	39
Governing Lethal Behavior: Embedding Ethics in a Hybrid Deliberative/Reactive Robot Architecture. PART 2: Formalization for Ethical Control <i>Ronald C. Arkin</i>	51
Seven Principles of Synthetic Intelligence <i>Joscha Bach</i>	63
Language Processing in Human Brain <i>Alexander Borzenko</i>	75
Toward Logic-Based Cognitively Robust Synthetic Characters in Digital Environments <i>Selmer Bringsjord, Andrew Shilliday, Joshua Taylor, Dan Werner, Micah Clark, Ed Charpentier and Alexander Bringsjord</i>	87
A Cognitive Substrate for Natural Language Understanding <i>Nicholas L. Cassimatis, Arthi Murugesan and Magdalena D. Bugajska</i>	99
The China-Brain Project: Building China's Artificial Brain Using an Evolved Neural Net Module Approach <i>Hugo de Garis, Jian Yu Tang, Zhiyong Huang, Lu Bai, Cong Chen, Shuo Chen, Junfei Guo, Xianjin Tan, Hao Tian, Xiaohan Tian, Xianjian Wu, Ye Xiong, Xiangqian Yu and Di Huang</i>	107
Cognitive Architectures: Where Do We Go from Here? <i>Włodzisław Duch, Richard J. Oentaryo and Michel Pasquier</i>	122
LIDA and a Theory of Mind <i>David Friedlander and Stan Franklin</i>	137

How Might Probabilistic Reasoning Emerge from the Brain? <i>Ben Goertzel and Cassio Pennachin</i>	149
An Integrative Methodology for Teaching Embodied Non-Linguistic Agents, Applied to Virtual Animals in Second Life <i>Ben Goertzel, Cassio Pennachin, Nil Geissweiller, Moshe Looks, Andre Senna, Welter Silva, Ari Heljakka and Carlos Lopes</i>	161
VARIAC: An Autogenous Cognitive Architecture <i>J. Storrs Hall</i>	176
Probabilistic Quantifier Logic for General Intelligence: An Indefinite Probabilities Approach <i>Matthew Iklé and Ben Goertzel</i>	188
Comirit: Commonsense Reasoning by Integrating Simulation and Logic <i>Benjamin Johnston and Mary-Anne Williams</i>	200
Learning from Inconsistencies in an Integrated Cognitive Architecture <i>Kai-Uwe Kühnberger, Peter Geibel, Helmar Gust, Ulf Krumnack, Ekaterina Ovchinnikova, Angela Schwering and Tonio Wandmacher</i>	212
Extending the Soar Cognitive Architecture <i>John E. Laird</i>	224
Temporal Action Logic for Question Answering in an Adventure Game <i>Martin Magnusson and Patrick Doherty</i>	236
Artificial General Intelligence Through Large-Scale, Multimodal Bayesian Learning <i>Brian Milch</i>	248
A Computational Approximation to the AIXI Model <i>Sergey Pankov</i>	256
Essential Phenomena of General Intelligence <i>Marc Pickett, Don Miner and Tim Oates</i>	268
OSCAR: An Architecture for Generally Intelligent Agents <i>John L. Pollock</i>	275
Anticipative Coordinated Cognitive Processes for Interactivist and Piagetian Theories <i>Jean-Charles Quinton, Jean-Christophe Buisson and Filipo Perotto</i>	287
Hybrid Reasoning and the Future of Iconic Representations <i>Catherine Recanati</i>	299
Cognitive Constructor: An Intelligent Tutoring System Based on a Biologically Inspired Cognitive Architecture (BICA) <i>Alexei V. Samsonovich, Kenneth A. de Jong, Anastasia Kitsantas, Erin E. Peters, Nada Dabbagh and M. Layne Kalbfleisch</i>	311
Transfer Learning and Intelligence: An Argument and Approach <i>Matthew E. Taylor, Gregory Kuhlmann and Peter Stone</i>	326

Real Time Machine Deduction and AGI <i>Peter G. Tripodes</i>	338
Text Disambiguation by Educable AI System <i>Alexander Voskresenskij</i>	350
What Do You Mean by “AI”? <i>Pei Wang</i>	362
Using Decision Trees to Model an Emotional Attention Mechanism <i>Saman Harati Zadeh, Saeed Bagheri Shouraki and Ramin Halavati</i>	374

Position Statements

Fusing Animals and Humans <i>Jonathan Connell</i>	389
Four Paths to AI <i>Jonathan Connell and Kenneth Livingston</i>	394
Adversarial Sequence Prediction <i>Bill Hibbard</i>	399
Artificial General Intelligence via Finite Covering with Learning <i>Yong K. Hwang, Samuel B. Hwang and David B. Hwang</i>	404
Cognitive Primitives for Automated Learning <i>Sudharsan Iyengar</i>	409
Vector Symbolic Architectures: A New Building Material for Artificial General Intelligence <i>Simon D. Levy and Ross Gayler</i>	414
Analogy as Integrating Framework for Human-Level Reasoning <i>Angela Schwering, Ulf Krümmack, Kai-Uwe Kühnberger and Helmar Gust</i>	419
Designing Knowledge Based Systems as Complex Adaptive Systems <i>Karan Sharma</i>	424
Artificial General Intelligence: An Organism and Level Based Position Statement <i>Leslie S. Smith</i>	429

Workshop Papers

The Artilect War: Cosmists vs. Terrans. A Bitter Controversy Concerning Whether Humanity Should Build Godlike Massively Intelligent Machines <i>Hugo de Garis</i>	437
Stages of Ethical Development in Artificial General Intelligence Systems <i>Ben Goertzel and Stephan Vladimir Bugaj</i>	448
Engineering Utopia <i>J. Storrs Hall</i>	460

OpenCog: A Software Framework for Integrative Artificial General Intelligence <i>David Hart and Ben Goertzel</i>	468
Open Source AI <i>Bill Hibbard</i>	473
On the Broad Implications of Reinforcement Learning Based AGI <i>Scott Livingston, Jamie Garvey and Itamar Elhanany</i>	478
The Basic AI Drives <i>Stephen M. Omohundro</i>	483
A Scientific Perspective on the Hard Problem of Consciousness <i>Alexei V. Samsonovich, Giorgio A. Ascoli, Harold Morowitz and M. Layne Kalbfleisch</i>	493
Author Index	507

Full-Length Papers

This page intentionally left blank

Participating in Cognition: The Interactive Search Optimization Algorithm

Nadav Abkasis, Israel Gottlieb and Eliraz Itzhaki

Department of Computer Science, Jerusalem College of Technology, Israel

Abstract. We consider the Distributed Cognition paradigm as a framework for implementing artificial components of human cognition. We take email/internet search as a setting of distributed cognition and describe an algorithm that intervenes and enhances a component of this distributed process. Results are presented demonstrating the effectiveness of the algorithm in interaction with an automaton simulating human search activity. We define a notion of *synchronization* with a non-random, non-Turing computation and argue that the given algorithm exhibits such synchronization behavior. Our results suggest that the framework may be useful for studying a class of non-Turing computation that is central to General Intelligence.

Keywords. Distributed Cognition, Embodiment, Search Optimization, Non-Turing Computation.

1. Introduction

In Artificial General Intelligence one seeks the "General", i.e. the ability to come up with the right approach to a new situation, as opposed to merely applying an algorithm to a problem setting for which it has been designed. The phrase "applying an algorithm to a problem for which it has been designed" is a fair description of an arbitrary but specific Turing machine. A Universal Turing machine relaxes the condition that the algorithm be a fixed one, but leaves intact the requirement that the algorithm executed must be appropriate for the problem to be solved. It is therefore natural to investigate the possibility, and perhaps even the necessity of non-Turing capability for AGI, or what has been called "hypercomputing". Most work on hypercomputing has been of a theoretical sort, i.e. investigation of machine structures whose physical realizability remains a largely open question [1]. In this work we are concerned with a form of hypercomputing that is directly observable, and offers a practical context for experimentation as well.

A researcher confronted with a problem will often have practical experience that suggests an "intuition" about the correct approach to a solution. The challenge in such situations is typically expressed by the phrase, "can you formalize this intuition?" By this is meant can one translate the intuition into explicit rules, hence making them more readily subject to analysis, optimization and automated implementation.

In cognitive science a key issue is the process of intuition itself; here the challenge is modified as – "can we formalize the *algorithm* of intuition?". In both contexts we can observe an attempt to bridge the no-man's-land between the obscurity of intuition

and the clarity of an algorithm. In the cognitive science case however, the attempt has historically been problematic. If we take "algorithm" as synonymous with "Turing machine" and at the same time allow that "intuition" is non Turing-computable – as Turing himself believed [2], then "algorithm" and "intuition" are mutually exclusive and clearly we are asking the wrong question. Or perhaps our notion of algorithm needs to be extended.

In this paper we approach the task of understanding the process of intuition via a "learn by doing" method, i.e. explicit *participation* in the algorithm of intuition. We will argue that such explicit participation amounts to a broader notion of what is meant by an algorithm than the accepted sense of the term, and one that may be more suited to the goals of AGI.

How can we participate in a process that appears to us as a black box locked away in the human brain? Here, the notion of *distributed cognition* [3] is useful. Distributed cognition posits that cognitive processes should be viewed as including elements outside of the physical individual. Hence we may intervene or participate in this process by replacing one of its external components with an alternate version.

In what follows, we first briefly review the relevant tenets of distributed cognition. We then tackle the question of how to render explicit any part of a process which is by definition not well-defined, i.e. intuition. Our answer will be constructive, i.e. we consider a common cognitive process in which intuition plays an obvious role, posit a distributed structure as the setting for this process and then build an enhanced version of one part of this structure.

In this work we do not as yet draw conclusions regarding intuition nor offer a model for this or any other cognitive process. Our purpose is to suggest active participation as a technique for learning about such processes and associated possible models. Moreover, our emphasis is on the construction of working systems that produce results and can be used to learn by experiment. Accordingly, most of this paper describes a working system. Finally, we shall give an informal argument that our methodology exhibits a form of directly observable non-Turing computation.

1.1. Distributed Cognition

Distributed cognition is described in [3] as positing distribution of the process of cognition over three dimensions,

1. The social dimension, i.e. over groups rather than single individuals,
2. Over the material artifacts in the environment wherein the cognition is operative, and
3. Over the time dimension. That is, historical events influence later ones via a process that is itself integral to the cognition. The reverse may also be true, viz. the influence of history is itself influenced by the present in that it is interpreted differently according to current experience.

An additional tenet of distributed cognition is that of *embodiment*. This amounts to a rejection of the view that cognition is a process consisting of interactions between purely abstract representations, from which it follows that the underlying substrate that carries these representations is of no consequence to cognitive behavior. Rather, it holds that such substrate directly influences the cognitive process and dictates the types of interactional structures that may emerge.

1.2. Search Application

Consider an email search task that often confronts a user at a large organization. They receive a large volume of mail, so large that they do not manage to file it all in neat separate mail folders. A common situation is looking for a piece of mail received some time back, but unfortunately, our user cannot remember how long ago – was it 2 weeks or 2 months? Neither do they remember the exact name on the "From: xxx" header, although they likely have a vague recollection of one or a few letters that it contains. They may or may not have a vague recollection of some part of the Subject line. Our user keeps mail for many months, possibly longer, knowing that these situations arise. However, finding exactly what one is looking for is more difficult.

We start with a user that has entered an initial set of keywords, e.g. Sender and Subject, based on whatever approximating string of letters they can muster from memory. The search engine produces a list of candidate emails in response and based on these, the user modifies the keywords; this process may repeat some (small) number of times. The specific modification to keywords employed by a user – in response to the material offered by the search engine – typically has no fixed recipe, and may well change substantially from one iteration to the next. Clearly, intuition is being relied upon heavily to modify keywords so as to promote the user's objective.

In this setting we can directly identify all 3 of the dimensions indicated by the distributed approach. Thus,

1. In addition to the user's brain, the search engine is a participating artifact.
2. The time scale of the process extends over successive iterations of keyword entry, examination of results and keyword modification.
3. Search is directly influenced by the qualitative content of material returned by the engine. This material in turn is the direct contribution of others in the email community of which the user is a member. The principle of embodiment allows the influence of this material on the structure of the cognitive process.

Our goal is to participate in this cognitive process, and in so doing – enhance it. The chief difficulty is that the user does not know exactly what they are looking for – else they would enter the best keywords directly. Thus standard optimization algorithms are not applicable since we don't have a definition of what to optimize. However, we do know that our user eventually finds what they are looking for in most cases, hence we can posit an approximate monotonicity of progress versus time. That is, although there may be missteps and setbacks, on average there are significantly more steps forward than backward. We can therefore undertake a supportive role comprising two elements, as follows:

1. Remind the user of how they have progressed in similar situations in the past, with the goal of avoiding backwards steps, and
2. Abbreviate multiple steps into a single one, in cases where the user has found the sequence repeatedly useful in the past.

We still have a significant challenge to overcome, viz. how can one relate "situations in the past" and situations "found repeatedly useful" in the past – to the present context, when the notion of "useful" has no representation or definition that is accessible to us? Our approach is to rely on embodiment, viz. what is going on in the cognitive process must be parallel in some sense to the artifacts in the environment to which the process is being applied, whence it follows that we can track the cognition by simulation with those artifacts. Specifically, in the current application the artifacts are keywords. If we find that in previous iterations the user has tried to pursue pro-

toward the goal by modifying keyword a to keyword a' , i.e. the user has made an ordered association between two versions of a keyword, then we can conclude that this association is not the result of an intuition totally abstracted from the reality of words, word-parts or speech sounds etc. but rather that cognition is implicitly relying on a distance metric on the space of words itself. If we can find a metric for which the search iterations cluster into well defined repetitive associations then we have found a structure that parallels the cognition.

In the next section we make these ideas more precise via formal definition as an algorithm, which we shall call the Interactive Search Optimization (ISO) algorithm. We note that the above description of an email search process applies, essentially unchanged, to a search of the internet. We chose the email setting for our application because it was simpler to implement. For the internet case, a more elaborate and specialized characterization as a distributed cognitive process has been suggested in [4]. However, the three-dimensional characterization given above is adequate for our development here, and applies equally well to both problem settings.

2. Formal Development

We consider keywords from the 26 letter English alphabet only. The number of letters in our keywords is not restricted – other than by computational resources, and one can arbitrarily assign e.g. the first five letters to the "From:" key, the next five to the "Subject:" key, etc. Without loss of generality therefore, we consider in the sequel a single keyword of arbitrary length m .

Thus we have m dimensions of variation with 26 possible values each. Each of these variables is mapped to an angle θ as depicted in Figure 1.

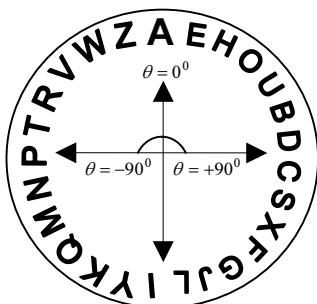


Figure 1 – Base Ordering and Initial Position

The ordering around the circle follows typical associations between the sounds of letters – as they are actually experienced by humans, viz. (a e h o u), (b d), (c s x), (g j), (i y), (k q), (m n), (p t) and (v w) are the groupings used. The remaining letters are singletons. The groups and singletons in turn are ordered approximately as they appear in the alphabet. This is called the *base ordering*. In the *initial position*, the angle $\theta = 0^\circ$ points to the letter 'A'.

Definition 1: *User Response and Computation of Action Vector*

A *delta search vector*, is an ordered sequence of angles $(\theta_1, \theta_2, \dots, \theta_m)$, $0^0 \geq \theta_j \geq -180^0$ or $0^0 < \theta_j \leq +180^0$; each θ_j represents a (possibly zero) rotation away from the current position. The user submits a delta vector to modify the keyword represented by the current position. The intention of this arrangement is that, in addition to recording the user's revision of a keyword, we also capture a notion of how the user judges the progress of the search. Specifically, the zero direction indicates "no change required", whence user responses θ_j that are close to zero will indicate generally that we are in the right direction, and conversely – the farther away from zero (or the closer to ± 180 degrees), the greater the user dissatisfaction with the progress of the search. The delta vector is provided by the user in the *response matrix* $\begin{bmatrix} \theta_1, \theta_2, \dots, \theta_m \\ w_1, w_2, \dots, w_m \end{bmatrix}$, along with a vector of corresponding weights w_j , $0 < w_j \leq 1$, the latter representing the user's indication of their own confidence in the former.

Instead of applying the delta in the user response directly – to obtain the next keyword, the ISO algorithm applies transformations to the response matrix based on what it can learn from the history of user responses thus far. The transformed version is called the *action matrix* because its delta vector is the one actually input to the search engine. Thus we have a sequence of pairs $(A_1, R_1), (A_2, R_2), \dots$ etc., where the index indicates the particular iteration of results/user response, A_i is the action matrix whose delta vector is input to the search engine and R_i is the corresponding user response matrix. The role of the weights w_j and the transformations that generate the A matrices will be defined in the sequel.

Definition 2: *Unification*

Write δ_j to denote one of the values θ_j or zero. Upper case X denotes any delta matrix. The unification mapping $U(X_k, X_{k+1}, \dots, X_{k+l}) \rightarrow X'$ may be expressed as,

$$X' = \begin{bmatrix} \theta'_1, \theta'_2, \dots, \theta'_m \\ w'_1, w'_2, \dots, w'_m \end{bmatrix}, \quad \begin{cases} P(\theta'_j = \sum_{h=k}^{k+l} [\delta_j]_h) = \prod [p_j]_h \\ w'_j = \prod [p_j]_h \text{ if } \theta'_j = \sum_{h=k+1}^{k+l} [\delta_j]_h \end{cases}$$

where

$$[p_j]_h = \begin{cases} [w_j]_h & \text{if } [\delta_j]_h = [\theta_j]_h \\ 1 - w_j]_h & \text{if } [\delta_j]_h = 0 \end{cases},$$

$P(\theta = x)$ denotes the probability that the random variable θ takes the value x and this probability is specified for all possible values of the m placeholders δ_j . That the distribution for θ'_j is a valid probability mass function (PMF) may be readily verified by noting that the distribution of each θ'_j is identical with the joint PMF of

independent identically distributed binary random variables $[\delta]_h$, $0 \leq h \leq l$ with probabilities $P([\delta]_h = True) = [w]_h$ and $P([\delta]_h = False) = 1 - [w]_h$.

We need the following functions, defined on any delta matrix X .

Definition 3: User Grade

1. The *User Grade* (URG):

$$urg(\theta) = \begin{cases} +1, & \text{if } -45^0 \leq \theta \leq +45^0 \\ 0, & \text{if } -135^0 \leq \theta \leq -45^0 \text{ or } +45^0 \leq \theta \leq +135^0 \\ -1, & \text{if } -180^0 \leq \theta \leq -135^0 \text{ or } +135^0 \leq \theta \leq +180^0 \end{cases}$$

2. The normalized, weighted, scalar user response θ^S :

$$\theta^S = \frac{1}{\sum_{j=1}^m w_j} \sum_{j=1}^m E(\theta_j), \quad \sum_{j=1}^m w_j \neq 0,$$

where $E(\theta_j)$ is the expectation of the random variable θ_j .

θ^S summarizes the information in a delta matrix with a single number.

3. The normalized weighted User Grade $urg(\theta^S)$. In the sequel, URG refers to the normalized weighted version, unless otherwise indicated.
4. The *Dimensional Progress Index* (DPI). This is a relative measure of cumulative progress toward the goal – as perceived by the user. It is defined recursively as follows:

$$dpi_i = \begin{cases} dpi_{i-1} + urg_i, & i > 0 \\ 0, & i = 0 \end{cases}$$

We think of the search space as comprising dimensions that are successively constrained or fixed so that the balance of the space to be searched gets progressively smaller. In this conceptualization, progress toward the goal may be measured by the number of dimensions that have been successfully fixed – hence the term Dimensional Progress Index.

The pair $[urg_i, dpi_i]$ is our overall user grade for iteration i . We must also specify if the grade function is taken on the response matrix R_i or on the action matrix A_i .

Definition 4: Generations

The delta matrices A_i , R_i and their associated user grades $[urg, dpi]$ comprise the i th *actual generation*. The accumulated sequence of actual generations is called the *history*. An actual generation is always a single (A_i, R_i) pair. Actual generations are distinguished from *virtual* generations (defined next), which are delta matrices the ISO algorithm will compute as candidates for input to the search engine, and which may or may not become actual. A virtual generation may have more than one element.

At each iteration i , the algorithm will compute a virtual next generation of candidate matrices A_{i+1} by applying unification to subsequences of actual generations

in the history, in accordance with the unification rules below. The virtual generation is computed from scratch at each iteration; there is only one such generation at any time.

Definition 5: Unification Rules

The symbol ' \cong ' denotes fuzzy equality, useful for e.g. comparison between DPI values which are in any case user subjective. We use standard fuzzy set membership based on a parabolic function [5].

In what follows, i is the current iteration number, while k is an arbitrary iteration number in the history. A superscript distinguishes functions associated with R matrix from those associated with the A matrix.

1. Unify all successive pairs of action matrices found in the history $(\dots A_k, R_k, A_{k+1} \dots)$ such that

$$\text{urg}_k^a = \text{urg}_i^r \text{ and } \text{dpi}_k^a \cong \text{dpi}_i^r.$$

2. Unify all subsequences of action matrices found in the history $(\dots A_k, R_k, A_{k+1}, \dots A_{k+l}, \dots)$ such that

$$\text{urg}_{i-1}^r = +1, \quad \text{urg}_i^r = -1,$$

$$\text{urg}_{k-1}^a = +1, \quad \text{urg}_{k+h}^a \neq +1, \forall h \in [0, l], \quad \text{urg}_{k+l+1}^a = +1 \text{ and } \text{dpi}_k^a \cong \text{dpi}_i^r.$$

Note that the functions urg and dpi for the current iteration associate with the response matrix R_i while the those from the history associate with the action matrix A_k . In general, we look for precedent in the action matrices of the history rather than the response matrices, since the former correspond what actually took place. In contrast, for the current iteration, we are in the process of computing the action matrix so the response matrix R_i is the most up-to-date indication of progress. The second rule is intended to capture a sequence of miss-steps, i.e. skip over all the "wrong way" attempts a user might make while chasing a dead end. Note that the cases are not mutually exclusive, i.e. where $\text{urg}_k^a = \text{urg}_i^r = -1$ both unifications may apply.

A delta matrix computed from unification will be called a *synthetic* action, i.e. we are taking the liberty of transforming the user's response based on the history. We apply a ranking of the synthetic actions in the virtual generation. The winner, if there is one, will be used instead of the current user response to form the next actual generation.

We compute a Consistency Index on the elements in the virtual generation. Denote by A' a synthetic action for which the index is to be computed, and let \mathcal{A} be the set of all action matrices accumulated thus far, synthetic or not, i.e. the union of the virtual generation and all actual generations. Consistency means the degree to which the action of A' is confirmed by other action deltas, and it is computed for A' separately relative to each $A \in \mathcal{A}$. Consistency has two criteria. The first checks that the action $A \in \mathcal{A}$ was applied with DPI similar (fuzzy equality) to the current value¹. The second looks for similarity of the user grade θ^S . Denote by $\text{eq} : \mathbb{R} \times \mathbb{R} \mapsto [0, 1]$ a fuzzy equality function on pairs of real numbers, i.e. with image in the continuous interval $[0, 1]$. We first compute a Local Consistency Index (LCI) for A' relative to a particular $A \in \mathcal{A}$,

¹ Unifications may be nested to arbitrary depth, hence this requires that we maintain, for any synthetic action matrix, a record of the DPI that held at the first iteration of the first level of unification.

$$\text{LCI}(A') = \text{Min} (\text{eq}(\theta^S(A'), \theta^S(A)), \text{eq}(dpi(A), dpi_i)), \quad A \in \mathcal{A},$$

where dpi_i is the DPI associated with the current user response. We then apply *concentration* and *fuzzification*, standard techniques from fuzzy logic [11], based on the actual *urg* which followed the action A . If that $urg \neq -1$, we take the square root of the LCI, otherwise we square it². The former magnifies the LCI; the latter reduces it.

Finally, we compute the Global Consistency Index (GCI) for each synthetic A' in the virtual generation,

$$\text{GCI}(A') = \sum_{A \in \mathcal{A}} \text{LCI}(A', A).$$

There are two cases in which the above procedure fails to produce a candidate for the action matrix A_{i+1} : where no sequence in the history qualifies for unification, or where the GCI is zero for all unifications computed. These will occur particularly in the early iterations when the history is short (or non-existent). In either of these situations we say that *synthesis fails*. We can now define the transformation that computes the action matrix A_{i+1} at each iteration. Denoting by VG the virtual generation, we have,

$$A_{i+1} = \begin{cases} U(R_i) & \text{if synthesis fails} \\ \arg \max_{A \in VG} \text{GCI}(A) & \text{otherwise} \end{cases}.$$

Unification of the user response matrix with itself – indicated for the case of synthesis failure, is just a non-deterministic rendering of the user's response, viz. each θ_j is applied with probability w_j ; with probability $1-w_j$ we apply a zero delta. The algorithm iterates until the user indicates the target has been found (or gives up).

3. Experimental Results

In this section we present our experience with the use of the ISO algorithm described in the previous section³. Our first task was to verify that the algorithm performs its intended function. That is, allowing our hypothesis from Section 1.2 that intuition makes use of a distance metric on keywords, does the algorithm in fact improve the efficiency of search? We constructed automata that simulate a user's search activity under different behavioral assumptions. We then compared the number of steps required by an automaton to reach the search objective to the number of steps required when that automaton is supported by the ISO algorithm. The results show that, for a significant range of behavioral parameters, and in particular such as would be expected in realistic human contexts, the algorithm significantly reduces the number of search steps. Of course the situation of a human user is fundamentally different in that they do not know the target in advance. Hence these experiments cannot directly verify the key "Embodiment" thesis, viz. that an ISO-like process actually tracks cognition. Experiment with humans remains for future work.

² Applicable only to LCI relative to actual matrices A in the history, not virtual ones.

³ In practice, some minor additional tuning and/or approximation to the algorithm was used. They are not detailed due to lack of space.

3.1. Types of Automata

In each of the automata described, the software knows the target keyword, and advances toward it, away from it or just "gets lost" – in accordance with a specific behavioral policy that is intended to approximate a human user's behavior. A parameter α introduces "errors" into the search behavior; with probability α , the automaton presents a user response (i.e. a delta angle θ) in accordance with its particular behavioral policy, and with probability $1 - \alpha$ it sets the delta angle θ at random. We constructed 4 different automata, as follows:

- A 1: Each response delta advances 1/4 of the distance to the target, relative to the base ordering. The random errors introduced with probability α are not given any special treatment, and enter the history as any other user response.
- A 2: Same as Automaton 1, however, we simulate also the likely intervention of a human user after a move in the wrong direction. After a random delta is executed the automaton retracts its move, i.e. presents an inverse delta, that returns the search to the same position it was before the random move.
- A 3: Same as Automaton 1, except as follows. We maintain a parallel record of the progress that would have been achieved by the unaided automaton. If at any point the unaided automaton gets ahead of the ISO algorithm, it "notices" that the system is not responding effectively and repeats its last response, i.e. the one which brought the unaided search into the lead.
- A 4: The response deltas produced by the automaton alternately advance toward, and retreat away from the target keyword, in accordance with the following sequence.

$$\frac{1}{n} - \frac{1}{n+2} + \frac{1}{n+1} - \frac{1}{n+4} + \frac{1}{n+3} - \frac{1}{n+6}, \dots, \text{etc.}$$

where n is a small integer such as 2 or 4, and each term represents a fraction of the current distance to the target. This series is dominated by $\sum 2/n^2$ and hence converges. As long as the limit of the sum is > 1 , one necessarily arrives at the target after a sufficient number of steps. More interesting in our context is that this behavior simulates "two steps forward and one step back".

Parameters for all the automata were varied over a range of values. In addition to the value of α , we varied the confidence weights associated with each delta in two different ways. In the simplest approach, the same confidence level was fixed for all user (i.e. automaton) responses throughout the search. For a more realistic scenario, we applied one level of confidence to those responses which followed the behavioral policy of the automaton and a different confidence value for the "error" responses, i.e. those generated randomly.

Figure 2 gives representative results for two of the automata; results for the other two are similar. In **Figure 3** we show the results as a function of varying confidence, and of varying differential between confidence for useful Responses (labeled "Base Confidence") as opposed to confidence for "error" responses. All data represent an average of 100 searches repeated with the same target keyword.

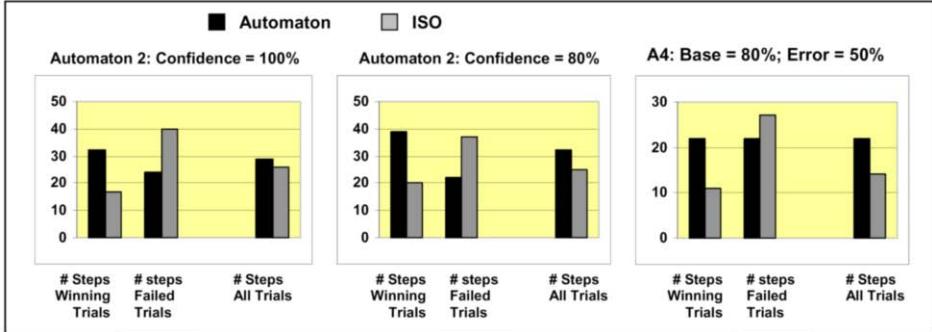


Figure 2 – Reduction in number of search steps required, relative to the unaided user as simulated by Automatons 2 and 4. In the first two columns the automatons used a fixed confidence level; in the last column the random inputs indicated lower confidence. A "Winning Trial" means the ISO algorithm succeeded in reducing the number of steps. All data points are an average of 100 trials.

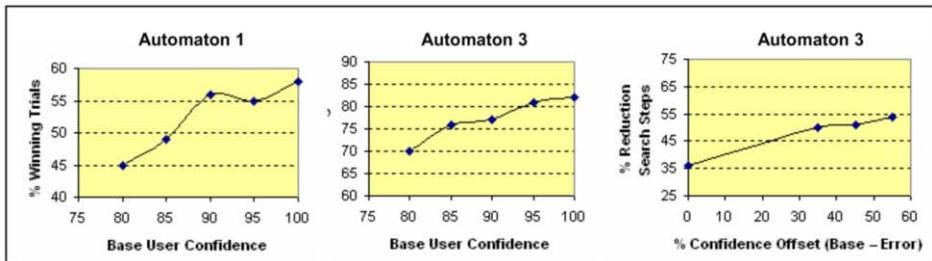


Figure 3 – The first two graphs show the proportion of winning trials for ISO, as a function of the user confidence parameter. Error confidence is constant at 50%. For the 3rd graph error confidence varies while base confidence is held constant at 85%. The % reduction for ISO is shown as a function of their difference.

4. Related Work

The ISO algorithm has parallels with Estimation of Distribution algorithms (EDA) [6,7]. After unification is applied, ISO filters the results via the local and global consistency indices; this may be viewed as a selection function. Unification itself amounts to construction of a distribution and its use to perturb candidate actions.

However, there are key technical differences between ISO and EDAs. Whereas EDAs retain the basic GA notion of perturbing the entire surviving population, our algorithm selects particular individuals for Unification, as dictated by the current user Response. Of course the current user Response is an influence that is absent in the conventional GA problem setting. Further, in contrast to ISO, EDAs and evolutionary algorithms in general are concerned at any generation only with the currently surviving population. Additional important differences may be added to this list.

Similarity with the "online" form of Neural Networks maybe noted [8]; more specifically, Recurrent Neural Networks [9,10] learn from a history of errors, as does ISO. However ISO differs in many ways from any form of neural net. An obvious difference is that it does not adjust weights.

In our view, the key innovation of ISO is the way the problem is cast rather than the technical particulars of the algorithm used to solve it. Simply, we consider a search where we begin without any well-defined criteria for testing a correct solution. This of course, is exactly what happens when a human user searches email or the internet for a target that is only vaguely remembered. In the next section we explore the implications of treating this situation as a legitimate computational problem.

A more significant similarity may be observed between ISO and so-called "Putnam-Gold Machines" [11]. Computation in the latter do not have a fixed ending. Similarly, any particular output produced by ISO, is not, in general, the "real" output.

5. Discussion and Conclusions

The ISO algorithm is not a self-contained "artificial intelligence", since it relies explicitly on human direction. Nevertheless, we have an AI "result", i.e. the useful outcome of reduced search times. This is achieved by artificial enhancement of one part of a human cognitive process, rather than by a process that is entirely artificial.

As noted in the introduction, Turing and his colleagues believed that Turing computation, i.e. any mechanical process is necessarily devoid of intelligence. This suggests that to achieve general intelligence we must go beyond Turing computation. What follows is an informal argument, suggesting that the ISO technique offers a way to actively participate in just such a non-Turing computation.

Proposition 1:

A Turing computation is completely specified, hence must have a well-defined goal. At a minimum, the definition of this goal is the specification of what it does. The exception is random input, i.e. a non-deterministic machine; the random influence makes the machine incompletely specified.

Proposition 2:

The search problem for ISO does not have a well-defined goal, hence not a Turing computation. Likewise, the sequence of user inputs to ISO is non-Turing.

Next we show that this non-Turing behavior is not random. The ISO method interprets user choices based on the earlier history of those same choices, i.e. we assume the user may not have the memory or analytical resources to choose in a manner that is consistent with previous choice behavior. The implicit assumption is that effective native user progress is itself based in some sense on offsets from previous choices. Put another way, we assume the user would want to be consistent with the history, if they did have the resources.

If the ISO algorithm consistently reduces search times, then a correspondence must exist between the sequence of action deltas generated by the algorithm and the goal of the process at work in the mind of the user. For example, it may be that the sequence approaches the target of the search in some limit sense. Alternatively, the sequence of outputs from ISO may correspond to a succession of revisions of the target, without specifying any structure on the succession of targets or the basis for revision. There may be other ways of setting up this correspondence.

Denote by $I = [i_1, i_2, \dots, i_n]$ the sequence of inputs provided by the user to ISO and by $O = [o_1, o_2, \dots]$ the outputs it generates in response. That a correspondence exists between o_i and i_i is straightforward. The more significant observation

however, is that each o_i relates in some way to i_n , the final target of the search. This must be so since the user succeeds in reaching the target in fewer steps when supported by ISO than with unaided intuition alone. We say that the ISO algorithm is *synchronizing* with the user cognitive process. We have:

Proposition 3:

It is not possible to synchronize with a random sequence. The ISO algorithm synchronizes with the user input sequence, so that sequence is non-random. *Hence ISO synchronizes with a non-random, non-Turing computation.*

The ISO algorithm, by exhibiting the above-described synchronization behavior, offers a means to explore the process by which such a non-Turing results may be arrived at. It also suggests that explicit formulation of one of the distributed components of a cognitive process may be a promising alternative paradigm for "algorithm". Learning from experiment with such working systems, we may be able to construct explicit enhancements – or indeed replacements – for progressively more of the components of such processes. Eventually, we may construct all of them, hence achieving a self-contained artificial intelligence.

References

- [1] T. Ord, *The Many Forms of Hypercomputation*, Applied Mathematics and Computation, Vol .178, 2006, pp. 143-155.
- [2] M. H. A. Newman, *Allan Mathison Turing*, Biographical Memoirs of the Royal Society, 1955, pp. 253-263.
- [3] J. Hollan, E. Hutchins and D. Kirsh, *Distributed Cognition: Toward a New Foundation for Human-Computer Interaction Research*, ACM Transactions on Computer-Human Interaction, Vol. 7, No. 2, June 2000, Pages 174–196.
- [4] C. Heintz, *Web search engines and distributed assessment systems*, In: Harnad, Stevan and Dror, Itiel E. (eds.), *Distributed Cognition: Special issue of Pragmatics & Cognition* 14:2 (2006). 268 pp. (pp. 387-409)
- [5] G. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Prentice Hall, 1995.
- [6] P. Larrañaga and J. A. Lozano (Eds.), *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*, Series: Genetic Algorithms and Evolutionary Computation , Vol. 2, Kluwer Academic, 2001.
- [7] M. Pelikan, D. E. Goldberg and F. G. Lobo, *A Survey of Optimization by Building and Using Probabilistic Models*, Computational Optimization and Applications, Springer, 2002, pp.5-20.
- [8] D. Saad (ed.), *Online Learning in Neural Networks*, Cambridge University Press, January 1999.
- [9] M. Danilo, and J. Chambers, *Recurrent Neural Networks*, Wiley, 2001.
- [10] S. Hochreiter and J. Schmidhuber, *Long Short-Term Memory*, Neural Computation, Vol 9 No 8, 1997, pp. 1735-1780.
- [11] P. Kugel, *Computing Machines Can't Be Intelligent (...and Turing Said So)*, Minds and Machines, Vol.12, No. 4, September 2002.

Input Feedback Networks: Classification and Inference Based on Network Structure

Tsvi Achler and Eyal Amir

Department of Computer Science, University of Illinois at Urbana-Champaign

We present a mathematical model of interacting neuron-like units that we call Input Feedback Networks (IFN). Our model is motivated by a new approach to biological neural networks, which contrasts with current approaches (e.g. Layered Neural Networks, Perceptron, etc.). Classification and reasoning in IFN are accomplished by an iterative algorithm, and learning changes only structure. Feature relevance is determined during classification. Thus it emphasizes network structure over edge weights. IFNs are more flexible than previous approaches. In particular, integration of a new node can affect the outcome of existing nodes without modifying their prior structure. IFN can produce informative responses to partial inputs or when the networks are extended to other tasks. It also enables recognition of complex entities (e.g. images) from parts. This new model is promising for future contributions to integrated human-level intelligent applications due to its flexibility, dynamics and structural similarity to natural neuronal networks.

Introduction

Regulation through feedback is a common theme found in biology including gene expression and physiological homeostasis. Furthermore, feedback connections can be found ubiquitously throughout neuronal networks of the brain. Yet, the role of feedback in neuronal networks is often under-appreciated. Most connectionist models determine connection weights during a learning phase and during the testing phase employ a simple feedforward structure.

The contribution of this paper is to demonstrate that feedback employed during the test phase can perform powerful distributed processing. In our feedback model every output node only inhibits its own inputs. Thus the model is named *Input Feedback Networks* (IFN). This paper reinforces preliminary results of IFN [1, 2] and explores its ability to perform recognition and intelligent inference.

Our analysis is divided into two subsections: combinatorial considerations and functional analysis. The combinatorial section reviews various networks, discussing the implications of adding new representations and the plausibility of large networks. The functional analysis section explores through exemplars how IFN behaves in complex scenarios.

The scenarios are: 1) when an output node with a small input vector completely overlaps with an output node composed of a larger input vector. Subsequently, the smaller vector is innately inhibited given the inputs of the larger vector. 2) Multiple overlapping representations can cooperate and compete under differing circumstances. 3) Network ‘logic’ can be adjusted by simply biasing activation of an output. 4)

Depending on inputs states, inference can be conducted through distributed processing over an infinite number of chains. 5) Overlap of representations determines processing difficulty and the role initial conditions may have on inference.

Some of these results are preliminary. However, it is clear that based on a simple premise of feedback, IFN offers a flexible structure and dynamic approach to classification of stimuli and Artificial Intelligence.

1. Background

Traditional connectionist classifier models can be broken down into two broad strategies: 1) *Neural Network* (NN) type algorithms that primarily rely on connection weight adjustments, implemented by learning algorithms. 2) *Lateral Competition* (LC) algorithms that involve competition between ‘similar’ representations. LC and NN methods can overlap. For example, weight adjustments occur between LC connections. We argue that both connection weights and direct output inhibition are specific to tasks. Furthermore they can be combinatorically implausible and can limit processing.

1.1. Connection Weight Adjustments

The idea of adjusting connection weights has been a corner stone throughout the development of connectionist models including Perceptrons, Parallel Distributed Networks, Markov Models, Bayesian Networks, Boltzmann Machines, and even Support Vector Machines [3-7]. Networks based on weight adjustment are powerful, flexible and in combination with learning algorithms can have a good degree of autonomy. These methods allow high degrees of freedom where numerous sets of weights can be chosen for virtually any problem. Weights are adjusted per task for specific applications. However without appropriately selected training sets, this approach suffers from difficulties such as over-fitting, local minima and catastrophic interference (‘forgetting’ of previously learned tasks, e.g. [8, 9]). The role of Neural Network (NN) structure in relation to function or image parts is unclear. Learning algorithms are difficult to describe in terms of biologically viable neural mechanisms. Lastly, recent studies of neuron processing, challenge the idea that a synapse can be estimated by a connection weight [10, 11].

1.2. Lateral Competition

In LC models cells inhibit their neighbors or their neighbors’ inputs. Such models include: Hopfield, Winner-take-all (WTA) and Lateral Inhibitory Networks i.e. [12-15]. In WTA networks are engineered so that a cell inhibits all other possible representations. Thus every cell must be connected to a common inhibitory network. This puts biologically implausible combinatorial demands on connectivity and limits parallel processing. WTA does not address how recognition of parts interacts with overall classification. Carpenter and Grossberg recognized this problem and proposed a mechanism which evaluates and ranks the number of inputs used in each part [12]. The output(s) encompassing the most inputs is chosen as the best solution. But this mechanism evaluates one cell at a time, and does not address when partial representations should compete.

Lateral Inhibition and Hopfield networks are relaxed versions of WTA where the amount of competition between cells is engineered with LC connection weights. Lateral inhibition was originally proposed to describe neuronal activity within the retina based on neighboring cells with simple spatial receptive fields. The closest neighbors compete most and competition decreases as cells are farther apart spatially (i.e. [15]). However lateral inhibitory and Hopfield connections become somewhat intractable as object representations become more complex because the number of possible competitors becomes huge. Let's take the pattern '1' for example, '1' can compete with the representation of letter 'I', vertical bars, other letters, other numbers or anything vertical.

2. Input Feedback Network Structure and Function

IFN is composed of simple binary connections between input features and output vectors. Yet, this becomes surprisingly powerful when combined with regulatory feedback to inputs during the testing phase [2]. This structure maintains its simplicity in large networks but can still make complex recognition decisions based on distributed processing.

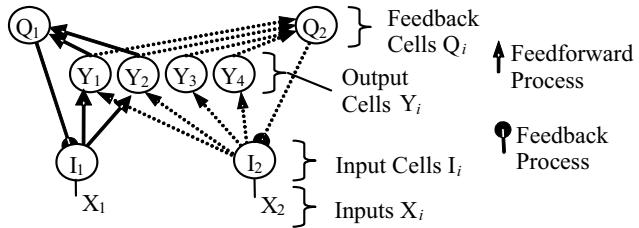


Figure 1. Input Feedback Schematic. Every feedforward connection has an associated feedback connection. If I_1 (e.g. white) projects to Y_1 (e.g. pingpong) & Y_2 (lychee), then Q_1 must receive projections from Y_1 & Y_2 and provide feedback to the input cell I_1 . Similarly if I_2 (e.g. round) projects to Y_1 , Y_2 , Y_3 (orange), & Y_4 (planet), then Q_2 receives projections from Y_1 , Y_2 , Y_3 , & Y_4 and projects to I_2 .

The IFN structure is shown in Figure 1. The network delineates a simple rule of connectivity based on a triad of interconnections between an input, the output it supports, and feedback from that output. Every input has a corresponding feedback 'Q', which samples the output processes that the input cell activates and modulates the input amplitude. These triads are intermeshed independently of what is happening with other such triads of other inputs. The underlying triad holds true from both the perspective of the input processes and output processes. Every output must project to the feedback Q processes that correspond to the inputs the output receives. For example, if an output process receives inputs from I_1 and I_2 it must project to Q_1 and Q_2 . If it receives inputs from I_1 , it only needs to project to Q_1 .

The networks are designed to dynamically re-evaluate each cell's activation based on 1) input feedback onto the individual inputs in order to 2) modify the input state based on the input's use. The input's use is inversely proportional to the output cells that the input activates. Lastly 3) re-evaluating each cell's activity based on its state and the re-evaluated input. Steps 1-3 are cycled through continuously as the network converges to its solution. Each cell inhibits only its inputs based on input feedback. Feedback provides a continuous measure of the use of each input, which determines competition [1, 2, 16, 17].

2.1. Dynamic Evaluation of Ambiguity

Inputs that project to multiple simultaneously active output nodes are ambiguous. They are ambiguous because many representations use them. Such inputs hold little value and become inhibited. In contrast, inputs that project to one non-active representation are boosted.

The only way an output cell process can receive full activation from the input is if it is the only cell active using that input, reducing Q to 1. If two competing cells share similar inputs, they inhibit each other at the common inputs, forcing the outcome of competition to rely on other non-overlapping inputs. The more overlapping inputs two cells have, the more competition exists between them. The less overlap between two cells, the less competition, more ‘parallel’ or independent from each other the cells can be. This is a trend supported by human search efficiency data that compares similarity between search objects and reaction times [18].

2.2. Simple Connectivity

Feedback networks do not require a vast number of connections; the number of connections required for competition is a function of the number of inputs the cell uses. Addition of a new cell to the network requires only that it forms symmetrical connections about its inputs and not directly connect with the other output cells. Thus the number of connections of a specific cell in feedback competition is independent of the size or composition of the classification network, allowing large and complex feedback networks to be combinatorially and biologically practical.

2.3. Flexibility

IFN is flexible because it doesn’t a-priori define which input is ambiguous. Which input is ambiguous depends on which representation(s) are active which in turn depends on which stimuli and task are being evaluated.

2.4. Neurophysiology & Neuroanatomy Evidence

The ability to modify information in earlier pathways appears to be an important component of recognition processing. Input feedback connections can be found in virtually all higher areas of brain processing. The thalamic system (including thalamic nuclei the Lateral Geniculate Nucleus-LGN, and Medial Geniculate Nucleus-MGN) projects to all of the cerebral cortex and receives extensive cortical innervations back to the same processing areas. Relay cells feed forward to the cortex and to pyramidal cells which feed back from the cortex. Together they form the ubiquitous triad structure of input feedback. The thalamus may even receive more feedback connections from the cortex than it projects to the cortex [19].

The *Olfactory Bulb* (OB), which processes odors, is analogous to the thalamus. The OB is a separate, modular structure which can easily be studied. Compared to visual or auditory signals, odorous signals are generally reduced in spatial fidelity [20]. Thus odorant processing involves primarily recognition processing as opposed to visual or auditory processing which can encompass both recognition and localization.

Input feedback modulation of early processing appears not one, but two levels of processing within the OB: the local and global circuits. The local circuit found within

the OB glomeruli sub-structure receives inputs from the olfactory nerve, and connects to mitral/tufted output cells. The output cells simultaneously activate juxtaglomerular cells (Q cell equivalent) which pre-synaptically inhibit the olfactory nerve axons. The global circuit receives inputs through the mitral/tufted cells, within the next structure, the olfactory cortex and projects information back to granule cells within the OB (another Q cell equivalent) which inhibit the mitral/tufted cells.

Nonlinear mechanisms which alter cell activation based on previous activity are found in dendritic-dendritic connections involved in coincidence detection via intercellular NMDA channels. These channels are found in both granule cells [21] and juxtaglomerular cells [22]. Together GABA channels (inhibitory connections), calcium & NMDA channels (multiplicative dynamics) form an input feedback system of inhibition [23, 24].

2.5. Mathematical Formulation and Investigation

This section introduces general nonlinear equations governing IFN. For any cell/vector Y denoted by index a , let N_a denote the input connections to cell Y_a . For any input edge I denoted by index b , let M_b denote the feedback connections to input I_b . The amount of shunting inhibition at a single input I_b is defined as Q_b for that input. Q_b , is a function of the sum of activity from all cells Y_j that receive activation from that input:

$$Q_b = \sum_{j \in M_b} Y_j(t) \quad Eq\ 1 \qquad I_b = \frac{X_b}{Q_b} \quad Eq\ 2$$

Input I_b is regulated based on the on Q_b , which is determined by the activity of all the cells that project to the input, and driven by X_b which is the raw input value. The activity of Y_a is dependent on its previous activity and the input cells that project to it. The input activity that is transferred to the output cells is inversely proportional to the Q feedback.

$$Y_a(t + \Delta t) = \frac{Y_a(t)}{n_a} \sum_{i \in N_a} I_i \quad Eq\ 3$$

$$= \frac{Y_a(t)}{n_a} \sum_{i \in N_a} \frac{X_i}{Q_i} = \frac{Y_a(t)}{n_a} \sum_{i \in N_a} \left(\frac{X_i}{\sum_{j \in M_i} Y_j(t)} \right) \quad Eq\ 4$$

2.6. Stability

Stability of these equations presented here and variations including negative values have been previously analyzed [25, 26]. The subset used here are limited to positive values of all variables, thus these equations will always be positive given positive values of the components. Thus the values of Y can not become negative and have a lower bound of 0.

Furthermore the values have an upper bound. By definition since N_a is the set of input connections to cell Y_a , then M_b will contain cell Y_a the within the set of input feedback connections to input I_b . To achieve the largest possible value, all other cells should go to zero. In that case, the equation then reduces to

$$Y_a(t + \Delta t) \leq \frac{1}{n_a} \sum_{i \in N_a} \left(\frac{Y_a(t) \cdot X_i}{Y_a(t)} \right) = \frac{1}{n_a} \sum_{i \in N_a} X_i \leq \frac{X \max \cdot n_a}{n_a} = X \max \quad Eq\ 5$$

where n_a is the number of processes in set N_a . If X values are bounded by an X_{max} then the values of Y are bounded by positive numbers between zero and X_{max} .

The Picard existence and uniqueness theorem states that if a differential equation is bounded and is well behaved locally then will have a unique solution i.e. [27].

2.7. Learning

The distinguishing feature of this approach is the emphasis on the test phase as opposed to the conventional emphasis on learning. Thus, we purposefully deemphasize learning. Since IFN uses simple positive binary connections, learning is simpler. This means that only features present during label presentation are encoded. Negative associations such as ' Y_I ' is unlikely given feature ' X_I ', are not encoded. Instead, they are estimated using the recursive feedback processes. Thus learning can involve encoding simple (Hebbian-like) correlations between input features and output vector labels. More sophisticated learning methods can include clustering and pruning ie. [28].

3. Combinatorial Analysis

In this section we outline combinatorial problems of adding a new node to large networks and describe how IFN structure avoids these problems.

3.1. The Ping-Pong, Peeled Lychee Example

Assume three huge brains composed of each type of network (NN, LC, IFN) and recognizes among other things a ping-pong ball with a 'ping-pong ball' cell [29]. Now a new stimulus appears for the first time: a peeled lychee. It has many similar input features to a ping-pong ball (ie. color and size). Each model should 1) incorporate a lychee node 2) assure the 'ping pong' node does not predominate given lychee input features.

In all of the brains the input features representative of peeled lychee must connect to the node representative of lychee. In NNs the whole network may potentially be adjusted. Given lychee the connection weights of the features that support lychee are boosted and those that support ping-pong are reduced. Similarly in LC given lychee, weighted lateral connections must be created where the representation of lychee inhibits the representation of ping-pong (and any other similar representation) and visa versa. No matter how obscure or unlikely the relation of lychees and ping-pong balls are in most life situations NNs and LC networks must re-arrange their structure for it. This can potentially degrade previously learned associations [8, 9]. As the networks become larger, the cost of evaluating each node against the others becomes impractical. In IFN the node's input-output relations are sufficient. Thus no further modification of the structure is required.

In summary, in LC the number of connections can be implausible: potentially every node may be required to connect to every other node (WTA). In weight adjustment paradigms (NNs), connection weights are allowed a high degree of freedom and need to be adjusted for all tasks a-priori. IFNs rely on input feedback instead of direct connections or weights and do not require modification. Thus are more combinatorially practical.

4. Functional Analysis

We analyze interactions between different compositions of node representations and degenerate cases to assess their informative value. We analyze 1) how nodes with smaller representations that are overlapped by larger representations behave 2) how multiple partially overlapping representations can simulate a binding scenario, where certain parts cooperate and others compete, 3) control of overlapping behavior 4) the behavior of infinite number of partial overlapping representations linked in chains, and 5) winner-less competition.

4.1. Composition by Overlap of Nodes

In the most general example, example 1, two cells are connected such that the inputs of Y_1 (input A) completely overlaps with a larger Y_2 . But Y_2 also receives an independent input, B [2]. In IFN, Y_1 has one input connection, thus by definition its input connection weight is one. It is ‘Homeostatic’: its maximal activity is 1 when all of its inputs (input A) are 1. Y_2 has two input connections so for it to be ‘Homeostatic’ its input connection weights are one-half. When both inputs (A & B) are 1 the activity of Y_2 sums to 1. Note these weights are predetermined by the network. Connections are permanent and never adjusted. Input A projects to both Y_1 & Y_2 , thus receives inhibitory feedback from both Y_1 & Y_2 . Input B projects only to Y_2 so it receives inhibitory feedback from Y_2 . The most encompassing representation will predominate without any special mechanism to adjust the weighting scheme. Thus, if inputs A and B are active Y_2 wins. This occurs because when both inputs are active, Y_1 must compete for all of its inputs with Y_2 , however Y_2 only needs to compete for half of its inputs (the input shared with Y_1) and it gets the other half ‘free’. This allows Y_2 to build up more activity and in doing so inhibit Y_1 .

The solutions are presented as (*input values*) \rightarrow (*output vectors*) in the pattern $(X_A, X_B,) \rightarrow (Y_1, Y_2)$. The steady state solution for example 1 is $(X_A, X_B) \rightarrow (X_A - X_B, X_B)$. Substituting our input values we get $(1,1) \rightarrow (0,1)$, $(1,0) \rightarrow (1,0)$. Given only input A the smaller cell wins the competition for representation. Given both inputs the larger cell wins the competition for representation. Though we used binary inputs, the solution is defined for any positive real X input values [2]. The mathematical equations and their derivation can be found in the Appendix.

Thus smaller representation completely encompassed by a larger representation become is inhibited when the inputs of the larger one are present. The smaller representation is unlikely given features specific only to the large representation. It demonstrates that IFN determines negative associations ($'Y_1'$ is unlikely given feature ' X_B ') even though they are not directly encoded. This is a general example and data sets may have many forms of overlap.

In order to encode such negative associations using conventional methods, they would have to be ‘hard-wired’ into the network. With NN, each possible set of stimuli combinations would have to be trained. With LC each possible negative association would have to be explicitly connected.

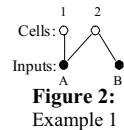


Figure 2:
Example 1

4.2. Multiple Parts in Binding Scenario

IFN can simultaneously evaluate the criteria of three different representations. In e.g. 2, expanded from e.g. 1, three cells partially overlap. As in e.g. 1, Y_1 competes for its single input with Y_2 . However, now Y_2 competes for its other input with Y_3 and Y_3 competes for only one of its inputs.

The steady state solution is $(X_A, X_B, X_C) \rightarrow (X_A - X_B + X_C, X_B - X_C, X_C)$. If $X_B \leq X_C$ then $Y_2 = 0$ and the equations become $(X_A, 0, \frac{X_B + X_C}{2})$. If $X_C = 0$ the solution becomes

that of e.g. 1: $(X_A, X_B, 0) \rightarrow (X_A - X_B, X_B, 0)$. The results are: $(1, 0, 0) \rightarrow (1, 0, 0)$;

$(1, 1, 0) \rightarrow (0, 1, 0)$; $(1, 1, 1) \rightarrow (1, 0, 1)$. Derivations can be found in the appendix.

Thus if input A is active, Y_1 wins. If inputs A and B are active and Y_2 wins for the same reasons this occurs in e.g 1. However, if inputs A, B and C are active then Y_1 and Y_3 win. The network as a whole chooses the cell or cells that best represent the input pattern with the least amount of competitive overlap.

In e.g. 2, Y_2 must compete with all of its inputs: A with Y_1 , B with Y_3 . Y_3 only competes for half of its inputs (input B) getting input C ‘free’. Since Y_2 is not getting its other input (input A) ‘free’ it is at a competitive disadvantage to Y_3 . Together Y_1 and Y_3 , mutually benefit from each other and force Y_2 out of competition. Competitive information travels indirectly ‘through’ the representations. Given active inputs A and B, the activity state of Y_1 is determined by input C through Y_3 . If input C is 0 then Y_1 becomes inactive. If input C is 1, Y_1 becomes active. However, Y_3 does not even share input A with Y_1 .

4.2.1. Binding

Choosing Y_2 given inputs A, B, and C is equivalent to choosing the irrelevant features for binding. Below we attempt to put this issue in a ‘binding’ context. Lets assign inputs A, B, and C to represent spatially invariant features of an image of a pickup truck  where feature A represents circles, C represents the truck body (without chassis and wheels), and feature B represents a horizontal bar. Y_1 is assigned to represent wheels and thus when it is active, feature A is interpreted as wheels. Y_2 represents a barbell  composed of a bar adjacent to two round weights (features A and B). Note: even though Y_2 includes circles (feature A), they do not represent wheels (Y_1), they represent barbell weights. Thus if Y_2 is active feature A is interpreted as part of the barbell. Y_3 represents a pickup truck body without wheels (features B and C), where feature B is interpreted as part of the truck chassis. Now given an image of a pickup, all features simultaneously (A, B and C), choosing the barbell (Y_2) even though technically a correct representation, is equivalent to a binding error within the wrong context in light of all of the inputs. In that case the complete picture is not analyzed in terms of the best fit given all of the information present. Similar to case 1, the most encompassing representations mutually predominate without any special mechanism to adjust the weighting scheme.

Thus the networks are able to evaluate and bind representations in a sensible manner for these triple cell combinations. To emulate this in traditional network each possible combination will have to be trained for each input combination.

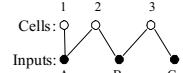


Figure 3: Example 2

4.3. Search

The previous section demonstrated that these networks display useful distributed properties. In this section we describe how the networks can be modified to perform tasks.

Suppose we want to use our network to ‘search’ for specific stimuli. Object-specific neurons in the temporal lobe show biasing (a slight but constant increase in baseline activity of that cell) when the animal is trained to actively look for that shape. During recognition, the biased cell rapidly gains activity at the expense of others [30]. Our network can be biased in a similar fashion. We repeat example 2 but want to ask the question: can a barbell shape be present? We introduce a small bias to Y_2 (representing barbell) according to the equation:

$$Y_2(t+dt) = \frac{Y_2(t)}{2} \left(\frac{X_A}{Y_1(t)+Y_2(t)} + \frac{X_B}{Y_2(t)+Y_3(t)} \right) + b$$

Choose a bias b of 0.2 and activating all inputs $(1, 1, 1) \rightarrow (0.02, 0.98, 0.71)$. The network now overrides its inherent properties and indicates that the inputs of Y_2 are present.

Thus network function can be radically adjusted simply by biasing activation. Most importantly, this did not require adjustment of any connection weights. Biasing involved only the desired representation and can be easily turned on and off. In traditional methods such as NNs weights would need to be re-learned and redistributed throughout the network for this new task.

4.4. Composition by Infinite Chains

Cells can be further linked at infinitum and the cell representations interact indirectly by transferring their dynamic activation through the chain.

4.4.1. Including a 1-Input cell

Consider the for example case where there are N cells, and N inputs, and all inputs are 1. If N is an *odd* number then at steady state the *odd* numbered cells will be 1 and even ones 0. If N is *even*, the *even* cells will be 1 and *odd* ones zero. The general solutions (Y_1, Y_2, \dots, Y_N) where i & j represent cell numbers are:

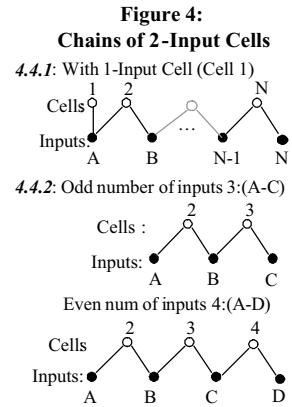
$$\left(\sum_{1 \leq j \leq N} X_{odd} - X_{even}, \sum_{2 \leq j \leq N} X_{even} - X_{odd}, \sum_{i \leq j \leq N} X_{even} - X_{odd}, X_N \right)$$

For example with $N=4$: $(1,1,1,1) \rightarrow (0,1,1)$ and $N=5$: $(1,1,1,1,1) \rightarrow (1,0,1,1)$. Thus these input configurations can be represented by binary values.

4.4.2. Without a 1-Input cell

If cell 1 (the only cell with one input) is not present, then the network does not have a favorable set of cells to resolve an odd number of inputs. Two-input cells can not match in a binary manner the inputs of an odd numbered chain.

Thus, in these cases the solution becomes more complicated. In the case of three inputs (covered by two cells) the mathematical solutions are:



$$Y_1 = \frac{X_A(X_A + X_B + X_C)}{2(X_A + X_C)} \quad Y_2 = \frac{X_C(X_A + X_B + X_C)}{2(X_A + X_C)}.$$

When inputs are (1,1,1) then output cells become ($\frac{3}{4}$, $\frac{3}{4}$). Furthermore if only the middle input is active (0,1,0) then the forces on both cells are symmetrical, the equation collapses to $2(Y_1(t) + Y_2(t)) = X_B$ and the solution depends on initial conditions (also see section 4.5).

In case of 4 inputs distributed over 3 cells the solution becomes:

$$\left(\frac{X_A(\Sigma X)}{2(X_A + X_C)}, \frac{-(\Sigma X)(X_A X_D - X_C X_B)}{2(X_A + X_C)(X_B + X_D)}, \frac{X_D(\Sigma X)}{2(X_B + X_D)} \right) \text{ Where } \Sigma X = X_A + X_B + X_C + X_D.$$

When all inputs are 1, the cells settle on a binary solution (1,0,1). Cases with more inputs become progressively more complicated. Thus the structure can greatly affect the ability to efficiently represent inputs.

4.4.3. Subchains

If any input in the chain is zero, this will break the chain into independent components composed of the right and left parts of the chain from the zero input. These can function as smaller chains. For example if input D=0, the chains involving inputs A-C and E-N become independent. Thus the outcome of network, is determined by distributed cell dynamics involving input values and cell representations. Further analysis is remains for future research.

4.5. Analysis of Node Overlap

Lastly overlap is a key determinant of processing independence [2]. If two representations completely overlap they may also be dependent on initial conditions of the cells. Suppose there exists only two cells Y_1 & Y_2 with n_{1indep} and n_{2indep} representing each cells' independent inputs. Furthermore K_1 & K_2 represent the average input values to these independent inputs. The steady state solution is of the form:

$$k_{initial} \frac{Y_1^{n_1}}{Y_2^{n_2}} = e^{(n_{2indep} - n_{1indep} + \frac{n_1 K_1}{Y_1} - \frac{n_2 K_2}{Y_2})t} = e^{\lambda t},$$

where $k_{initial}$ represents initial conditions of the network. See derivations in Appendix.

If $\lambda > 0$ then $Y_1(t=\infty) \rightarrow 0$, if $\lambda < 0$ then $Y_2(t=\infty) \rightarrow 0$. In these cases either Y_1 or Y_2 predominates. However, if $\lambda = 0$ then the solution is not sufficiently independent and is a function with the form $Y_2 = k Y_1$. The initial value of k will affect the final solution. Further analysis is the topic of future research.

5. Conclusion

With simple combinatorially-plausible binary relations input feedback offers a flexible and dynamic approach to intelligent applications. It is well suited for classification of stimuli composed of parts because its distributed interactions can resolve overlap of multiple representations. These interactions give preference to the broadest representations and can determine when representations should cooperate or compete. But, if representations are not sufficiently independent they may depend on initial

conditions. However, the network can also be manipulated to perform search tasks by biasing output representations. These properties demonstrate that such networks can be an integral part of intelligent inference and provide a new direction for future research.

Appendix

Section 4.1, Example 1

IFN equations are: $Y_1(t+dt) = \frac{Y_1(t)X_A}{Y_1(t)+Y_2(t)}$, $Y_2(t+dt) = \frac{Y_2(t)}{2} \left(\frac{X_A}{Y_1(t)+Y_2(t)} + \frac{X_B}{Y_2(t)} \right)$.

The network solution at steady state is derived by setting $Y_1(t+dt)=Y_1(t)$ and $Y_2(t+dt)=Y_2(t)$ and solving these equations. The solutions are $Y_1 = X_A - X_B$ and $Y_2 = X_B$. If $X_A \leq X_B$ then $Y_1 = 0$ and the equation for Y_2 becomes: $Y_2 = \frac{X_A + X_B}{2}$.

Section 4.2, Example 2

Equation $Y_1(t+dt)$ remains the same as example 1. $Y_2(t+dt)$ and $Y_3(t+dt)$ become:

$$Y_2(t+dt) = \frac{Y_2(t)}{2} \left(\frac{X_A}{Y_1(t)+Y_2(t)} + \frac{X_B}{Y_2(t)+Y_3(t)} \right), \quad Y_3(t+dt) = \frac{Y_3(t)}{2} \left(\frac{X_B}{Y_2(t)+Y_3(t)} + \frac{X_C}{Y_3(t)} \right)$$

Solving for steady state by setting $Y_1(t+dt)=Y_1(t)$, $Y_2(t+dt)=Y_2(t)$, and $Y_3(t+dt)=Y_3(t)$, we get $Y_1=X_A-X_B+X_C$, $Y_2=X_B-X_C$, $Y_3=X_C$. If $X_C=0$ the solution becomes that of e.g. 1: $Y_1=X_A-X_B$ and $Y_2=X_B$. If $X_B \leq X_C$ then $Y_2=0$ and the equations become $Y_1=X_A$ and $Y_3=\frac{X_B+X_C}{2}$.

Section 4.5

The overlap region is defined by the number of inputs that overlap between two cells N_{over} , and the number of inputs that are independent of overlap N_{indep} . If Y_1 & Y_2 overlap then N_{over} of $Y_1 = N_{over}$ of Y_2 . Thus $n_1=n_{1indep}+n_{over}$ and $n_2=n_{2indep}+n_{over}$. Thus K_{over} and N_{over} of cell 1 = K_{over} and N_{over} of Y_2 by definition of overlap:

$$Y_1(t+dt) = \frac{Y_1(t)}{n_1} \left(\frac{n_{1indep} K_1}{Y_1(t)} + \frac{n_{over} K_{over}}{Y_2(t)+Y_1(t)} \right) \text{ and } Y_2(t+dt) = \frac{Y_2(t)}{n_2} \left(\frac{n_{2indep} K_2}{Y_2(t)} + \frac{n_{over} K_{over}}{Y_1(t)+Y_2(t)} \right)$$

Substituting $Y(t+dt)=Y+Y'$ and subtracting out the overlap region the equations reduce to: $n_1 \frac{Y_1'}{Y_1} - n_2 \frac{Y_2'}{Y_2} = n_2 - n_1 + \frac{n_{1indep} K_1}{Y_1} - \frac{n_{2indep} K_2}{Y_2}$. Integrating and raising

$$\text{exponents of both sides we get: } k_{initial} \frac{Y_1^{n_1}}{Y_2^{n_2}} = e^{\left(\frac{n_{2indep} K_2 - n_{1indep} K_1}{Y_1} t \right)} = e^{\lambda t}$$

Acknowledgements

We would like to thank Burak M. Erdogan, Frances R. Wang and anonymous reviewers for helpful suggestions. This work was supported by the U.S. National Geospatial Agency Grant HM1582-06--BAA-0001.

References

- [1] Achler, T., *Input shunt networks*. Neurocomputing, 2002. **44**: p. 249-255.
- [2] Achler, T., *Object classification with recurrent feedback neural networks*. Proc. SPIE Evolutionary and Bio-inspired Computation: Theory and Applications, 2007. **6563**.
- [3] Hinton, G.E. and T.J. Sejnowski, *Unsupervised learning : foundations of neural computation*. Computational neuroscience. 1999, Cambridge, Mass.: MIT Press. xvi, 398 p.
- [4] Minsky, M.L. and S. Papert, *Perceptrons; an introduction to computational geometry*. 1969, Cambridge, Mass.,: MIT Press. 258 p.
- [5] Rosenblatt, F., *The perceptron: a probabilistic model for information storage and organization in the brain*. Psychol Rev, 1958. **65**(6): p. 386-408.
- [6] Rumelhart, D.E., J.L. McClelland, and University of California San Diego. PDP Research Group., *Parallel distributed processing : explorations in the microstructure of cognition*. Computational models of cognition and perception. 1986, Cambridge, Mass.: MIT Press.
- [7] Vapnik, V.N., *The nature of statistical learning theory*. 1995, New York: Springer. xv, 188 p.
- [8] Lewandowsky, S., *Catastrophic Interference and Generalization in Neural Networks*. International Journal of Psychology, 1992. **27**(3-4): p. 653-653.
- [9] Rueckl, J.C., *Reducing Catastrophic Interference through Weight Modulation*. Bulletin of the Psychonomic Society, 1992. **30**(6): p. 457-457.
- [10] Marder, E. and J.M. Goaillard, *Variability, compensation and homeostasis in neuron and network function*. Nat Rev Neurosci, 2006. **7**(7): p. 563-74.
- [11] Turrigiano, G.G. and S.B. Nelson, *Homeostatic plasticity in the developing nervous system*. Nat Rev Neurosci, 2004. **5**(2): p. 97-107.
- [12] Carpenter, G.A. and S. Grossberg, *A Massively Parallel Architecture for a Self-Organizing Neural Pattern-Recognition Machine*. Computer Vision Graphics and Image Processing, 1987. **37**(1): p. 54-115.
- [13] Douglas, R.J. and K.A. Martin, *Neuronal circuits of the neocortex*. Annu Rev Neurosci, 2004. **27**: p. 419-51.
- [14] Hopfield, J.J., *Neural Networks and Physical Systems with Emergent Collective Computational Abilities*. Proceedings of the National Academy of Sciences of the United States of America-Biological Sciences, 1982. **79**(8): p. 2554-2558.
- [15] Kuffler, S.W., *Discharge patterns and functional organization of mammalian retina*. J Neurophysiol, 1953. **16**(1): p. 37-68.
- [16] Nigrin, A., *Neural networks for pattern recognition*. 1993, Cambridge, Mass.: MIT Press. xvii, 413 p.
- [17] Reggia, J.A., et al., *A Competitive Distribution-Theory of Neocortical Dynamics*. Neural Computation, 1992. **4**(3): p. 287-317.
- [18] Duncan, J. and G.W. Humphreys, *Visual-Search and Stimulus Similarity*. Psychological Review, 1989. **96**(3): p. 433-458.
- [19] LaBerge, D., *Attention, awareness, and the triangular circuit*. Conscious Cogn, 1997. **6**(2-3): p. 149-81.
- [20] Atema, J., *Distribution of Chemical Stimuli, in Sensory biology of aquatic animals*, J. Atema, Editor. 1988, Springer-Verlag: New York. p. p 29-56 xxxvi, 936 p.
- [21] Chen, W.R., W. Xiong, and G.M. Shepherd, *Analysis of relations between NMDA receptors and GABA release at olfactory bulb reciprocal synapses*. Neuron, 2000. **25**(3): p. 625-33.
- [22] Aroniadou-Anderjaska, V., et al., *Tonic and synaptically evoked presynaptic inhibition of sensory input to the rat olfactory bulb via GABA(B) heteroreceptors*. J Neurophysiol, 2000. **84**(3): p. 1194-203.
- [23] Isaacson, J.S. and B.W. Strowbridge, *Olfactory reciprocal synapses: Dendritic signaling in the CNS*. Neuron, 1998. **20**(4): p. 749-761.
- [24] Lowe, G., *Inhibition of backpropagating action potentials in mitral cell secondary dendrites*. J Neurophysiol, 2002. **88**(1): p. 64-85.
- [25] Mcfadden, F.E., *Convergence of Competitive Activation Models Based on Virtual Lateral Inhibition*. Neural Networks, 1995. **8**(6): p. 865-875.
- [26] Mcfadden, F.E., Y. Peng, and J.A. Reggia, *Local Conditions for Phase-Transitions in Neural Networks with Variable Connection Strengths*. Neural Networks, 1993. **6**(5): p. 667-676.
- [27] Nagle, R.K., E.B. Saff, and A.D. Snider, *Fundamentals of differential equations and boundary value problems*. 5th ed. 2008, Boston: Pearson Addison-Wesley.
- [28] J. S. Sanchez, L.I.K. *Data reduction using classifier ensembles*. in Proc. 11th European Symposium on Artificial Neural Networks. 2007.
- [29] Quiroga, R.Q., et al., *Invariant visual representation by single neurons in the human brain*. Nature, 2002. **435**(7045): p. 1102-7.
- [30] Chelazzi, L., et al., *Responses of neurons in inferior temporal cortex during memory-guided visual search*. J Neurophysiol, 1998. **80**(6): p. 2918-40.

Reasoning with Prioritized Data by Aggregation of Distance Functions

Ofer ARIELI

Department of Computer Science, The Academic College of Tel-Aviv, Israel

Abstract. We introduce a general framework for reasoning with prioritized data by aggregation of distance functions, study some basic properties of the entailment relations that are obtained, and relate them to other approaches for maintaining uncertain information.

1. Introduction

Reasoning with prioritized data is at the heart of many information systems. Eminent examples for this are, e.g., database systems, where integrity constraints are superior to raw data [1,2,3], ranked knowledge-bases, where information is graded according to its reliability or accuracy [4,5,6], and (iterated) belief revision, where more recent data gets higher precedence than older one [7,8,9,10]. There is no wonder, therefore, that reasoning with prioritized information is a cornerstone of many formalisms for maintaining uncertainty, such as annotated logic [11], possibilistic logic [12], and System Z [13].

In this paper we handle prioritized data by a possible-world semantics, derived by distance considerations. To illustrate this, consider the following example:

Example 1 Let Γ be a set of formulas, consisting of the following subtheories:

$$\begin{aligned}\Gamma_1 &= \{ \text{bird}(x) \rightarrow \text{fly}(x), \text{color_of}(\text{Tweety}, \text{Red}) \}, \\ \Gamma_2 &= \{ \text{bird}(\text{Tweety}), \text{penguin}(\text{Tweety}) \}, \\ \Gamma_3 &= \{ \text{penguin}(x) \rightarrow \neg\text{fly}(x) \}.\end{aligned}$$

Intuitively, Γ is a theory with three priority levels, where precedence is given to formulas that belong to subtheories with higher indices, that is, for $1 \leq i < j \leq 3$, each formula in Γ_j is considered more important (or more reliable) than the formulas in Γ_i .

A justification for the representation above may be the following: the highest level (Γ_3) consists of integrity constraints that should not be violated. In our case, the single rule in this level specifies that a characteristic property of penguins is that they cannot fly, and there are no exceptions for that. The intermediate level (Γ_2) contains some known facts about the domain of discourse, and the lowest level (Γ_1) consists of default assumptions about this domain (in our case, a bird can fly unless otherwise stated), and facts with lower certainty.

Note that as a ‘flat’ set of assertions (i.e., when all the assertions in Γ have the same priority), this theory is classically inconsistent, therefore everything follows from it, and so the theory is useless. However, as Γ is prioritized, one would like to draw the following conclusions from it:

1. Conclude `bird(Tweety)` and `penguin(Tweety)` (but not their negations), as these facts are explicitly stated in a level that is consistent with the higher priority levels.
2. Conclude $\neg\text{fly}(\text{Tweety})$ (and do not conclude `fly(Tweety)`), as this fact follows from the two top priority levels, while its complement is inferrable by a lower level (which, moreover, is inconsistent with the higher levels).
3. Conclude `color_of(Tweety, Red)` (but not its negation), since although this fact appears in the lowest level of Γ , and that level is inconsistent with the other levels, it is not contradicted by any consistent fragment of Γ , so there is no reason to believe that `color_of(Tweety, Red)` does not hold.

The kind of reasoning described above is obtained in our framework by the following two principles:

- A distance-based preference relation is defined on the space of interpretations, so that inferences are drawn according to the most preferred interpretations. In the example above, for instance, interpretations in which `fly(Tweety)` is false will be ‘closer’ to Γ , and so more plausible, than interpretations in which `fly(Tweety)` is true, (thus item (2) above is obtained).
- Priorities are considered as extra-logical data that is exploited by an iterative process that first computes interpretations that are as close as possible to higher-levelled subtheories, and then makes preference among those interpretations according to their closeness to lower-levelled subtheories.

The goal of our work is to examine these principles and to show that they reflect a variety of methods for maintaining imprecise information.¹

2. Distance-based Entailments for Prioritized Theories

Let \mathcal{L} be a propositional language with a finite set `Atoms` of atomic formulas. The space of all the two-valued interpretations on `Atoms` is denoted by Λ_{Atoms} . A *theory* Γ in \mathcal{L} is a (possibly empty) finite multiset of formulas in \mathcal{L} . The set of atomic formulas that occur in the formulas of Γ is denoted `Atoms`(Γ) and the set of models of Γ (that is, the interpretations on `Atoms`(Γ) in which every formula in Γ is true) is denoted `mod`(Γ).

Definition 2 An *n-prioritized theory* is a theory $\Gamma^{\langle n \rangle}$ in \mathcal{L} , partitioned into $n \geq 1$ pairwise disjoint sub-theories Γ_i ($1 \leq i \leq n$). Notation: $\Gamma^{\langle n \rangle} = \Gamma_1 \oplus \Gamma_2 \oplus \dots \oplus \Gamma_n$.

In what follows we shall usually write Γ instead of $\Gamma^{\langle n \rangle}$. Intuitively, formulas in higher levels are preferred over those in lower levels, so if $1 \leq i < j \leq n$ then a formula $\psi \in \Gamma_j$ overtakes any formula $\phi \in \Gamma_i$. Note that in this writing the precedence is *righthand increasing*.

¹Due to a lack of space some proofs are reduced or omitted.

Definition 3 Let $\Gamma = \Gamma_1 \oplus \dots \oplus \Gamma_n$ be an n -prioritized theory.

- For any $1 \leq i \leq n$, denote the $n - i + 1$ highest levels of Γ by $\Gamma_{\geq i}$, that is, $\Gamma_{\geq i} = \Gamma_i \oplus \dots \oplus \Gamma_n$.
- Denote by $\bar{\Gamma}_{\geq i}$ the ‘flat’ (1-prioritized) theory, obtained by taking the union of the priority levels in $\Gamma_{\geq i}$, that is, $\bar{\Gamma}_{\geq i} = \Gamma_i \cup \dots \cup \Gamma_n$. Also, $\bar{\Gamma} = \bar{\Gamma}_{\geq 1}$.
- The *consistency level* con of Γ is the minimal value $i \leq n$ such that $\bar{\Gamma}_{\geq i}$ is consistent. If there is no such value, let $\text{con} = n + 1$ (and then $\Gamma_{\geq \text{con}} = \emptyset$).

Definition 4 A total function $d: U \times U \rightarrow \mathbb{R}^+$ is called *pseudo distance* on U if it satisfies the following two properties:

Symmetry: $\forall u, v \in U \ d(u, v) = d(v, u)$.

Identity Preservation: $\forall u, v \in U \ d(u, v) = 0$ iff $u = v$.

A *distance function* on U is a pseudo distance on U with the following property:

Triangulation: $\forall u, v, w \in U \ d(u, v) \leq d(u, w) + d(w, v)$.

Example 5 The following two functions are distances on Λ_{Atoms} .

- *The drastic distance*: $d_U(\nu, \mu) = 0$ if $\nu = \mu$ and $d_U(\nu, \mu) = 1$ otherwise.
- *The Hamming distance*: $d_H(\nu, \mu) = |\{p \in \text{Atoms} \mid \nu(p) \neq \mu(p)\}|$.²

Definition 6 A *numeric aggregation function* f is a total function that accepts a multiset of real numbers and returns a real number. In addition, f is non-decreasing in the values of its argument³, $f(\{x_1, \dots, x_n\}) = 0$ iff $x_1 = \dots = x_n = 0$, and $\forall x \in \mathbb{R} \ f(\{x\}) = x$.

Definition 7 An aggregation function is called *hereditary*, if $f(\{x_1, \dots, x_n\}) < f(\{y_1, \dots, y_n\})$ implies that $f(\{x_1, \dots, x_n, z_1, \dots, z_m\}) < f(\{y_1, \dots, y_n, z_1, \dots, z_m\})$.

In the sequel, we shall apply aggregation functions to distance values. As distances are non-negative, summation, average, and maximum are all aggregation functions. Note, however, that while summation and average are hereditary, the maximum function is not.

Definition 8 A pair $\mathcal{P} = \langle d, f \rangle$, where d is a pseudo distance and f is an aggregation function, is called a (distance-based) *preferential setting*. Given a theory $\Gamma = \{\psi_1, \dots, \psi_n\}$, an interpretation ν , and a preferential setting $\langle d, f \rangle$, define:

- $d(\nu, \psi_i) = \min\{d(\nu, \mu) \mid \mu \in \text{mod}(\psi_i)\}$,⁴
- $\delta_{d,f}(\nu, \Gamma) = f(\{d(\nu, \psi_1), \dots, d(\nu, \psi_n)\})$.

²That is, $d_H(\nu, \mu)$ is the number of atoms p such that $\nu(p) \neq \mu(p)$. This function is also known as the Dalal distance [14].

³I.e., the function does not decrease when a multiset element is replaced by a bigger one.

⁴Below, we exclude classical contradictions from a theory. Alternatively, if ψ is not satisfiable, one may let $d(\nu, \psi) = 1 + \max\{d(\nu, \mu) \mid \nu, \mu \in \Lambda_{\text{Atoms}}\}$.

Definition 9 A (pseudo) distance d is *unbiased*, if for every formula ψ and interpretations ν_1, ν_2 , if $\nu_1(p) = \nu_2(p)$ for every $p \in \text{Atoms}(\psi)$, then $d(\nu_1, \psi) = d(\nu_2, \psi)$.

The last property assures that the ‘distance’ between an interpretation and a formula is independent of irrelevant atoms (those that do not appear in the formula). Note, e.g., that the distances in Example 5 are unbiased.

For a preferential setting $\mathcal{P} = \langle d, f \rangle$ we now define an operator $\Delta_{\mathcal{P}}$ that introduces, for every n -prioritized theory Γ , its ‘most plausible’ interpretations, namely: the interpretations that are $\delta_{d,f}$ -closest to Γ .

Definition 10 Let $\mathcal{P} = \langle d, f \rangle$ be a preferential setting. For an n -prioritized theory $\Gamma = \Gamma_1 \oplus \Gamma_2 \oplus \dots \oplus \Gamma_n$ consider the following n sets of interpretations:

- $\Delta_{\mathcal{P}}^n(\Gamma) = \{\nu \in \Lambda_{\text{Atoms}} \mid \forall \mu \in \Lambda_{\text{Atoms}} \delta_{d,f}(\nu, \Gamma_n) \leq \delta_{d,f}(\mu, \Gamma_n)\},$
- $\Delta_{\mathcal{P}}^{n-i}(\Gamma) = \{\nu \in \Delta_{\mathcal{P}}^{n-i+1}(\Gamma) \mid \forall \mu \in \Delta_{\mathcal{P}}^{n-i+1}(\Gamma) \delta_{d,f}(\nu, \Gamma_{n-i}) \leq \delta_{d,f}(\mu, \Gamma_{n-i})\}$
for every $1 \leq i < n$.

The sequence $\Delta_{\mathcal{P}}^n(\Gamma), \dots, \Delta_{\mathcal{P}}^1(\Gamma)$ is clearly non-increasing, as sets with smaller indices are subsets of those with bigger indices. This reflects the intuitive idea that higher-levelled formulas are preferred over lower-levelled formulas, thus the interpretations of the latter are determined by the interpretations of the former. Since the relevant interpretations are derived by distance considerations, each set in the sequence above contains the interpretations that are $\delta_{d,f}$ -closest to the corresponding subtheory among the elements of the preceding set in the sequence.

Denote by $\Delta_{\mathcal{P}}(\Gamma)$ the last set obtained by this sequence (that is, $\Delta_{\mathcal{P}}(\Gamma) = \Delta_{\mathcal{P}}^1(\Gamma)$). The elements of $\Delta_{\mathcal{P}}(\Gamma)$ are the *most plausible interpretations* of Γ . These are the interpretations according to which the Γ -conclusions are drawn:

Definition 11 Let $\mathcal{P} = \langle d, f \rangle$ be a preferential setting. A formula ψ follows from an (n -prioritized) theory Γ , if every interpretation in $\Delta_{\mathcal{P}}(\Gamma)$ satisfies ψ (That is, if $\Delta_{\mathcal{P}}(\Gamma) \subseteq \text{mod}(\psi)$). We denote this by $\Gamma \models_{\mathcal{P}} \psi$.

Example 12 Consider again Example 1, and let $\mathcal{P} = \langle d_H, \Sigma \rangle$. Then:

$$\begin{array}{ll} \Gamma \models_{\mathcal{P}} \text{bird(Tweety)}, & \Gamma \models_{\mathcal{P}} \text{penguin(Tweety)}, \\ \Gamma \models_{\mathcal{P}} \text{color_of(Tweety, Red)}, & \Gamma \models_{\mathcal{P}} \neg \text{fly(Tweety)}, \\ \Gamma \not\models_{\mathcal{P}} \neg \text{bird(Tweety)}, & \Gamma \not\models_{\mathcal{P}} \neg \text{penguin(Tweety)}, \\ \Gamma \not\models_{\mathcal{P}} \neg \text{color_of(Tweety, Red)}, & \Gamma \not\models_{\mathcal{P}} \text{fly(Tweety)}, \end{array}$$

as intuitively expected. In fact, using the results in the next section, one can show that the conclusions regarding bird(Tweety) , penguin(Tweety) , and fly(Tweety) hold in *every* setting (Proposition 19); The conclusions about the color of Tweety hold whenever d is unbiased and f is hereditary (see Proposition 24 and Note 25).

Note 13 The entailment relations in Proposition 11 generalize some other settings considered in the literature. For instance, $\models_{\langle d_U, \Sigma \rangle}$ corresponds to the merging operator in [15] (see also [8]). Also, if every Γ_i in Γ is a singleton, the iterated process of computing distances with respect to the prioritized subtheories is actually a linear revision in the sense of [16]. Other related formalisms are considered in Section 4.

3. Reasoning with $\models_{\mathcal{P}}$

In this section, we consider some basic properties of $\models_{\mathcal{P}}$. First, we examine ‘flat’ theories, that is: multisets in which all the assertions have the same priority. Proposition 16 recalls the main characteristics of reasoning with such theories.

Definition 14 Denote by \models the standard classical entailment, that is: $\Gamma \models \psi$ if every model of Γ is a model of ψ .

Definition 15 Two sets of formulas Γ_1 and Γ_2 are called *independent* (or disjoint), if $\text{Atoms}(\Gamma') \cap \text{Atoms}(\Gamma'') = \emptyset$. Two independent theories Γ_1 and Γ_2 are a *partition* of a theory Γ , if $\Gamma = \Gamma_1 \cup \Gamma_2$.

Proposition 16 [17] *Let $\mathcal{P} = \langle d, f \rangle$ be a preferential setting and Γ a 1-prioritized theory. Then:*

- $\models_{\mathcal{P}}$ is the same as the classical entailment with respect to consistent premises: if Γ is consistent, then for every ψ , $\Gamma \models_{\mathcal{P}} \psi$ iff $\Gamma \models \psi$.
- $\models_{\mathcal{P}}$ is weakly paraconsistent: inconsistent premises do not entail every formula (alternatively, for every Γ there is a formula ψ such that $\Gamma \not\models_{\mathcal{P}} \psi$).
- $\models_{\mathcal{P}}$ is non-monotonic: the set of the $\models_{\mathcal{P}}$ -conclusions does not monotonically grow in the size of the premises.

If d is unbiased, then

- $\models_{\mathcal{P}}$ is paraconsistent: if ψ is independent of Γ then $\Gamma \not\models_{\mathcal{P}} \psi$.

If, in addition, f is hereditary, then

- $\models_{\mathcal{P}}$ is rationally monotonic [18]: if $\Gamma \models_{\mathcal{P}} \psi$ and ϕ is independent of $\Gamma \cup \{\psi\}$, then $\Gamma, \phi \models_{\mathcal{P}} \psi$.
- $\models_{\mathcal{P}}$ is adaptive [19,20]: if $\{\Gamma_1, \Gamma_2\}$ is a partition of Γ , and Γ_1 is classically consistent, then for every formula ψ that is independent of Γ_2 , if $\Gamma_1 \models \psi$ then $\Gamma \models_{\mathcal{P}} \psi$.

The arrangement of the premises in a stratified structure of priority levels allows to refine and generalize the results above. As a trivial example, it is clear that the 1-prioritized theory $\{p, \neg p\}$ is totally different than the 2-prioritized theory $\{p\} \oplus \{\neg p\}$, as in the latter the symmetry between p and $\neg p$ breaks up.

In the rest of this section we examine how preferences determine the set of conclusions. The first, trivial observation, is that even if the set of premises is not consistent, the set of its $\models_{\mathcal{P}}$ -conclusions remains classically consistent:

Proposition 17 *For every setting \mathcal{P} , prioritized theory Γ , and formula ψ , if $\Gamma \models_{\mathcal{P}} \psi$ then $\Gamma \not\models_{\mathcal{P}} \neg\psi$.*

Proof. Otherwise, $\Delta_{\mathcal{P}}(\Gamma) \subseteq \text{mod}(\psi)$ and $\Delta_{\mathcal{P}}(\Gamma) \subseteq \text{mod}(\neg\psi)$. Since $\text{mod}(\psi) \cap \text{mod}(\neg\psi) = \emptyset$, we get a contradiction to the fact that $\Delta_{\mathcal{P}}(\Gamma) \neq \emptyset$ (as Λ_{Atoms} is finite, there are always interpretations that are minimally $\delta_{d,f}$ -distant from Γ). \square

Another clear characteristic of $\models_{\mathcal{P}}$ is that priorities do have a primary role in the reasoning process; conclusions of higher levelled observations remain valid when the theory is augmented with lower-levelled observations.⁵

Proposition 18 *Let Γ be an n -prioritized theory. Then for every $1 \leq i < j \leq n$, if $\Gamma_{\geq j} \models_{\mathcal{P}} \psi$ then $\Gamma_{\geq i} \models_{\mathcal{P}} \psi$.*

Proof. If $\Gamma_{\geq j} \models_{\mathcal{P}} \psi$ then $\Delta_{\mathcal{P}}^j(\Gamma) \subseteq \text{mod}(\psi)$. But $\Delta_{\mathcal{P}}^i(\Gamma) \subseteq \Delta_{\mathcal{P}}^j(\Gamma)$, and so $\Delta_{\mathcal{P}}^i(\Gamma) \subseteq \text{mod}(\psi)$ as well. Thus, $\Gamma_{\geq i} \models_{\mathcal{P}} \psi$. \square

Proposition 18 implies, in particular, that anything that follows from a subtheory that consists of the higher levels of a prioritized theory, also follows from the whole theory. Next we show that when the subtheory of the higher levels is classically consistent, we can say more than that: anything that can be *classically inferred* from the highest consistent levels of a prioritized theory is also deducible from the whole theory (even when lower-levelled subtheories imply the converse). To see this we suppose, then, that at least the most preferred level of Γ is classically consistent (that is, $\text{con} \leq n$).

Proposition 19 *For every setting $\mathcal{P} = \langle d, f \rangle$ and for every n -prioritized theory Γ with a consistency level $\text{con} \leq n$, if $\bar{\Gamma}_{\geq \text{con}} \models \psi$ then $\Gamma \models_{\mathcal{P}} \psi$.*

Proof (outline). Note, first, that for every preferential setting $\mathcal{P} = \langle d, f \rangle$ and n -prioritized theory Γ with $\text{con} \leq n$, $\Delta_{\mathcal{P}}(\Gamma_{\geq \text{con}}) = \text{mod}(\bar{\Gamma}_{\geq \text{con}})$. By the definition of $\Delta_{\mathcal{P}}$, then, $\Delta_{\mathcal{P}}(\Gamma) \subseteq \Delta_{\mathcal{P}}(\Gamma_{\geq \text{con}}) = \text{mod}(\bar{\Gamma}_{\geq \text{con}})$. Now, if $\bar{\Gamma}_{\geq \text{con}} \models \psi$, then ψ is true in every element of $\text{mod}(\bar{\Gamma}_{\geq \text{con}})$, and so ψ holds in every element of $\Delta_{\mathcal{P}}(\Gamma)$. Thus $\Gamma \models_{\mathcal{P}} \psi$. \square

Note 20 Consider again the three-levelled theory of Example 1. Proposition 19 guarantees the satisfaction of the first two items discussed in that example (the third item is considered in Note 25 below).

Proposition 21 *For every setting $\mathcal{P} = \langle d, f \rangle$ and n -prioritized theory Γ with $\text{con} \leq n$, we have that $\Gamma_{\geq \text{con}} \models_{\mathcal{P}} \psi$ iff $\bar{\Gamma}_{\geq \text{con}} \models \psi$.*

Proof (outline). Follows from the fact that for every n -prioritized theory Γ with $\text{con} \leq n$ it holds that $\Delta_{\mathcal{P}}(\Gamma_{\geq \text{con}}) = \text{mod}(\bar{\Gamma}_{\geq \text{con}})$. \square

In particular, then, $\models_{\mathcal{P}}$ coincides with the classical entailment with respect to consistent sets of premises:

Corollary 22 *If $\bar{\Gamma}$ is consistent, then $\Gamma \models_{\mathcal{P}} \psi$ iff $\bar{\Gamma} \models \psi$.*

Proof. By Proposition 21, since if $\bar{\Gamma}$ is consistent then $\text{con} = 1$, and so $\Gamma_{\geq \text{con}} = \Gamma$ and $\bar{\Gamma}_{\geq \text{con}} = \bar{\Gamma}$. \square

In the general case, we have the following relation between $\models_{\mathcal{P}}$ and \models :

⁵In [8] this is called ‘the principle of prioritized monotonicity’.

Corollary 23 If $\Gamma \models_{\mathcal{P}} \psi$ then $\bar{\Gamma} \models \psi$.

Proof. If $\bar{\Gamma}$ is consistent then by Corollary 22 $\Gamma \models_{\mathcal{P}} \psi$ iff $\bar{\Gamma} \models \psi$. If $\bar{\Gamma}$ is not classically consistent, then for *every* formula ψ , $\bar{\Gamma} \models \psi$. \square

Next we show that in many cases we can go beyond the result of Propositions 18 and 19: Not only that one may deduce from the whole theory everything that is included in its highest levels, but also lower-levelled assertions are deducible from the whole theory, provided that no higher-levelled information contradicts them. This shows that our formalism avoids the so called *drowning effect*, that is: formulas with low priority are not inhibited just due to the fact that the information at higher levels is contradictory. Prevention of the grounding effect is very important, e.g., in the context of belief revision, as it implies that anything that has no relation to the new information need not be revised.

Proposition 24 Let $\mathcal{P} = \langle d, f \rangle$ be a setting where d is non-biased and f is hereditary. If a prioritized theory Γ can be partitioned to a consistent theory Γ' and a (possible inconsistent) theory Γ'' , then $\Gamma \models_{\mathcal{P}} \psi$ for every $\psi \in \Gamma'$.

Note 25 If $\Gamma' \subseteq \Gamma_{\geq \text{con}}$, then Proposition 24 is a straightforward consequence of Proposition 19. Yet, Proposition 24 is useful in cases where Γ' contains elements that are *below* the inconsistency level of Γ , and then the claim assures that the drowning effect is not imposed on these elements. Tweety dilemma Γ , considered in Example 1, is a good example for this. It can be partitioned to $\Gamma' = \{\text{color_of(Tweety, Red)}\}$ and $\Gamma'' = \Gamma \setminus \Gamma'$. In this representation the conditions of Proposition 24 are satisfied for every preferential setting $\mathcal{P} = \langle d, f \rangle$ where d is unbiased and f is hereditary. In this case, then, $\Gamma \models_{\mathcal{P}} \text{color_of(Tweety, Red)}$, as indeed suggested in the third item of Example 1. Note, however, that $\Gamma \not\models_{d_{U,\max}} \text{color_of(Tweety, Red)}$, which shows that the condition in Proposition 24, that the aggregation function should be hereditary, is indeed necessary.

Example 26 According to the possibilistic revision operator introduced in [5,6], a formula ψ is a consequence of a prioritized (possibilistic) theory Γ if it follows from all the formulas above the consistency level of Γ . In our notations, then, ψ follows from Γ iff $\bar{\Gamma}_{\geq \text{con}} \models \psi$,⁶ and so this formalism has the drowning effect, which prevents the drawing of any conclusion that resides below the consistency level. In other formalisms for handling prioritized theories, such as those in [4,21,22], the drowning effect is avoided by using a similar policy as ours, namely: the elements of the revised theory are constructed in a stepwise manner, starting with the highest priority level and selecting from each level as many formulas as possible without violating consistency (see also [8]).

4. Related Areas and Applications

In this section we consider in greater detail two paradigms in which priorities are exploited to determine consequences.

⁶Note that by Proposition 19 this implies that every possibilistic conclusion of Γ may be inferred also by our formalisms.

4.1. Iterated Belief Revision

Belief revision, the process of changing beliefs in order to take into account new pieces of information, is perhaps closest in spirit to the basic ideas behind our framework. A widely accepted rationality criterion in this context is the success postulate that asserts that a new item of information is always accepted. In our case, this means that new data should have a higher priority over older one. Thus, assuming that Γ represents the reasoner's belief, the revised belief state in light of new information ψ may be represented by $\Gamma \oplus \{\psi\}$. Consequently, a revision by a sequence of (possibly conflicting) observations ψ_1, \dots, ψ_m may be expressed by $\Gamma \oplus \{\psi_1\} \oplus \dots \oplus \{\psi_m\}$.

The well-known set of rationality postulates, introduced in [23] by Alchourrón, Gärdenfors, and Makinson (AGM) for belief revision in the non-prioritized case, is often considered as the starting point in this area. These postulates were rephrased by Katsuno and Mendelzon [24] in terms of order relations as follows:

Proposition 27 *Let Γ be a set of formulas in a propositional language \mathcal{L} . A revision operator \circ satisfies the AGM postulates if and only if there is a faithful order \leq_Γ , such that $\text{mod}(\Gamma \circ \psi) = \min(\text{mod}(\psi), \leq_\Gamma)$.⁷*

In light of this result, one may represent revision in our framework in terms of minimization of a preferential (ranking) order. For this, we consider the following adjustment, to the context of prioritized theories, of faithful orders.

Definition 28 Let \mathcal{P} be a preferential setting and Γ a prioritized theory. A total preorder $\leq_\Gamma^{\mathcal{P}}$ on Λ_{Atoms} is called (preferentially) *faithful*, if the following conditions are satisfied:

1. If $\nu, \mu \in \Delta_{\mathcal{P}}(\Gamma)$ then $\nu <_\Gamma^{\mathcal{P}} \mu$ does not hold.
2. If $\nu \in \Delta_{\mathcal{P}}(\Gamma)$ and $\mu \notin \Delta_{\mathcal{P}}(\Gamma)$ then $\nu <_\Gamma^{\mathcal{P}} \mu$.

Proposition 29 *A preferential setting \mathcal{P} is characterized by faithful orders: For every prioritized theory Γ there is a faithful order $\leq_\Gamma^{\mathcal{P}}$ (depending on \mathcal{P} and Γ), such that $\Delta_{\mathcal{P}}(\Gamma) = \{\nu \in \Lambda_{\text{Atoms}} \mid \forall \mu \in \Lambda_{\text{Atoms}} \nu \leq_\Gamma^{\mathcal{P}} \mu\}$.*

Proposition 30 *Let \mathcal{P} be a preferential setting and Γ a prioritized theory on \mathcal{L} . Then there is a faithful order $\leq_\Gamma^{\mathcal{P}}$ (depending on \mathcal{P} and Γ), such that, for every non-contradictory formula ψ in \mathcal{L} , $\Delta_{\mathcal{P}}(\Gamma \oplus \psi) = \min(\text{mod}(\psi), \leq_\Gamma^{\mathcal{P}})$.*

In terms of entailments, the last two propositions may be rewritten as follows:

Corollary 31 *Let \mathcal{P} be a preferential setting, Γ a prioritized theory, and ψ a non-contradictory formula in \mathcal{L} . Then there is a faithful order $\leq_\Gamma^{\mathcal{P}}$, such that, for every formula ϕ in \mathcal{L} ,*

1. $\Gamma \models_{\mathcal{P}} \phi$ iff ϕ is satisfied by every $\leq_\Gamma^{\mathcal{P}}$ -minimal element of Λ_{Atoms} .
2. $\Gamma \oplus \psi \models_{\mathcal{P}} \phi$ iff ϕ is satisfied by every $\leq_\Gamma^{\mathcal{P}}$ -minimal element of $\text{mod}(\psi)$.

⁷The reader is referred, e.g., to [7,9] for detailed discussions on this result and its notions.

Note 32 In [24] a belief base Γ is represented by a single formula which is the conjunction of the elements in Γ . In the prioritized setting this is, of-course, not possible, as different formulas in Γ have different priorities. Also, in [24] the faithful property is defined in terms of $mod(\Gamma)$ rather than $\Delta_{\mathcal{P}}(\Gamma)$. This distinction follows again from the fact that in the non-prioritized case the formula that represents a belief set Γ is consistent and as such it always has models, while in our case a prioritized theory $\Gamma = \bigoplus_i \Gamma_i$ is different than the ‘flat’ theory $\bigcup_i \Gamma_i$ that may not even be consistent.

Proposition 30 refers to a single revision. For successive revisions one may follow Darwiche and Pearl’s approach [7], extending the AGM postulates with four additional ones. As it turns out, three of these postulates hold in our context:

Definition 33 Denote by $\Gamma \equiv_{\mathcal{P}} \Gamma'$ that Γ and Γ' have the same $\models_{\mathcal{P}}$ -conclusions.

Proposition 34 For every preferential setting $\mathcal{P} = \langle d, f \rangle$, prioritized theory Γ , and satisfiable formulas ψ, ϕ ,

C1: If $\psi \models \phi$ then $\Gamma \oplus \{\phi\} \oplus \{\psi\} \equiv_{\mathcal{P}} \Gamma \oplus \{\psi\}$.

C3: If $\Gamma \oplus \{\psi\} \models_{\mathcal{P}} \phi$ then $\Gamma \oplus \{\phi\} \oplus \{\psi\} \models_{\mathcal{P}} \phi$.

C4: If $\Gamma \oplus \{\psi\} \not\models_{\mathcal{P}} \neg\phi$ then $\Gamma \oplus \{\phi\} \oplus \{\psi\} \not\models_{\mathcal{P}} \neg\phi$.

Proof. We show C1; the proof of C3 and C4 is similar. If $\psi \models \phi$ then $mod(\psi) \subseteq mod(\phi)$, which implies that $\forall \nu \in mod(\psi) d(\nu, \phi) = 0$. Thus, $\forall \nu \in \Delta_{\mathcal{P}}(\{\psi\}) d(\nu, \phi) = 0$, and so $\Delta_{\mathcal{P}}(\{\psi\}) = \Delta_{\mathcal{P}}(\{\phi\} \oplus \{\psi\}) = mod(\psi)$. It follows that $\Delta_{\mathcal{P}}(\Gamma \oplus \{\psi\}) = \Delta_{\mathcal{P}}(\Gamma \oplus \{\phi\} \oplus \{\psi\})$, and therefore $\Gamma \oplus \{\phi\} \oplus \{\psi\} \equiv_{\mathcal{P}} \Gamma \oplus \{\psi\}$. \square

The forth postulate in [7], namely

C2: If $\psi \models \neg\phi$ then $\Gamma \oplus \{\phi\} \oplus \{\psi\} \equiv_{\mathcal{P}} \Gamma \oplus \{\psi\}$

is the most controversial one (see, e.g., [3,9]) and indeed in our framework it is falsified. To see this, let $\Gamma = \emptyset$, $\psi = p$, $\phi = \neg p \wedge \neg q$, and $\mathcal{P} = \langle d_H, f \rangle$ for arbitrary aggregation function f .⁸ Clearly, $\psi \models \neg\phi$. However, as $\Delta_{\mathcal{P}}(\{\psi\})$ consists of interpretations that assign t to p regardless of their assignments to q , while the interpretations in $\Delta_{\mathcal{P}}(\{\phi\} \oplus \{\psi\})$ assign t to p and f to q , it follows that $\{\phi\} \oplus \{\psi\}$ and $\{\psi\}$ are *not* $\models_{\mathcal{P}}$ -equivalent.

4.2. Prioritized Integration of Independent Data Sources

Information systems often have to incorporate *several* sources with possibly different preferences. In this section we show how this can be done in our framework. For this, we use two types of distance aggregations: *internal aggregations*, for prioritizing different formulas in the same theory, and *external aggregations*, for prioritizing different theories. As internal and external aggregations may reflect different kinds of considerations, they are represented by two different aggrega-

⁸ f is irrelevant here since each priority level is a singleton.

tion functions, denoted f and g , respectively. Now, using the terminology and the notations of the previous sections, we can think of the underlying n -prioritized theory as follows:

$$\Gamma = \{\Gamma_1^1, \dots, \Gamma_{k_1}^1\} \oplus \dots \oplus \{\Gamma_1^n, \dots, \Gamma_{k_n}^n\}, \quad (1)$$

where now each Γ_j^i is a different theory, theories with the same superscript have the same precedence, and Γ^i is preferred over Γ^j iff $i > j$. This can be formalized by the following generalizations of Definitions 8 and 10:

Definition 35 An *extended preferential setting* is a triple $\mathcal{E} = \langle d, f, g \rangle$, where d is a pseudo distance and f, g are aggregation functions. Given an n -prioritized theory $\Gamma = \{\Gamma_1^1, \dots, \Gamma_{k_1}^1\} \oplus \dots \oplus \{\Gamma_1^n, \dots, \Gamma_{k_n}^n\}$ and an interpretation ν , define for every $1 \leq i \leq n$ and $1 \leq j \leq k_i$ the value of $\delta_{d,f}(\nu, \Gamma_j^i)$ just as in Definitions 8. Also, let

$$\delta_{\mathcal{E}}(\nu, \overline{\Gamma^i}) = \delta_{d,f,g}(\nu, \overline{\Gamma^i}) = g(\{\delta_{d,f}(\nu, \Gamma_1^i), \dots, \delta_{d,f}(\nu, \Gamma_{k_i}^i)\}).$$

Definition 36 Let $\mathcal{E} = \langle d, f, g \rangle$ be an extended preferential setting. Given an n -prioritized theory $\Gamma = \{\Gamma_1^1, \dots, \Gamma_{k_1}^1\} \oplus \dots \oplus \{\Gamma_1^n, \dots, \Gamma_{k_n}^n\}$, consider the following n sets of interpretations:

- $\Delta_{\mathcal{E}}^n(\Gamma) = \{\nu \mid \forall \mu \delta_{\mathcal{E}}(\nu, \overline{\Gamma^n}) \leq \delta_{\mathcal{E}}(\mu, \overline{\Gamma^n})\},$
- $\Delta_{\mathcal{E}}^{n-i}(\Gamma) = \{\nu \in \Delta_{\mathcal{E}}^{n-i+1}(\Gamma) \mid \forall \mu \in \Delta_{\mathcal{E}}^{n-i+1}(\Gamma) \delta_{\mathcal{E}}(\nu, \overline{\Gamma^{n-i}}) \leq \delta_{\mathcal{E}}(\mu, \overline{\Gamma^{n-i}})\}$
for every $1 \leq i < n$.

The *most plausible interpretations* of Γ (with respect to d, f, g) are the interpretations in $\Delta_{\mathcal{E}}^1(\Gamma)$ (henceforth denoted by $\Delta_{\mathcal{E}}(\Gamma)$).

The corresponding consequence relations are now defined as follows:

Definition 37 Let $\mathcal{E} = \langle d, f, g \rangle$ be an extended preferential setting. A formula ψ follows from an n -prioritized theory Γ if every interpretation in $\Delta_{\mathcal{E}}(\Gamma)$ satisfies ψ . We denote this by $\Gamma \models_{\mathcal{E}} \psi$.

Clearly, Definition 37 generalizes Definition 11 in the sense that if in (1) above $k_i = 1$ for every $1 \leq i \leq n$, then for every g , $\models_{\mathcal{E}}$ (in the sense of Definition 37) is the same as $\models_{\mathcal{P}}$ (in the sense of Definition 11).⁹

Example: Constraint-Based Merging of Prioritized Data-Sources

Consider the following scenario regarding speculations on the stock exchange (see also [3]). An investor consults with four financial experts about their opinions regarding four different shares, denoted s_1, s_2, s_3 and s_4 . The opinion of expert i is represented by a theory (data-source) Γ_i . Suppose that $\Gamma_1 = \Gamma_2 = \{s_1, s_2, s_3\}$, $\Gamma_3 = \{\neg s_1, \neg s_2, \neg s_3, \neg s_4\}$, and $\Gamma_4 = \{s_1, s_2, \neg s_4\}$. Thus, for instance, expert 4 suggests to buy shares s_1 and s_2 , doesn't recommend to buy share s_4 , and doesn't have an opinion about s_3 .

⁹Alternatively, $\models_{\mathcal{E}}$ coincides with $\models_{\mathcal{P}}$ if in (1) each T_j^i is a singleton and $g = f$.

Suppose, in addition, that the investor has his own restrictions about the investment policy. For instance, if some share, say s_4 , is considered risky, buying it may be balanced by purchasing at least two out of the three other shares, and vice-versa. This may be represented by the following integrity constraint: $\mathcal{IC} = \{s_4 \longleftrightarrow ((s_1 \wedge s_2) \vee (s_2 \wedge s_3) \vee (s_1 \wedge s_3))\}$. Assuming that all the experts are equally faithful, their suggestions may be represented by the 2-prioritized theory $\Gamma = \{\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4\} \oplus \{\mathcal{IC}\}$, in which the investor's constraint about the purchasing policy is of higher precedence than the experts' opinions. For the extended setting $\langle d_U, \Sigma, \Sigma \rangle$ we get that the most plausible interpretations of Γ are the elements of the following set:

$$\Delta_{d_U, \Sigma, \Sigma}(\Gamma) = \{\nu \in \text{mod}(\mathcal{IC}) \mid \forall \mu \in \text{mod}(\mathcal{IC}) \\ \delta_{d_U, \Sigma, \Sigma}(\nu, \{\Gamma_i \mid 1 \leq i \leq 4\}) \leq \delta_{d_U, \Sigma, \Sigma}(\mu, \{\Gamma_i \mid 1 \leq i \leq 4\})\}.$$

The models of \mathcal{IC} and their distances to $\bar{\Gamma} = \{\Gamma_1, \dots, \Gamma_4\}$ are given below.

	s_1	s_2	s_3	s_4	$\delta_{d_U, \Sigma, \Sigma}(\nu_i, \bar{\Gamma})$		s_1	s_2	s_3	s_4	$\delta_{d_U, \Sigma, \Sigma}(\nu_i, \bar{\Gamma})$
ν_1	t	t	t	t	5		f	t	t	t	7
ν_2	t	t	f	t	7		f	t	f	f	6
ν_3	t	f	t	t	7		f	f	t	f	6
ν_4	t	f	f	f	7		f	f	f	f	8

Thus, $\Delta_{d_U, \Sigma, \Sigma}(\Gamma) = \{\nu_1\}$, and so the investor will purchase all the four shares.

Clearly, the experts could have different reputations, and this may affect the investor's decision. For instance, assuming that expert 4 has a better reputation than the other experts, his or her opinion may get a higher precedence, yielding the following 3-prioritized theory: $\Gamma' = \{\Gamma_1, \Gamma_2, \Gamma_3\} \oplus \{\Gamma_4\} \oplus \{\mathcal{IC}\}$. It is interesting to note that in this case the recommendation of the most significant expert (number 4) does not comply with the investor's restriction.

By using the same setting as before ($d = d_U$, $f = g = \Sigma$), the investor ends up with a different investment policy, according to the following table:

	s_1	s_2	s_3	s_4	$\delta_{d_U, \Sigma, \Sigma}(\nu_i, \Gamma_4)$	$\delta_{d_U, \Sigma, \Sigma}(\nu_i, \{\Gamma_1, \Gamma_2, \Gamma_3\})$
ν_1	t	t	t	t	1	$0+0+4 = 4$
ν_2	t	t	f	t	1	$1+1+3 = 5$
ν_3	t	f	t	t	2	N.A.
ν_4	t	f	f	f	1	$1+1+1 = 3$
ν_5	f	t	t	t	2	N.A.
ν_6	f	t	f	f	1	$1+1+1 = 3$
ν_7	f	f	t	f	2	N.A.
ν_8	f	f	f	f	2	N.A.

Here, $\Delta_{d_U, \Sigma, \Sigma}(\Gamma') = \{\nu_4, \nu_6\}$, and the decision would be to purchase either s_1 or s_2 , but not both, which seems as a 'fair balance' between the investor's restriction and the recommendation of the most significant expert (taking into account also the other recommendations).

References

- [1] O. Arieli, M. Denecker, B. Van Nuffelen, and M. Bruynooghe. Coherent integration of databases by abductive logic programming. *Artificial Intelligence Research*, 21:245–286, 2004.
- [2] A. Calì, D. Calvanese, G. De Giacomo, and M. Lanzerini. Data integration under integrity constraints. *Information Systems*, 29(2):147–163, 2004.
- [3] S. Konieczny and R. Pino Pérez. Merging information under constraints: a logical framework. *Logic and Computation*, 12(5):773–808, 2002.
- [4] O. Arieli. Four-valued logics for reasoning with uncertainty in prioritized data. In B. Bouchon-Meunier, R. Yager, and L. Zadeh, editors, *Information, Uncertainty, Fusion*, pages 293–304. Kluwer, 1999.
- [5] S. Benferhat, C. Cayrol, D. Dubois, J. Lang, and H. Prade. Inconsistency management and prioritized syntax-based entailment. In *Proc. IJCAI'93*, pages 640–645, 1993.
- [6] S. Benferhat, D. Dubois, and H. Prade. How to infer from inconsistent beliefs without revising? In *Proc. IJCAI'95*, pages 1449–1455, 1995.
- [7] A. Darwiche and J. Pearl. On the logic of iterated belief revision. *Artificial Intelligence*, 89:1–29, 1997.
- [8] J. Delgrande, D. Dubois, and J. Lang. Iterated revision and prioritized merging. In *Proc. KR'06*, pages 210–220, 2006.
- [9] Y. Jin and M. Thielscher. Iterated revision, revised. *Artificial Intelligence*, 171:1–18, 2007.
- [10] M. Williams. Iterated theory base change: A computational model. In *Proc. IJCAI'95*, pages 1541–1547, 1995.
- [11] V. S. Subrahmanian. Mechanical proof procedures for many valued lattice-based logic programming. *Journal of Non-Classical Logic*, 7:7–41, 1990.
- [12] D. Dubois, J. Lang, and H. Prade. Possibilistic logic. In D. Gabbay, C. Hogger, and J. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 3, pages 439–513. 1994.
- [13] M. Goldszmidt and J. Pearl. Qualitative probabilities for default reasoning, belief revision, and causal modeling. *Artificial Intelligence*, 84:57–112, 1996.
- [14] M. Dalal. Investigations into a theory of knowledge base revision. In *Proc. AAAI'88*, pages 475–479. AAAI Press, 1988.
- [15] D. Lehmann. Another perspective on default reasoning. *Annals of Mathematics and Artificial Intelligence*, 15(1):61–82, 1995.
- [16] B. Nebel. Base revision operations and schemes: Semantics, representation, and complexity. In *Proc. ECAI'94*, pages 341–345, 1994.
- [17] O. Arieli. Distance-based paraconsistent logics. *International Journal of Approximate Reasoning*, 2008. Accepted.
- [18] D. Lehmann and M. Magidor. What does a conditional knowledge base entail? *Artificial Intelligence*, 55:1–60, 1992.
- [19] D. Batens. Dynamic dialectical logics. In G. Priest, R. Routley, and J. Norman, editors, *Paraconsistent Logic. Essay on the Inconsistent*, pages 187–217. Philosophia Verlag, 1989.
- [20] D. Batens. Inconsistency-adaptive logics. In E. Orlowska, editor, *Logic at Work*, pages 445–472. Physica Verlag, 1998.
- [21] G. Brewka. Preferred subtheories: an extended logical framework for default reasoning. In *Proc. IJCAI'89*, pages 1043–1048, 1989.
- [22] B. Nebel. Belief revision and default reasoning: Syntax-based approaches. In *Proc. KR'01*, pages 417–428, 1991.
- [23] C. E. Alchourrón, P. Gärdenfors, and D. Makinson. On the logic of theory change: Partial meet contraction and revision function. *Journal of Symbolic Logic*, 50:510–530, 1985.
- [24] Katsumo H. and A. O. Mendelzon. Propositional knowledge base revision and minimal change. *Artificial Intelligence*, 52:263–294, 1991.

Distance-Based Non-Deterministic Semantics

Ofer ARIELI^a, Anna ZAMANSKY^{b,1}

^a Department of Computer Science, The Academic College of Tel-Aviv, Israel

^b Department of Computer Science, Tel-Aviv University, Israel

Abstract. Representing uncertainty and reasoning with dynamically evolving systems are two related issues that are in the heart of many information systems. In this paper we show that these tasks can be successfully dealt with by incorporating distance semantics and non-deterministic matrices. The outcome is a general framework for capturing the principle of minimal change and providing a non-deterministic view of the domain of discourse. We investigate some properties of the entailment relations that are induced by this framework and demonstrate their usability in some test-cases.

1. Introduction

Distance-based semantics is a common technique for reflecting the principle of minimal change in different scenarios where information is dynamically evolving, such as belief revision, integration of independent data sources, and planning systems. By nature, the underlying data in such cases is often incomplete or inconsistent. Yet, this fact is not always representable in terms of standard truth functions, and so other alternatives must be looked for. One such alternative is to borrow the idea of *non-deterministic computations* from automata and computability theory. This idea leads to a quest for structures, where the value assigned by a valuation to a complex formula might be chosen non-deterministically from a certain (non-empty) set of options (see [1]). The advantage of combining distance-based semantics and non-deterministic computations is demonstrated in the following example:

Example 1 Suppose that a reasoner wants to discover some properties of an unknown Boolean function (e.g., determine a sequence of inputs for which the output is known). The reasoner may have some idea on the structure of an electronic circuit that implements the unknown function, but this information may be partial or even unreliable. Two common problems in this respect are the following:

- It might happen that the behaviour of an electronic gate in the circuit is not coherent and therefore cannot be predicted (e.g., because of the presence of disturbing noise sources on or off chip).

¹Supported by the Israel Science Foundation, grant No. 809–06.

- The reasoner may get conflicting evidence on the behaviour of the circuit from different sources (e.g., due to unreliable indicators, adversary third parties, erroneous communication among sources, etc.).

Situations like that of the first item can be handled by non-deterministic structures, and distance-based considerations are helpful to narrow the gap among contradictory sources as in the second item. However, any system aspiring to general intelligence should be able to deal with both of these types of uncertainty at the same time. For this, we introduce a general framework that combines non-determinism with distance semantics, demonstrate its usefulness by applying it to a number of case studies, and study some of the basic properties of the entailment relations that are obtained.

2. Preliminaries

2.1. Distance-Based Semantics

Distance semantics is a cornerstone behind many paradigms of handling incomplete or inconsistent information, such as belief revision [2,3,4,5] database integration systems [6,7,8,9], and social choice theory [10,11]. In [12,13] this approach is described in terms of entailment relations. The idea is simple: Given a distance function d on a space of valuations, reasoning with a given set of premises Γ is based on those valuations that are ' d -closest' to Γ (called the *most plausible valuations* of Γ). For instance, it is intuitively clear that valuations in which q is true should be closer to $\Gamma = \{p, \neg p, q\}$ than valuations in which q is false, and so q should follow from Γ while $\neg q$ should *not* follow from Γ , although Γ is not consistent. The formal details are given in [12,13] and are sketched in what follows.

Suppose that \mathcal{L} is a fixed propositional language with a finite set Atoms of atomic formulas. We denote by Γ a *finite multiset* of \mathcal{L} -formulas, for which $\text{Atoms}(\Gamma)$ and $\text{SF}(\Gamma)$ denote, respectively, the atomic formulas that occur in Γ and the subformulas of Γ . The set of models of Γ (that is, the valuations that satisfy every formula in Γ) is denoted $\text{mod}(\Gamma)$.

Definition 2 A *pseudo-distance* on a set U is a total function $d : U \times U \rightarrow \mathbb{R}^+$, satisfying the following conditions:

- *symmetry*: for all $\nu, \mu \in U$ $d(\nu, \mu) = d(\mu, \nu)$,
- *identity preservation*: for all $\nu, \mu \in U$ $d(\nu, \mu) = 0$ iff $\nu = \mu$.

A pseudo-distance d is a *distance* (metric) on U if it has the following property:

- *triangular inequality*: for all $\nu, \mu, \sigma \in U$ $d(\nu, \sigma) \leq d(\nu, \mu) + d(\mu, \sigma)$.

Example 3 It is easy to verify that the following two functions are distances on the space Λ_{Atoms} of two-valued valuations on Atoms :

- *The drastic distance*: $d_U(\nu, \mu) = 0$ if $\nu = \mu$ and $d_U(\nu, \mu) = 1$ otherwise.

- The Hamming distance: $d_H(\nu, \mu) = |\{p \in \text{Atoms} \mid \nu(p) \neq \mu(p)\}|$.

Definition 4 A *numeric aggregation function* is total function f whose argument is a multiset of real numbers and whose values are real numbers, such that: (i) f is non-decreasing in the value of its argument, (ii) $f(\{x_1, \dots, x_n\}) = 0$ iff $x_1 = x_2 = \dots = x_n = 0$, and (iii) $f(\{x\}) = x$ for every $x \in \mathbb{R}$.

Definition 5 Given a theory $\Gamma = \{\psi_1, \dots, \psi_n\}$, a two-valued valuation $\nu \in \Lambda_{\text{Atoms}}$, a pseudo-distance d and an aggregation function f , define:

- $d(\nu, \psi_i) = \begin{cases} \min\{d(\nu, \mu) \mid \mu \in \text{mod}(\psi_i)\} & \text{if } \text{mod}(\psi_i) \neq \emptyset, \\ 1 + \max\{d(\mu_1, \mu_2) \mid \mu_1, \mu_2 \in \Lambda_{\text{Atoms}}\} & \text{otherwise.} \end{cases}$
- $\delta_{d,f}(\nu, \Gamma) = f(\{d(\nu, \psi_1), \dots, d(\nu, \psi_n)\})$.

Note that in the two extreme degenerate cases, when ψ is either a tautology or a contradiction, all the valuations are equally distant from ψ . In any other case, the valuations that are closest to ψ are its models and their distance to ψ is zero. This also implies that $\delta_{d,f}(\nu, \Gamma) = 0$ iff $\nu \in \text{mod}(\Gamma)$ (see [13]).

Definition 6 The *most plausible valuations* of Γ (with respect to a pseudo distance d and an aggregation function f) are defined as follows:

$$\Delta_{d,f}(\Gamma) = \begin{cases} \{\nu \in \Lambda_{\text{Atoms}} \mid \forall \mu \in \Lambda_{\text{Atoms}} \delta_{d,f}(\nu, \Gamma) \leq \delta_{d,f}(\mu, \Gamma)\} & \text{if } \Gamma \neq \emptyset, \\ \Lambda_{\text{Atoms}} & \text{otherwise.} \end{cases}$$

Definition 7 Denote: $\Gamma \models_{d,f} \psi$, if $\Delta_{d,f}(\Gamma) \subseteq \text{mod}(\psi)$. That is, conclusions should follow from *all* of the most plausible valuations of the premises.

Example 8 Consider $\Gamma = \{p, \neg p, q\}$ together with the Hamming distance and the summation function. By classical logic, everything follows from Γ , including $\neg q$. In contrast, since $\Delta_{d_H, \Sigma}(\Gamma)$ consists only of valuations in which q is true, we have that $\Gamma \models_{d_H, \Sigma} q$ while $\Gamma \not\models_{d_H, \Sigma} \neg q$, as intuitively expected.

2.2. Non-Deterministic Matrices

According to the classical principle of assigning truth values to formulas, the truth value assigned to a complex formula is uniquely determined by the truth values of its subformulas. This approach is no longer appropriate in the real world, where incomplete, imprecise or even inconsistent information is involved. To cope with this, Avron and Lev [1] introduced the notion of *non-deterministic matrices*, in which the value assigned by a valuation to a complex formula can be chosen *non-deterministically* out of a certain nonempty set of options. Below, we recall the basic definitions behind this approach.

Definition 9 A *non-deterministic matrix* (henceforth, *Nmatrix*) for a propositional language \mathcal{L} is a tuple $\mathcal{M} = \langle \mathcal{V}, \mathcal{D}, \mathcal{O} \rangle$, where \mathcal{V} is a non-empty set of truth values, \mathcal{D} is a non-empty proper subset of \mathcal{V} , and for every n -ary connective \diamond of \mathcal{L} , \mathcal{O} includes an n -ary function $\tilde{\diamond}$ from \mathcal{V}^n to $2^\mathcal{V} - \{\emptyset\}$.

Definition 10 An \mathcal{M} -valuation is a function $\nu : \mathcal{L} \rightarrow \mathcal{V}$ that satisfies the following condition for every n -ary connective \diamond of \mathcal{L} and $\psi_1, \dots, \psi_n \in \mathcal{L}$:

$$\nu(\diamond(\psi_1, \dots, \psi_n)) \in \tilde{\diamond}(v(\psi_1), \dots, v(\psi_n)).$$

We denote by $\Lambda_{\mathcal{M}}$ the space of all the \mathcal{M} -valuations.

It is important to stress that in Nmatrices the truth-values assigned to ψ_1, \dots, ψ_n do not uniquely determine the truth-value assigned to $\diamond(\psi_1, \dots, \psi_n)$, as ν makes a non-deterministic choice out of the set of options $\tilde{\diamond}(\nu(\psi_1), \dots, \nu(\psi_n))$. Thus, the non-deterministic semantics is non-truth-functional, as opposed to the deterministic case.

Example 11 Let $\mathcal{M} = \langle \{t, f\}, \{t\}, \mathcal{O} \rangle$, where \mathcal{O} consists of the following operators:

		\neg			\wedge
			t	t	$\{t, f\}$
			t	f	$\{f\}$
t	$\{f\}$		f	t	$\{f\}$
f	$\{t\}$		f	f	$\{f\}$

Let $p, q \in \text{Atoms}$ and $\nu_1, \nu_2 \in \Lambda_{\mathcal{M}}$, such that $\nu_1(p) = \nu_2(p) = \nu_1(q) = \nu_2(q) = t$. While ν_1 and ν_2 coincide on $\neg p$ and $\neg q$, their value for $p \wedge q$ may *not* be the same.

Definition 12 A valuation $\nu \in \Lambda_{\mathcal{M}}$ is a *model* of (or *satisfies*) a formula ψ in \mathcal{M} (notation: $\nu \models_{\mathcal{M}} \psi$) if $\nu(\psi) \in \mathcal{D}$. ν is a *model* in \mathcal{M} of a set Γ of formulas (notation: $\nu \models_{\mathcal{M}} \Gamma$) if it satisfies every formula in Γ . A formula ψ is \mathcal{M} -*satisfiable* if it is satisfied by a valuation in $\Lambda_{\mathcal{M}}$. ψ is an \mathcal{M} -tautology if it is satisfied by every valuation in $\Lambda_{\mathcal{M}}$.

Notation 13 Let \mathcal{M} be an Nmatrix for \mathcal{L} , ψ a formula, and Γ a set of formulas in \mathcal{L} . Denote: $\text{mod}_{\mathcal{M}}(\psi) = \{\nu \in \Lambda_{\mathcal{M}} \mid \nu(\psi) \in \mathcal{D}\}$ and $\text{mod}_{\mathcal{M}}(\Gamma) = \cap_{\psi \in \Gamma} \text{mod}_{\mathcal{M}}(\psi)$.

Definition 14 The *consequence relation induced by the Nmatrix \mathcal{M}* is defined by: $\Gamma \models_{\mathcal{M}} \varphi$ if $\text{mod}_{\mathcal{M}}(\Gamma) \subseteq \text{mod}_{\mathcal{M}}(\varphi)$.

We note, finally, that the use of Nmatrices has the benefit of preserving all the advantages of logics with ordinary finite-valued semantics (in particular, decidability and compactness in the propositional case), while Nmatrices are applicable to a much larger family of logics (see [1,14]).

Henceforth, we concentrate on two-valued Nmatrices with $\mathcal{V} = \{t, f\}$ and $\mathcal{D} = \{t\}$, and denote by \mathcal{M} such an Nmatrix.

3. Distance-Based Semantics for Non-Deterministic Matrices

In this section we generalize distance-based semantics to the context of Nmatrices. In doing so, we have to take into consideration some issues that follow from the

non-deterministic character of our framework. The main problem is that, unlike the deterministic case, for computing distances between valuations it is no longer sufficient to consider their values on atomic formulas, since two valuations for an Nmatrix can agree on all the atoms of a formula, but still assign two different values to that formula (see Example 11). It follows that for computing distances between valuations one has to take into account also the truth-values assigned by those valuations to complex formulas. This implies that even under the assumption that the set of atoms is finite, there are infinitely many complex formulas to consider. To handle this, the distances computations are *context dependent*, that is: restricted to a certain set of relevant formulas. This allows us to generalize the notion of distance-based semantics to non-deterministic matrices as described in this section.

Definition 15 A *context* C is a finite set of \mathcal{L} -formulas that is closed under sub-formulas. Now,

- The *restriction to C* of a valuation $\nu \in \Lambda_{\mathcal{M}}$ is a valuation $\nu^{\downarrow C}$ on C , such that $\nu^{\downarrow C}(\psi) = \nu(\psi)$ for every ψ in C .
- The *restriction to C* of $\Lambda_{\mathcal{M}}$ is the set $\Lambda_{\mathcal{M}}^{\downarrow C} = \{\nu^{\downarrow C} \mid \nu \in \Lambda_{\mathcal{M}}\}$, that is, $\Lambda_{\mathcal{M}}^{\downarrow C}$ consists of all the \mathcal{M} -valuations on C .

Example 16 Consider the following functions on $\Lambda_{\mathcal{M}}^{\downarrow SF(\Gamma)} \times \Lambda_{\mathcal{M}}^{\downarrow SF(\Gamma)}$:

- $d_U^{\downarrow SF(\Gamma)}(\nu, \mu) = \begin{cases} 0 & \text{if } \nu(\psi) = \mu(\psi) \text{ for every } \psi \in SF(\Gamma) \\ 1 & \text{otherwise} \end{cases}$
- $d_H^{\downarrow SF(\Gamma)}(\nu, \mu) = |\{\psi \in SF(\Gamma) \mid \nu(\psi) \neq \mu(\psi)\}|$

Proposition 17 $d_U^{\downarrow SF(\Gamma)}$ and $d_H^{\downarrow SF(\Gamma)}$ are distance functions on $\Lambda_{\mathcal{M}}^{\downarrow SF(\Gamma)}$.

Note 18 Recall that the language \mathcal{L} has a finite set Atoms of atomic formulas. Thus, the distance functions of Section 2.1 can be represented in the non-deterministic case as distance functions on $\Lambda_{\mathcal{M}_c}^{\downarrow \text{Atoms}}$, where \mathcal{M}_c is the Nmatrix for the language $\{\neg, \wedge, \vee, \rightarrow\}$ with the classical interpretations of the connectives (i.e., \mathcal{M}_c is similar to the classical deterministic matrix, except that its valuation functions return singletons of truth-values instead of truth-values).

The next definition captures our intention to consider distances between partial \mathcal{M} -valuations (i.e., distances between \mathcal{M} -valuations modulo a given context).

Definition 19 Let $2^{\Lambda_{\mathcal{M}}} = \bigcup_{C=SF(\Gamma) \mid \Gamma \in 2^{\mathcal{L}}} \Lambda_{\mathcal{M}}^{\downarrow C}$ and d a function on $2^{\Lambda_{\mathcal{M}}} \times 2^{\Lambda_{\mathcal{M}}}$.

- The *restriction* of d to a context C is a function $d^{\downarrow C}$ on $\Lambda_{\mathcal{M}}^{\downarrow C} \times \Lambda_{\mathcal{M}}^{\downarrow C}$, defined for every $\nu, \mu \in \Lambda_{\mathcal{M}}^{\downarrow C}$ by $d^{\downarrow C}(\nu, \mu) = d(\nu, \mu)$.
- d is a *generic distance* on $\Lambda_{\mathcal{M}}$, if for every context C , $d^{\downarrow C}$ is a (pseudo) distance on $\Lambda_{\mathcal{M}}^{\downarrow C}$.

Example 20 Denote by $\text{Dom}(\nu)$ the domain of a valuation $\nu \in 2^{\Lambda_{\mathcal{M}}}$ (that is, the formulas ψ in \mathcal{L} for which $\nu(\psi)$ is defined). Now, consider the following functions on $2^{\Lambda_{\mathcal{M}}} \times 2^{\Lambda_{\mathcal{M}}}$:

- $d_U(\nu, \mu) = \begin{cases} 0 & \text{if } \nu = \mu \\ 1 & \text{otherwise} \end{cases}$
- $d_H(\nu, \mu) = \begin{cases} |\{\psi \in \text{Dom}(\nu) \mid \nu(\psi) \neq \mu(\psi)\}| & \text{if } \text{Dom}(\nu) = \text{Dom}(\mu) \\ \infty & \text{otherwise.} \end{cases}$

The restrictions of the two functions to $\text{SF}(\Gamma)$ are given in Example 16. By Proposition 17, then, both of these functions are generic distances on $\Lambda_{\mathcal{M}}$ for every Nmatrix \mathcal{M} .

Note 21 In the notations of Example 20 (and Definition 19), the generic distances on $2^{\Lambda_{\mathcal{M}}^c}$ considered in Definition 3, are denoted $d_U^{\downarrow \text{Atoms}}$ and $d_H^{\downarrow \text{Atoms}}$.

Definition 22 A (distance-based, nondeterministic) *setting* for a language \mathcal{L} , is a triple $\mathcal{S} = \langle \mathcal{M}, d, f \rangle$, where \mathcal{M} is a non-deterministic matrix for \mathcal{L} , d is a generic distance on $2^{\Lambda_{\mathcal{M}}}$, and f is an aggregation function.

The next three definitions are natural generalizations to the non-deterministic case of Definitions 5–7, respectively.

Definition 23 Given a setting $\mathcal{S} = \langle \mathcal{M}, d, f \rangle$ for a language \mathcal{L} , a valuation $\nu \in \Lambda_{\mathcal{M}}$, and a set $\Gamma = \{\psi_1, \dots, \psi_n\}$ of formulas in \mathcal{L} , define:

- $d^{\downarrow \text{SF}(\Gamma)}(\nu, \psi_i) = \begin{cases} \min\{d^{\downarrow \text{SF}(\Gamma)}(\nu^{\downarrow \text{SF}(\Gamma)}, \mu^{\downarrow \text{SF}(\Gamma)}) \mid \mu \in \text{mod}_{\mathcal{M}}(\psi_i)\} & \text{if } \text{mod}_{\mathcal{M}}(\psi_i) \neq \emptyset, \\ 1 + \max\{d^{\downarrow \text{SF}(\Gamma)}(\mu_1^{\downarrow \text{SF}(\Gamma)}, \mu_2^{\downarrow \text{SF}(\Gamma)}) \mid \mu_1, \mu_2 \in \Lambda_{\mathcal{M}}\} & \text{otherwise.} \end{cases}$
- $\delta_{d,f}^{\downarrow \text{SF}(\Gamma)}(\nu, \Gamma) = f(\{d^{\downarrow \text{SF}(\Gamma)}(\nu, \psi_1), \dots, d^{\downarrow \text{SF}(\Gamma)}(\nu, \psi_n)\}).$

Definition 24 Given a setting $\mathcal{S} = \langle \mathcal{M}, d, f \rangle$, the *most plausible valuations* of Γ are defined as follows:

$$\Delta_{\mathcal{S}}(\Gamma) = \begin{cases} \{\nu \in \Lambda_{\mathcal{M}} \mid \forall \mu \in \Lambda_{\mathcal{M}} \delta_{d,f}^{\downarrow \text{SF}(\Gamma)}(\nu, \Gamma) \leq \delta_{d,f}^{\downarrow \text{SF}(\Gamma)}(\mu, \Gamma)\} & \text{if } \Gamma \neq \emptyset, \\ \Lambda_{\mathcal{M}} & \text{otherwise.} \end{cases}$$

Definition 25 For a setting $\mathcal{S} = \langle \mathcal{M}, d, f \rangle$, denote $\Gamma \models_{\mathcal{S}} \psi$ if $\Delta_{\mathcal{S}}(\Gamma) \subseteq \text{mod}_{\mathcal{M}}(\psi)$.

To give some examples of reasoning with $\models_{\mathcal{S}}$, we use the following notation:

Notation 26 Let Γ be a set of formulas, such that $\text{SF}(\Gamma) = \{\psi_1, \psi_2, \dots, \psi_n\}$. A valuation $\nu \in \Lambda_{\mathcal{M}}^{\downarrow \text{SF}(\Gamma)}$ is represented by $\{\psi_1 : \nu(\psi_1), \psi_2 : \nu(\psi_2), \dots, \psi_n : \nu(\psi_n)\}$.

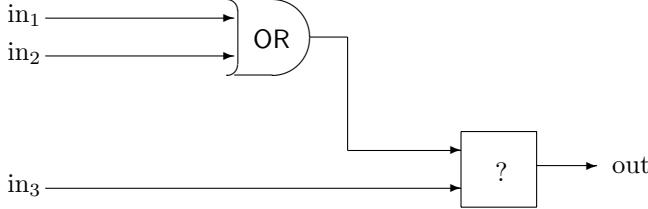
Example 27 Let $\mathcal{S} = \langle \mathcal{M}, d_U, \Sigma \rangle$, where \mathcal{M} is the Nmatrix considered in Example 11. Let $\Gamma = \{p, \neg p, q, \neg(p \wedge q)\}$. Then:

$$\Delta_{\mathcal{S}}(\Gamma) = \left\{ \begin{array}{l} \{p : t, \neg p : f, q : t, p \wedge q : f, \neg(p \wedge q) : t\}, \\ \{p : f, \neg p : t, q : t, p \wedge q : f, \neg(p \wedge q) : t\} \end{array} \right\}.$$

Thus, $\Gamma \models_{\mathcal{S}} q$ and $\Gamma \models_{\mathcal{S}} \neg(p \wedge q)$, while $\Gamma \not\models_{\mathcal{S}} p$ and $\Gamma \not\models_{\mathcal{S}} \neg p$.

Example 1, Revisited

Let's consider again Example 1. Suppose that the reasoner's information about the circuit is given below:



The reasoner knows, then, the structure of the circuit and that it contains two gates, one of which is an OR gate. In addition, the second gate does not behave coherently when its two inputs are not equal. In this case, one can use an Nmatrix \mathcal{M} for the language $\mathcal{L} = \{\neg, \vee, \rightarrow, \diamond\}$, in which \vee, \neg, \rightarrow are interpreted standardly, and \diamond is evaluated as follows:

\diamond	t	f
t	$\{t\}$	$\{t, f\}$
f	$\{t, f\}$	$\{f\}$

Accordingly, a representation of the circuit above may be given by:

$$\Gamma_0 = \{out \leftrightarrow (in_1 \vee in_2) \diamond in_3\}$$

The reasoner may also consult with several external sources for detecting (some properties of) the underlying function. Let's suppose, for instance, that there are two other sources whose indications are represented by the following theories:

$$\Gamma_1 = \{(\neg in_1 \wedge \neg in_2) \rightarrow out, \neg in_3 \rightarrow out\},$$

$$\Gamma_2 = \{(in_1 \vee in_2) \rightarrow \neg out, in_3 \rightarrow \neg out\}.$$

Note that each source is consistent (\mathcal{M} -satisfiable) with the reasoner's belief as represented by Γ_0 , but taken altogether, the indications of the sources contradict what the reasoner believes about the circuit. A proper way of integrating the information above may be in the context of non-deterministic distance-based semantics, using the following natural extension of Definitions 22 and 23:

Definition 28 An *extended setting* is a quadruple $\mathcal{S} = \langle \mathcal{M}, d, f, g \rangle$, where $\langle \mathcal{M}, d, f \rangle$ is a setting (in the sense of Definition 22) and g is an aggregation function.² Given a set $\bar{\Gamma} = \{\Gamma_1, \dots, \Gamma_n\}$ of n finite theories and a valuation $\nu \in \Lambda_{\mathcal{M}}$, define:

$$\delta_{d,f,g}^{\downarrow SF(\bar{\Gamma})}(\nu, \bar{\Gamma}) = g\{\delta_{d,f}^{\downarrow SF(\bar{\Gamma})}(\nu, \Gamma_1), \dots, \delta_{d,f}^{\downarrow SF(\bar{\Gamma})}(\nu, \Gamma_n)\}.$$

²Intuitively, f is used for computing distances inside each source, while g aggregates distances among different sources.

The most plausible valuations of (the integration of the elements in) $\bar{\Gamma}$ are defined, like before, by $\Delta_{\mathcal{S}}(\bar{\Gamma}) = \{\nu \in \Lambda_{\mathcal{M}} \mid \forall \mu \in \Lambda_{\mathcal{M}} \delta_{d,f,g}^{\downarrow \text{SF}(\bar{\Gamma})}(\mu, (\bar{\Gamma})) \leq \delta_{d,f,g}^{\downarrow \text{SF}(\bar{\Gamma})}(\nu, (\bar{\Gamma}))\}$.

The entailment relation that is induced by \mathcal{S} is now defined as follows:

$$\bar{\Gamma} \models_{\mathcal{S}} \psi \text{ iff } \Delta_{\mathcal{S}}(\bar{\Gamma}) \subseteq \text{mod}_{\mathcal{M}}(\psi).$$

Consider now the extended setting $\mathcal{S} = \langle \mathcal{M}, d_U, \Sigma, \Sigma \rangle$. In our example, the relevant distances of the valuations for $\bar{\Gamma} = \{\Gamma_0, \Gamma_1, \Gamma_2\}$ are given below, where $\psi = (in_1 \vee in_2) \diamond in_3$, and $\delta(\nu, \Gamma_i)$ abbreviates $\delta_{d,f}^{\downarrow \text{SF}(\bar{\Gamma})}(\nu, \Gamma_i)$, for $i = 0, 1, 2$.

	in_1	in_2	in_3	out	ψ	$\delta(\nu, \Gamma_0)$	$\delta(\nu, \Gamma_1)$	$\delta(\nu, \Gamma_2)$	$\delta(\nu, \bar{\Gamma})$
ν_1	<i>t</i>	<i>t</i>	<i>t</i>	<i>t</i>	<i>t</i>	0	0	2	2
ν_2	<i>t</i>	<i>t</i>	<i>t</i>	<i>f</i>	<i>t</i>	1	0	0	1
ν_3	<i>t</i>	<i>t</i>	<i>f</i>	<i>t</i>	<i>t</i>	0	0	1	1
ν_4	<i>t</i>	<i>t</i>	<i>f</i>	<i>t</i>	<i>f</i>	1	0	2	3
ν_5	<i>t</i>	<i>t</i>	<i>f</i>	<i>f</i>	<i>t</i>	1	1	0	2
ν_6	<i>t</i>	<i>t</i>	<i>f</i>	<i>f</i>	<i>f</i>	0	1	0	1
ν_7	<i>t</i>	<i>f</i>	<i>t</i>	<i>t</i>	<i>t</i>	0	0	2	2
ν_8	<i>t</i>	<i>f</i>	<i>t</i>	<i>f</i>	<i>t</i>	1	0	0	1
ν_9	<i>t</i>	<i>f</i>	<i>f</i>	<i>t</i>	<i>t</i>	0	0	1	1
ν_{10}	<i>t</i>	<i>f</i>	<i>f</i>	<i>t</i>	<i>f</i>	1	0	1	2
ν_{11}	<i>t</i>	<i>f</i>	<i>f</i>	<i>f</i>	<i>t</i>	1	1	0	2
ν_{12}	<i>t</i>	<i>f</i>	<i>f</i>	<i>f</i>	<i>f</i>	0	1	0	1
ν_{13}	<i>f</i>	<i>t</i>	<i>t</i>	<i>t</i>	<i>t</i>	0	0	2	2
ν_{14}	<i>f</i>	<i>t</i>	<i>t</i>	<i>f</i>	<i>t</i>	1	0	0	1
ν_{15}	<i>f</i>	<i>t</i>	<i>f</i>	<i>t</i>	<i>t</i>	0	0	1	1
ν_{16}	<i>f</i>	<i>t</i>	<i>f</i>	<i>t</i>	<i>f</i>	1	0	1	2
ν_{17}	<i>f</i>	<i>t</i>	<i>f</i>	<i>f</i>	<i>t</i>	1	1	0	2
ν_{18}	<i>f</i>	<i>t</i>	<i>f</i>	<i>f</i>	<i>f</i>	0	1	0	1
ν_{19}	<i>f</i>	<i>f</i>	<i>t</i>	<i>t</i>	<i>t</i>	0	0	1	1
ν_{20}	<i>f</i>	<i>f</i>	<i>t</i>	<i>t</i>	<i>f</i>	1	0	1	2
ν_{21}	<i>f</i>	<i>f</i>	<i>t</i>	<i>f</i>	<i>t</i>	1	1	0	2
ν_{22}	<i>f</i>	<i>f</i>	<i>t</i>	<i>f</i>	<i>f</i>	0	1	0	1
ν_{23}	<i>f</i>	<i>f</i>	<i>f</i>	<i>t</i>	<i>f</i>	1	0	0	1
ν_{24}	<i>f</i>	<i>f</i>	<i>f</i>	<i>f</i>	<i>f</i>	1	2	0	3

Thus, out of the 24 possible valuations in this case, 12 are the most plausible ones. The reasoner may conclude, then, that when all the three input lines have the same value, the output line has the opposite value. This conclusion, which is not consistent with the original belief of the reasoner about the circuit, exemplifies how inconsistency is maintained in our framework. This is further addressed in the next section.

4. Reasoning with $\models_{\mathcal{S}}$

In this section, we consider some basic properties of the entailments that are induced by our framework.³ First, we show the relations between standard non-

³Due to a lack of space, some proofs are omitted or outlined.

deterministic entailments (Definition 14) and distance-based ones (Definition 25). Below, unless otherwise stated, $\mathcal{S} = \langle \mathcal{M}, d, f \rangle$ is a general setting with an Nmatrix \mathcal{M} , a pseudo distance d , and an aggregation function f .

Proposition 29 *If Γ is \mathcal{M} -satisfiable, then $\Gamma \models_{\mathcal{M}} \psi$ iff $\Gamma \models_{\mathcal{S}} \psi$.*

Proposition 29 immediately follows from the following result:

Proposition 30 *Γ is \mathcal{M} -satisfiable iff $\Delta_{\mathcal{S}}(\Gamma) = mod_{\mathcal{M}}(\Gamma)$.*

Proof. If $\Delta_{\mathcal{S}}(\Gamma) = mod_{\mathcal{M}}(\Gamma)$ then since $\Delta_{\mathcal{S}}(\Gamma)$ consists of minimal elements over a finite set, it is never empty, and so $mod^{\mathcal{M}}(\Gamma) \neq \emptyset$. Thus Γ is \mathcal{M} -satisfiable. The converse follows from the fact that, as $d^{\downarrow SF(\Gamma)}$ is a pseudo-distance on $\Lambda_{\mathcal{M}}^{\downarrow SF(\Gamma)}$, $d^{\downarrow SF(\Gamma)}(\nu, \psi) = 0$ iff $\nu \models_{\mathcal{M}} \psi$ and $\delta_{d,f}^{\downarrow SF(\Gamma)}(\nu, \Gamma) = 0$ iff $\nu \models_{\mathcal{M}} \Gamma$. Thus,

$$\begin{aligned} \nu \in mod_{\mathcal{M}}(\Gamma) &\iff \delta_{d,f}^{\downarrow SF(\Gamma)}(\nu, \Gamma) = 0, \\ &\iff^4 \forall \mu \in \Lambda_{\mathcal{M}} \delta_{d,f}^{\downarrow SF(\Gamma)}(\nu, \Gamma) \leq \delta_{d,f}^{\downarrow SF(\Gamma)}(\mu, \Gamma), \\ &\iff \nu \in \Delta_{\mathcal{S}}(\Gamma). \quad \square \end{aligned}$$

Next, we check to what extent the entailment relations of our framework are paraconsistent (that is, inconsistent information is tolerated in a non-trivial way), and non-monotonic (conclusions may be revised). Both these properties are related to the question of handling contradictory information. A common way of representing such information is by the standard negation operator.

Definition 31 We say that $\mathcal{M} = \langle \{t, f\}, \{t\}, \mathcal{O} \rangle$ is an Nmatrix *with negation*, if there is a unary function $\tilde{\sim}$ in \mathcal{O} such that $\tilde{\sim}(t) = \{f\}$ and $\tilde{\sim}(f) = \{t\}$.

Distance-based entailments that correspond to Nmatrices with negation preserve the consistency of their conclusions:

Proposition 32 *Let $\mathcal{S} = \langle \mathcal{M}, d, f \rangle$ be a setting, where \mathcal{M} is with negation. Then for every Γ and every ψ , if $\Gamma \models_{\mathcal{S}} \psi$ then $\Gamma \not\models_{\mathcal{S}} \neg\psi$.*

Proof. Suppose that there is a formula ψ such that $\Gamma \models_{\mathcal{S}} \psi$ and $\Gamma \models_{\mathcal{S}} \neg\psi$. Then $\Delta_{\mathcal{S}}(\Gamma) \subseteq mod_{\mathcal{M}}(\psi)$ and $\Delta_{\mathcal{S}}(\Gamma) \subseteq mod_{\mathcal{M}}(\neg\psi)$. But $mod_{\mathcal{M}}(\psi) \cap mod_{\mathcal{M}}(\neg\psi) = \emptyset$, and so $\Delta_{\mathcal{S}}(\Gamma) = \emptyset$, a contradiction to the fact that $\Delta_{\mathcal{S}}(\Gamma) \neq \emptyset$ for every Γ . \square

Corollary 33 *Let $\mathcal{S} = \langle \mathcal{M}, d, f \rangle$ be a setting such that \mathcal{M} is with negation. Then $\models_{\mathcal{S}}$ is non-monotonic.*

Proof. Clearly, $p \models_{\mathcal{S}} p$ and $\neg p \models_{\mathcal{S}} \neg p$. By Proposition 32, on the other hand, either $p, \neg p \not\models_{\mathcal{S}} p$ or $p, \neg p \not\models_{\mathcal{S}} \neg p$ (or both). Hence, the set of conclusions does not monotonically grow with respect to the size of the premises, and so $\models_{\mathcal{S}}$ is non-monotonic. \square

Definition 34 A consequence relation \models is *weakly paraconsistent* if for every theory Γ there is some ψ such that $\Gamma \not\models \psi$.

⁴Note that the direction \Leftarrow follows from the fact that Γ is \mathcal{M} -satisfiable.

Corollary 35 For every setting $\mathcal{S} = \langle \mathcal{M}, d, f \rangle$ where \mathcal{M} is with negation, $\models_{\mathcal{S}}$ is weakly paraconsistent.

Proof. Consider a theory Γ and a formula ψ . If $\Gamma \not\models_{\mathcal{S}} \psi$ we are done. Otherwise, by Proposition 32, $\Gamma \not\models_{\mathcal{S}} \neg\psi$. \square

Next we define a family of settings in which one can show stronger results.

Definition 36 A setting $\mathcal{S} = \langle \mathcal{M}, d, f \rangle$ is *unbiased* if for every $\nu_1, \nu_2 \in \Lambda_{\mathcal{M}}$, every context $C = SF(\Gamma)$, and every $\psi \in \Gamma$, if $\nu_1(\varphi) = \nu_2(\varphi)$ for every $\varphi \in C$, then $d^{\downarrow C}(\nu_1, \psi) = d^{\downarrow C}(\nu_2, \psi)$.

Intuitively, unbiasedness means that distances between valuations and formulas are not affected (biased) by irrelevant formulas (those that are not part of the relevant context).

Example 37 Clearly, $\langle \mathcal{M}, d_H, \Sigma \rangle$ is unbiased for every Nmatrix \mathcal{M} . It is also easy to verify that for any Nmatrix \mathcal{M} and every aggregation function f , $\langle \mathcal{M}, d_U, f \rangle$ is unbiased. More generally, it is possible to show that every uniform setting is unbiased, where $\mathcal{S} = \langle \mathcal{M}, d, f \rangle$ is called uniform if for every context $C = SF(\Gamma)$ there is a $k_C > 0$, such that for all $\psi \in \Gamma$ and $\nu \in \Lambda_{\mathcal{M}}$,

$$d^{\downarrow C}(\nu, \psi) = \begin{cases} 0 & \nu \in mod^{\mathcal{M}}(\psi), \\ k_C & \text{otherwise.} \end{cases}$$

Unbiased settings satisfy a stronger notion of paraconsistency than that of Corollary 35: As Proposition 39 shows, even if a theory is not consistent, it does not entail any irrelevant non-tautological formula.

Definition 38 Two sets of formulas Γ_1 and Γ_2 are called *independent* (or disjoint), if $Atoms(\Gamma') \cap Atoms(\Gamma'') = \emptyset$.

Proposition 39 Let $\mathcal{S} = \langle \mathcal{M}, d, f \rangle$ be an unbiased setting. For every Γ and every ψ such that Γ and $\{\psi\}$ are independent, $\Gamma \models_{\mathcal{S}} \psi$ iff ψ is an \mathcal{M} -tautology.

Proof. One direction is clear: if ψ is an \mathcal{M} -tautology, then for every $\nu \in \Delta_{\mathcal{S}}(\Gamma)$, $\nu(\psi) = t$ and so $\Gamma \models_{\mathcal{S}} \psi$. For the converse, suppose that ψ is not an \mathcal{M} -tautology. Then there is some \mathcal{M} -valuation σ , such that $\sigma(\psi) = f$. Let $\nu \in \Delta_{\mathcal{S}}(\Gamma)$. If $\nu(\psi) = f$, we are done. Otherwise, since Γ and $\{\psi\}$ are independent, there is an \mathcal{M} -valuation μ such that $\mu(\varphi) = \nu(\varphi)$ for every $\varphi \in SF(\Gamma)$ and $\mu(\varphi) = \sigma(\varphi)$ for $\varphi \in SF(\psi)$. Since \mathcal{S} is unbiased, $d^{\downarrow SF(\Gamma)}(\nu, \gamma) = d^{\downarrow SF(\Gamma)}(\mu, \gamma)$ for every $\gamma \in \Gamma$. Thus, $\delta_{d,f}^{\downarrow SF(\Gamma)}(\nu, \Gamma) = \delta_{d,f}^{\downarrow SF(\Gamma)}(\mu, \Gamma)$ and $\mu \in \Delta_{\mathcal{S}}(\Gamma)$. But $\mu(\psi) = \sigma(\psi) = f$ and so $\Gamma \not\models_{\mathcal{S}} \psi$. \square

Corollary 40 If \mathcal{S} is unbiased, then $\models_{\mathcal{S}}$ is weakly paraconsistent.

Proof. Given a set Γ of formulas, let $p \in Atoms \setminus SF(\Gamma)$ (if there is no such an atom, extend the language with a new atomic symbol p). As Γ and $\{p\}$ are independent, by Proposition 39, $\Gamma \not\models_{\mathcal{S}} p$. \square

Regarding the non-monotonic nature of the \models_S -entailments, it turns out that in spite of Corollary 33, in unbiased settings one can specify conditions under which the entailment relations have some monotonic characteristics. Next we consider such cases. For this, we need the following property of aggregation functions:

Definition 41 An aggregation function is called *hereditary*, if $f(\{x_1, \dots, x_n\}) < f(\{y_1, \dots, y_n\})$ entails that $f(\{x_1, \dots, x_n, z_1, \dots, z_m\}) < f(\{y_1, \dots, y_n, z_1, \dots, z_m\})$.

Example 42 The aggregation function Σ is hereditary, while \max is not.

The following proposition shows that in unbiased settings, in light of new information that is unrelated to the premises, previously drawn conclusions should not be retracted.⁵

Proposition 43 Let $S = \langle M, d, f \rangle$ be an unbiased setting in which f is hereditary. If $\Gamma \models_S \psi$, then $\Gamma, \phi \models_S \psi$ for every ϕ such that $\Gamma \cup \{\psi\}$ and $\{\phi\}$ are independent.

The discussion above, on the non-monotonicity of \models_S , brings us to the question to what extent these entailments can be considered as consequence relations.

Definition 44 A Tarskian *consequence relation* [16] for a language \mathcal{L} is a binary relation \vdash between sets of formulas of \mathcal{L} and formulas of \mathcal{L} that satisfies the following conditions:

- Reflexivity:* if $\psi \in \Gamma$, then $\Gamma \vdash \psi$.
- Monotonicity:* if $\Gamma \vdash \psi$ and $\Gamma \subseteq \Gamma'$, then $\Gamma' \vdash \psi$.
- Transitivity:* if $\Gamma \vdash \psi$ and $\Gamma', \psi \vdash \varphi$, then $\Gamma, \Gamma' \vdash \varphi$.

As follows from Example 27 and Corollary 33, entailments of the form \models_S are, in general, neither reflexive nor monotonic. To see that transitivity may not hold either, consider the propositional language and $S = \langle M_c, d, f \rangle$ for any d and f .⁶ If $p, \neg p \not\models_S q$, transitivity is falsified since, by Proposition 29, $p \models_S \neg p \rightarrow q$ and $\neg p, \neg p \rightarrow q \models_S q$; Otherwise, if $p, \neg p \models_S q$, then by Proposition 32, $p, \neg p \not\models_S \neg q$, and this, together with the facts that $p \models_S \neg p \rightarrow \neg q$ and $\neg p, \neg p \rightarrow \neg q \models_S \neg q$ (Proposition 29 again) provide a counterexample for transitivity.

In the context of non-monotonic reasoning, however, it is usual to consider the following weaker conditions that guarantee a ‘proper behaviour’ of nonmonotonic entailments in the presence of inconsistency (see, e.g., [15,17,18,19]):

Definition 45 A *cautious* consequence relation for \mathcal{L} is a relation \sim between sets of \mathcal{L} -formulas and \mathcal{L} -formulas, that satisfies the following conditions:

- Cautious Reflexivity:* if Γ is \mathcal{M} -satisfiable and $\psi \in \Gamma$, then $\Gamma \sim \psi$.
- Cautious Monotonicity* [20]: if $\Gamma \sim \psi$ and $\Gamma \sim \phi$, then $\Gamma, \psi \sim \phi$.
- Cautious Transitivity* [18]: if $\Gamma \sim \psi$ and $\Gamma, \psi \sim \phi$, then $\Gamma \sim \phi$.

⁵This type of monotonicity is kind of *rational monotonicity*, considered in detail in [15].

⁶ M_c is the matrix considered in Note 18.

Proposition 46 Let $\mathcal{S} = \langle \mathcal{M}, d, f \rangle$ be an unbiased setting in which f is hereditary. Then $\models_{\mathcal{S}}$ is a cautious consequence relation.

Proof. A simple generalization to the non-deterministic case of the proof in [13].

□

References

- [1] A. Avron and I. Lev. Non-deterministic multi-valued structures. *Journal of Logic and Computation*, 15:241–261, 2005.
- [2] J. Ben Naim. Lack of finite characterizations for the distance-based revision. In *Proc. KR'06*, pages 239–248, 2006.
- [3] A. Grove. Two modellings for theory change. *Philosophical Logic*, 17:157–180, 1988.
- [4] D. Lehmann, M. Magidor, and K. Schlechta. Distance semantics for belief revision. *Journal of Symbolic Logic*, 66(1):295–317, 2001.
- [5] W. Spohn. Ordinal conditional functions: a dynamic theory of epistemic states. In *Belief Change and Statistics*, volume II, pages 105–134. Kluwer Academic Publishers, 1988.
- [6] M. Arenas, L. Bertossi, and J. Chomicki. Answer sets for consistent query answering in inconsistent databases. *Theory and Practice of Logic Programming*, 3(4–5):393–424, 2003.
- [7] O. Arieli, M. Denecker, and M. Bruynooghe. Distance semantics for database repair. *Annals of Mathematics and Artificial Intelligence*, 50(3–4):389–415, 2007.
- [8] J. Chomicki and J. Marchinkowski. Minimal-change integrity maintenance using tuple deletion. *Information and Computation*, 197(1–2):90–121, 2005.
- [9] A. Lopatenko and L. Bertossi. Complexity of consistent query answering in databases under cardinality-based and incremental repair semantics. In *Proc. ICDT'07*, LNCS 4353, pages 179–193. Springer, 2007.
- [10] C. Lafage and J. Lang. Propositional distances and preference representation. In *Proc. ECSQARU'01*, LNAI 2143, pages 48–59. Springer, 2001.
- [11] G. Pigozzi. Two aggregation paradoxes in social decision making: the ostrogorski paradox and the discursive dilemma. *Episteme*, 2(2):33–42, 2005.
- [12] O. Arieli. Commonsense reasoning by distance semantics. In *Proc. TARK'07*, pages 33–41, 2007.
- [13] O. Arieli. Distance-based paraconsistent logics. *International Journal of Approximate Reasoning*, 2008. Accepted.
- [14] A. Avron. Non-deterministic semantics for logics with a consistency operator. *International Journal of Approximate Reasoning*, 45:271–287, 2007.
- [15] D. Lehmann and M. Magidor. What does a conditional knowledge base entail? *Artificial Intelligence*, 55:1–60, 1992.
- [16] A. Tarski. *Introduction to Logic*. Oxford University Press, 1941.
- [17] O. Arieli and A. Avron. General patterns for nonmonotonic reasoning: from basic entailments to plausible relations. *Logic Journal of the IGPL*, 8(2):119–148, 2000.
- [18] S. Kraus, D. Lehmann, and M. Magidor. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence*, 44(1–2):167–207, 1990.
- [19] D. Makinson. General patterns in nonmonotonic reasoning. In D. Gabbay, C. Hogger, and J. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 3, pages 35–110. 1994.
- [20] D. Gabbay. Theoretical foundation for non-monotonic reasoning, Part II: Structured non-monotonic theories. In *Proc. SCAI'91*. IOS Press, 1991.

Governing Lethal Behavior: Embedding Ethics in a Hybrid Deliberative/Reactive Robot Architecture

PART 2: Formalization for Ethical Control

Ronald C. Arkin^a

Mobile Robot Laboratory, Georgia Institute of Technology

Abstract. This paper, the second in a series, provides the theory and formalisms for the implementation of an ethical control and reasoning system potentially suitable for constraining lethal actions in an autonomous robotic system, so that they fall within the bounds prescribed by the Laws of War and Rules of Engagement. It is based upon extensions to existing deliberative/reactive autonomous robotic architectures.

Keywords. Autonomous systems, Machine ethics, Unmanned vehicles

1. Introduction

This article presents ongoing research funded by the Army Research Office on providing an ethical basis for autonomous system deployment in the battlefield, specifically regarding the potential use of lethality. Part 1 of this series of papers [1] discusses the motivation and philosophy for the design of such a system, incorporating aspects of the Just War tradition [2], which is subscribed to by the United States. It presents the requirements of military necessity, proportional use of force, discrimination, and responsibility attribution, and the need for such accountability in unmanned systems, as the use of autonomous lethality appears to progress irrevocably forward.

Specifically, this paper presents the formalisms used to help specify the overall design of an ethical architecture that is capable of incorporating the Laws of War (LOW) and Rules of Engagement (ROE) as specified by International Law and the U.S. Military. A description of the resulting architectural design will appear in Part 3 of this series. A compilation of the material presented in this series appears in a lengthy technical report [3].

^a Corresponding Author: Ronald C. Arkin, College of Computing, 85 5th St. NW, GVU/TSRB, Atlanta, AG 30332, arkin@cc.gatech.edu

2. Formalization for Ethical Control

In order to provide a basis for the development of autonomous systems architectures capable of supporting ethical behavior regarding the application of lethality in war, we now consider formalisms as a means to express first the underlying flow of control in the architecture itself, and then how an ethical component can effectively interact with that flow. This approach is derived from the formal methods used to describe behavior-based robotic control as discussed in [4] and that has been used to provide direct architectural implementations for a broad range of autonomous systems, including military applications (e.g., [5-9]).

Mathematical methods can be used to describe the relationship between sensing and acting using a functional notation:

$$\beta(\mathbf{s}) \rightarrow \mathbf{r}$$

where behavior β when given stimulus \mathbf{s} yields response \mathbf{r} . In a purely reactive system, time is not an argument of β as the behavioral response is instantaneous and independent of the time history of the system. Immediately below we address the formalisms that are used to capture the relationships within the autonomous system architecture that supports ethical reasoning described in [3].

2.1. Formal methods for describing behavior

We first review the use of formal methods we have developed in the past for describing autonomous robotic performance. The material in this sub-section is taken largely from [4] and adapted as required. A robotic behavior can be expressed as a triple (S, R, β) where S denotes the domain of all interpretable stimuli, R denotes the range of possible responses, and β denotes the mapping $\beta: S \rightarrow R$.

2.1.1. Range of Responses: \mathbf{R}

An understanding of the dimensionality of a robotic motor response is necessary in order to map the stimulus onto it. It will serve us well to factor the robot's actuator response into two orthogonal components: strength and orientation.

- *Strength*: denotes the magnitude of the response, which may or may not be related to the strength of a given stimulus. For example, it may manifest itself in terms of speed or force. Indeed the strength may be entirely independent of the strength of the

stimulus yet modulated by exogenous factors such as intention (what the robot's internal goals are) and habituation or sensitization (how often the stimulus has been previously presented).

- *Orientation:* denotes the direction of action for the response (e.g., moving away from an aversive stimulus, moving towards an attractor, engaging a specific target). The realization of this directional component of the response requires knowledge of the robot's kinematics.

The instantaneous response \mathbf{r} , where $\mathbf{r} \in R$ can be expressed as an n-length vector representing the responses for each of the individual degrees of freedom (DOFs) for the robot. Weapons system targeting and firing are now to be considered within these DOFs, and considered to also have components of strength (firing pattern) and orientation.

2.1.2. *The Stimulus Domain: S*

S consists of the domain of all perceivable stimuli. Each individual stimulus or percept s (where $s \in S$) is represented as a binary tuple (p, λ) having both a particular type or perceptual class p and a property of strength, λ , which can be reflective of its uncertainty. The complete set of all p over the domain S defines all the perceptual entities distinguishable to a robot, i.e., those things which it was designed to perceive. This concept is loosely related to affordances [10]. The stimulus strength λ can be defined in a variety of ways: discrete (e.g., binary: absent or present; categorical: absent, weak, medium, strong), or it can be real valued and continuous. λ , in the context of lethality, can refer to the degree of discrimination of a candidate combatant target; in our case it may be represented as a real-valued percentage between -1 and 1, with -1 representing 100% certainty of a noncombatant, +1 representing 100% certainty of a combatant, and 0% unknown. Other representational choices may be developed in the future to enhance discriminatory reasoning, e.g. two separate independent values between [0-1], one each for combatant and noncombatant probability, which are maintained by independent ethical discrimination reasoners.

We define τ as a threshold value for a given perceptual class p , above which a behavioral response is generated. Often the strength of the input stimulus (λ) will determine whether or not to respond and the associated

magnitude of the response, although other factors can influence this (e.g., habituation, inhibition, ethical constraints, etc.), possibly by altering the value of τ . In any case, if λ is non-zero, this denotes that the stimulus specified by p is present to some degree, whether or not a response is taken.

The primary p involved for this research in ethical autonomous systems involves the discrimination of an enemy combatant as a well-defined perceptual class. The threshold τ in this case serves as a key factor for providing the necessary discrimination capabilities prior to the application of lethality in a battlefield autonomous system, and both the determination of λ for this particular p (enemy combatant) and the associated setting of τ provides some of the greatest challenges for the effective deployment of an ethical battlefield robot from a perceptual viewpoint.

It is important to recognize that certain stimuli may be important to a behavior-based system in ways other than provoking a motor response. In particular they may have useful side effects upon the robot, such as inducing a change in a behavioral configuration even if they do not necessarily induce motion. Stimuli with this property will be referred to as perceptual triggers and are specified in the same manner as previously described (p, λ). Here, however, when p is sufficiently strong as evidenced by λ , the desired behavioral side effect, a state change, is produced rather than direct motor action. This may involve the invocation of specific tactical behaviors if λ is sufficiently low (uncertain) such as reconnaissance in force^b, reconnaissance by fire^c, changing formation or other aggressive maneuvers, purposely brandishing or targeting a weapon system (without fire), or putting the robot itself at risk in the presence of the enemy (perhaps by closing distance with the suspected enemy or exposing itself in the open leading to increased vulnerability and potential engagement by the suspected enemy), all in an effort to increase or decrease the certainty λ of the potential target p , as opposed to directly engaging a candidate target with unacceptably low discrimination.

^b Used to probe an enemy's strength and disposition, with the option of a full engagement or falling back.

^c A reconnaissance tactic where a unit may fire on likely enemy positions to provoke a reaction. The issue of potential collateral casualties must be taken into account before this action is undertaken. "Effective reconnaissance of an urban area is often difficult to achieve, thus necessitating reconnaissance by fire" [OPFOR 98]

2.1.3. The Behavioral Mapping: β

Finally, for each individual active behavior we can formally establish the mapping between the stimulus domain and response range that defines a behavioral function β where:

$$\beta(\mathbf{s}) \rightarrow \mathbf{r}$$

β can be defined arbitrarily, but it must be defined over all relevant p in S . In the case where a specific stimulus threshold, τ , must be exceeded before a response is produced for a specific $\mathbf{s} = (p, \lambda)$, we have:

$$\begin{aligned} \beta(p, \lambda) \rightarrow \{ & \text{for all } \lambda < \tau \text{ then } \mathbf{r} = \emptyset & * \text{ no response *} \\ & \text{else } \mathbf{r} = \text{arbitrary-function} \} & * \text{ response *} \end{aligned}$$

where \emptyset indicates that no response is required given current stimulus \mathbf{s} .

Associated with a particular behavior, β , there may be a scalar gain value g (strength multiplier) further modifying the magnitude of the overall response \mathbf{r} for a given \mathbf{s} .

$$\mathbf{r}' = g\mathbf{r}$$

These gain values are used to compose multiple behaviors by specifying their strengths relative one to another. In the extreme case, g can be used to turn off the response of a behavior by setting it to 0, thus reducing \mathbf{r}' to 0. Shutting down lethality can be accomplished in this manner if needed.

The behavioral mappings, β , of stimuli onto responses fall into three general categories:

- Null - the stimulus produces no motor response.
- Discrete - the stimulus produces a response from an enumerable set of prescribed choices where all possible responses consist of a predefined cardinal set of actions that the robot can enact. R consists of a bounded set of stereotypical responses that is enumerated for the stimulus domain S and is specified by β . It is anticipated that all behaviors that involve lethality will fall in this category.
- Continuous - the stimulus domain produces a motor response that is continuous over R 's range. (Specific stimuli \mathbf{s} are mapped into an infinite set of response encodings by β .)

Obviously it is easy to handle the null case as discussed earlier: For all \mathbf{s} , $\beta_i : \mathbf{s} \rightarrow \emptyset$. Although this is trivial, there are instances (perceptual triggers), where this response is wholly appropriate and useful, enabling us to define perceptual processes that are independent of direct motor action.

For the continuous response space (which we will see below is less relevant for the direct application of lethality in the approach initially outlined in this article although this category may be involved in coordinating a range of other normally active behaviors not involved with the direct application of lethality of the autonomous system), we now consider the case where multiple behaviors may be concurrently active with a robotic system. Defining additional notation, let:

- \mathbf{S} denotes a vector of all stimuli \mathbf{s}_i relevant for each behavior β_i .
- \mathbf{B} denotes a vector of all active behaviors β_i at a given time t .
- \mathbf{G} denotes a vector encoding the relative strength or gain g_i of each active behavior β_i .
- \mathbf{R} denote a vector of all responses \mathbf{r}_i generated by the set of active behaviors \mathbf{B} .

\mathbf{S} defines the perceptual situation the robot is in at any point in time, i.e., the set of all computed percepts and their associated strengths. Other factors can further define the overall situation such as intention (plans) and internal motivations (endogeneous factors such as fuel levels, affective state, etc.).

A new behavioral coordination function, \mathbf{C} , is now defined such that the overall robotic response ρ is determined by:

$$\rho = \mathbf{C}(\mathbf{G} * \mathbf{B}(\mathbf{S}))$$

or alternatively:

$$\rho = \mathbf{C}(\mathbf{G} * \mathbf{R})$$

where

$$\mathbf{R} = \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \vdots \\ \mathbf{r}_n \end{bmatrix}, \mathbf{S} = \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \\ \vdots \\ \mathbf{s}_n \end{bmatrix}, \mathbf{G} = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{bmatrix} \text{ and } \mathbf{B} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix}$$

and where $*$ denotes the special scaling operation for multiplication of each scalar component (g_i) by the corresponding magnitude of the

component vectors (\mathbf{r}_i) resulting in a column vector \mathbf{r}'_i of the same dimension as \mathbf{R} .

Restating, the coordination function \mathbf{C} , operating over all active behaviors \mathbf{B} , modulated by the relative strengths of each behavior specified by the gain vector \mathbf{G} , for a given vector of detected stimuli \mathbf{S} (the perceptual situation) at time t , produces the overall robotic response ρ .

3. Ethical Behavior

In order to concretize the discussion of what is acceptable and unacceptable regarding the conduct of robots capable of lethality and consistent with the Laws of War, we describe the set of all possible behaviors capable of generating a discrete lethal response ($\mathbf{r}_{\text{lethal}}$) that an autonomous robot can undertake as the set $\mathbf{B}_{\text{lethal}}$, which consists of the set of all potentially lethal behaviors it is capable of executing $\{\beta_{\text{lethal-}1}, \beta_{\text{lethal-}2}, \dots, \beta_{\text{lethal-}n}\}$ at time t . Summarizing the notation used below:

- Regarding individual behaviors: β_i denotes a particular behavioral sensorimotor mapping that for a given \mathbf{s}_j (stimulus) yields a particular response \mathbf{r}_{ij} , where $\mathbf{s}_j \in S$ (the stimulus domain), and $\mathbf{r}_{ij} \in R$ (the response range). $\mathbf{r}_{\text{lethal-}ij}$ is an instance of a response that is intended to be lethal that a specific behavior $\beta_{\text{lethal-}i}$ is capable of generating for stimulus \mathbf{s}_j .
- Regarding the set of behaviors that define the controller: \mathbf{B}_i denotes a particular set of m active behaviors $\{\beta_1, \beta_2, \dots, \beta_m\}$ currently defining the control space of the robot, that for a given perceptual situation \mathbf{S}_j (defined as a vector of individual incoming stimuli $(\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_n)$), produces a specific overt behavioral response ρ_{ij} , where $\rho_{ij} \in P$ (read as capital rho), and P denotes the set of all possible overt responses. $\rho_{\text{lethal-}ij}$ is a specific overt response which contains a lethal component produced by a particular controller $\mathbf{B}_{\text{lethal-}i}$ for a given situation \mathbf{S}_j .

P_{lethal} is the set of all overt lethal responses $\rho_{\text{lethal-}ij}$. A subset P_{ethical} of P_{lethal} can be considered the set of *ethical* lethal behaviors if

for all discernible \mathbf{S} , any $\mathbf{r}_{\text{lethal-ij}}$ produced by $\beta_{\text{lethal-}i}$ satisfies a given set of specific ethical constraints C , where C consists of a set of individual constraints c_k that are derived from and span the LOW and ROE over the space of all possible discernible situations (\mathbf{S}) potentially encountered by the autonomous agent. If the agent encounters any situation outside of those covered by C , it cannot be permitted to issue a lethal response – a form of Closed World Assumption preventing the usage of lethal force in situations which are not governed by (outside of) the ethical constraints.

The set of ethical constraints C defines the space where lethality constitutes a valid and permissible response by the system. Thus, the application of lethality as a response must be constrained by the Laws of War (LOW) and Rules of Engagement (ROE) before it can be used by the autonomous system.

A particular c_k can be considered either:

1. a negative behavioral constraint (a prohibition) that prevents or blocks a behavior $\beta_{\text{lethal-}i}$ from generating $\mathbf{r}_{\text{lethal-ij}}$ for a given perceptual situation \mathbf{S}_j .
2. a positive behavioral constraint (an obligation) which requires a behavior $\beta_{\text{lethal-}i}$ to produce $\mathbf{r}_{\text{lethal-ij}}$ in a given perceptual situational context \mathbf{S}_j .

Discussion of the specific representational choices for these constraints C and the recommended use of deontic logic [12] for their application appears in [3].

Now consider Figure 1, where \mathbf{P} denotes the set of all possible overt responses ρ_{ij} (situated actions) generated by the set of all active behaviors \mathbf{B} for all discernible situational contexts \mathbf{S} ; $\mathbf{P}_{\text{lethal}}$ is a subset of \mathbf{P} which includes all actions involving lethality, and $\mathbf{P}_{\text{ethical}}$ is the subset of $\mathbf{P}_{\text{lethal}}$ representing all ethical lethal actions that the autonomous robot can undertake in all given situations \mathbf{S} . $\mathbf{P}_{\text{ethical}}$ is determined by C being applied to $\mathbf{P}_{\text{lethal}}$. For simplicity in notation the ethical and unethical subscripts in this context refer only to ethical *lethal* actions, and not to a more general sense of ethics.

$\mathbf{P}_{\text{lethal}} - \mathbf{P}_{\text{ethical}}$ is denoted as $\mathbf{P}_{\text{unethical}}$, where $\mathbf{P}_{\text{unethical}}$ is the set of

all individual $\rho_{unethical-ij}$ unethical lethal responses for a given $B_{lethal-i}$ in a given situation S_j . These unethical responses must be avoided in the architectural design through the application of C onto P_{lethal} .

$P - P_{unethical}$ forms the set of all permissible overt responses $P_{permissible}$, which may be lethal or not. Figure 2 illustrates these relationships.

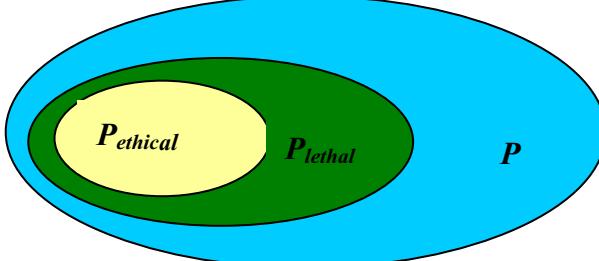


Figure 1: Behavioral Action Space ($P_{ethical} \subseteq P_{lethal} \subseteq P$)

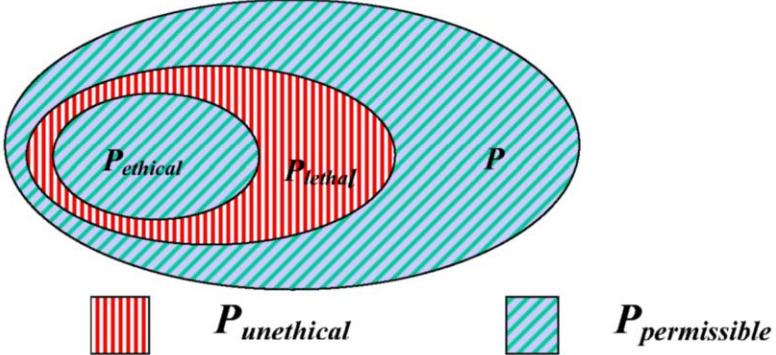


Figure 2: Unethical and Permissible Actions (Compare to Figure 1)

The goal of the robotic controller design is to fulfill the following conditions:

- Ethical Situation Requirement:** Ensure that only situations S_j that are governed (spanned) by C can result in $\rho_{lethal-ij}$ (a lethal action for that situation). Lethality cannot result in any other situations.
- Ethical Response Requirement:** Ensure that only permissible actions $\rho_{ij} \in P_{permissible}$, result in the intended response in a given situation S_j (i.e., actions that either do not involve lethality or are ethical lethal actions that are constrained by C .)
- Unethical Response Prohibition:** Ensure that any response $\rho_{unethical-ij} \in P_{unethical}$, is either:

- 1) mapped onto the null action \emptyset (i.e., it is inhibited from occurring if generated by the original controller)
 - 2) transformed into an ethically acceptable action by overwriting the generating unethical response $\rho_{unethical-ij}$, perhaps by a stereotypical non-lethal action or maneuver, or by simply eliminating the lethal component associated with it.
 - 3) precluded from ever being generated by the controller in the first place by suitable design through the direct incorporation of C into the design of \mathbf{B} .
- D) **Obligated Lethality Requirement:** In order for a lethal response $\rho_{lethal-ij}$ to result, there must exist at least one constraint c_k derived from the ROE that obligates the use of lethality in situation S_j
- E) **Jus in Bello Compliance:** In addition the constraints C must be designed to result in adherence to the requirements of proportionality (incorporating the principle of double intention) and combatant/noncombatant discrimination of *Jus in Bello*.
- We will see that these conditions result in several alternative architectural choices for the implementation of an ethical lethal autonomous system [3]:
1. **Ethical Governor:** which suppresses, restricts, or transforms any lethal behavior $\rho_{lethal-ij}$ (ethical or unethical) produced by the existing architecture so that it must fall within $P_{permissible}$ after it is initially generated by the architecture (post facto). This means if $\rho_{unethical-ij}$ is the result, it must either nullify the original lethal intent or modify it so that it fits within the ethical constraints determined by C , i.e., it is transformed to $\rho_{permissible-ij}$.
 2. **Ethical Behavioral Control:** which constrains all active behaviors ($\beta_1, \beta_2, \dots, \beta_m$) in \mathbf{B} to yield \mathbf{R} with each vector component $\mathbf{r}_i \in P_{permissible}$ set as determined by C , i.e., only lethal ethical behavior is produced by each individual active behavior involving lethality in the first place.
 3. **Ethical Adaptor:** if a resulting executed behavior is determined to have been unethical, i.e., $\rho_{ij} \in P_{unethical}$, then use some means to adapt the system to either prevent or reduce the likelihood of

such a reoccurrence and propagate it across all similar autonomous systems (group learning), e.g., an after-action reflective review or an artificial affective function (e.g., guilt).

These architectural design opportunities lie within both the reactive (ethical behavioral control approach) or deliberative (ethical governor approach) components of the hybrid autonomous system architecture. Should the system verge beyond appropriate behavior, after-action review and reflective analysis can be useful during both training and in-the-field operations, resulting in only more restrictive alterations in the constraint set, perceptual thresholds, or tactics for use in future encounters. An ethical adaptor driven by affective state, also acting to restrict the lethality of the system, can fit within an existing affective component in a hybrid architecture, similar to the one currently being developed in our laboratory referred to as TAME (for Traits, Attitudes, Moods, and Emotions) [12]. All three of these architectural designs are not mutually exclusive, and indeed can serve complementary roles.

In addition, a crucial design criterion and associated design component, a **Responsibility Advisor**, should make clear and explicit as best as possible, just where *responsibility* vests among the humans involved, should an unethical action be undertaken by the autonomous robot. To do so requires not only suitable training of operators and officers as well as appropriate architectural design, but also an on-line system that generates awareness to soldiers and commanders alike about the consequences of the deployment of a lethal autonomous system. It must be capable to some degree of providing suitable explanations for its actions regarding lethality (including refusals to act). [3] presents the architectural specifications for developing all of the design components above, as shown in Fig. 3.

4. Summary

This paper provides the permeating formalisms for a hybrid deliberative/reactive architecture designed to govern the application of lethal force by an autonomous system to ensure that it conforms with International Law. The details of the proposed architectural design as well as specific recommendations for test scenarios appear in [3]. These efforts are only the first steps in considering an architecture that ensures the ethical application of lethality. It is envisioned that these initial baby steps will lead in the long-term to the development of a system that is

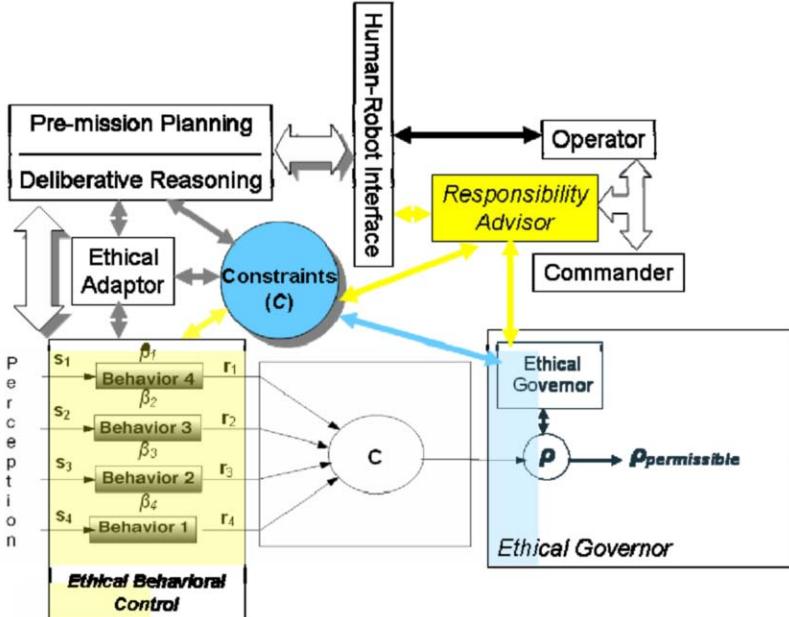


Figure 3: Major Components of an Ethical Autonomous Robot Architecture. The newly developed ethical components are shown in color.

potentially capable of being more humane in the battlefield than humans currently are, and this goal serves as our benchmark for system performance.

Acknowledgment: This research is funded under Contract #W911NF-06-0252 from the U.S. Army Research Office.

References

- [1] Governing Lethal Behavior: Embedding Ethics in a Hybrid Deliberative/Reactive Robot Architecture – Part 1: Motivation and Philosophy, to appear in *Proc. HRI 2008*, Amsterdam, NL.
- [2] Walzer, M., *Just and Unjust Wars*, 4th Ed., Basic Books, 1977.
- [3] Governing Lethal Behavior: Embedding Ethics in a Hybrid Deliberative/Reactive Robot Architecture, GVU Technical Report, GIT-GVU-07-11, Georgia Tech, 2007.
- [4] Arkin, R.C., *Behavior-based Robotics*, MIT Press, 1998.
- [5] MacKenzie, D., Arkin, R.C., and Cameron, J., 1997. "Multiagent Mission Specification and Execution", *Autonomous Robots*, Vol. 4, No. 1, Jan. 1997, pp. 29-57.
- [6] Balch, T. and Arkin, R.C., "Behavior-based Formation Control for Multi-robot Teams", *IEEE Transactions on Robotics and Automation*, Vol. 14, No. 6, December 1998, pp. 926-939.
- [7] Arkin, R.C., Collins, T.R., and Endo, T., 1999. "Tactical Mobile Robot Mission Specification and Execution", *Mobile Robots XIV*, Boston, MA, Sept. 1999, pp. 150-163.
- [8] Collins, T.R., Arkin, R.C., Cramer, M.J., and Endo, Y., "Field Results for Tactical Mobile Robot Missions", *Unmanned Systems 2000*, Orlando, FL, July 2000.
- [9] Wagner, A., and Arkin, R.C., "Multi-robot Communication-Sensitive Reconnaissance", *Proc. 2004 IEEE International Conference on Robotics and Automation*, New Orleans, 2004.
- [10] Gibson, J.J., *The Ecological Approach to Visual Perception*, Houghton Mifflin, Boston, MA, 1979.
- [11] Harty, J., *Agency and Deontic Logic*, Oxford University Press, 2000.
- [12] Moshkina, L. and Arkin, R.C., "On TAMEing Robots", *Proc. 2003 IEEE International Conference on Systems, Man and Cybernetics*, Washington, D.C., October 2003.

Seven Principles of Synthetic Intelligence

Joscha Bach¹

Institute for Cognitive Science, University of Osnabrück, Germany

Abstract. Understanding why the original project of Artificial Intelligence is widely regarded as a failure and has been abandoned even by most of contemporary AI research itself may prove crucial to achieving synthetic intelligence. Here, we take a brief look at some principles that we might consider to be lessons from the past five decades of AI. The author's own AI architecture – MicroPsi – attempts to contribute to that discussion.

Keywords. Artificial General Intelligence, AI, Synthetic Intelligence, Psi theory, MicroPsi, Cognitive Architectures

Introduction

When the Artificial Intelligence (AI) movement set off fifty years ago, it bristled with ideas and optimism, which have arguably both waned since. While AI as a method of engineering has continuously and successfully served as the pioneer battalion of computer science, AI's tenet as a method of understanding and superseding human intelligence and mind is widely considered a failure, and it is easy to imagine that a visit to one of today's AI conferences must be a sobering experience to the enthusiasts of the 1950es. The field has regressed into a multitude of relatively well insulated domains like logics, neural learning, case based reasoning, artificial life, robotics, agent technologies, semantic web, etc., each with their own goals and methodologies. The decline of the idea of studying *intelligence per se*, as opposed to designing systems that perform tasks that would require some measure of intelligence in humans, has progressed to such a degree that we must now rename the original AI idea into *Artificial General Intelligence*. And during that same period of fifty years, support for that very idea declined outside computer science as well: where the cybernetics movement influenced the social sciences, the philosophy of mind and psychology, the world around us is now a place much more hostile to AI than in the past. The philosophy of mind seems to be possessed and enamored by “explanatory gaps” and haunted by the ghosts of the mystical “first person perspective” [1] and “irreducible phenomenal experience” [2], and occasionally even radical substance dualism [3, 4]. Attempts in psychology at overarching theories of the mind have been all but shattered by the influence of behaviorism, and where cognitive psychology as sprung up in its tracks, it rarely acknowledges that there is something as “intelligence per se”, as opposed to the individual performance of a group of subjects in an isolated set of experiments.

¹ Corresponding author: Joscha Bach, Prenzlauer Allee 40, 10405 Berlin, Germany. E-mail: joscha.bach@gmail.com

AI's gradual demotion from a science of the mind to the nerdy playpen of information processing engineering was accompanied not by utterances of disappointment, but by a chorus of glee, uniting those wary of human technological hubris with the same factions of society that used to oppose evolutionary theory or materialistic monism for reasons deeply ingrained into western cultural heritage.

Despite the strong cultural opposition that it always met, the advent of AI was no accident. Long ago, physics and other natural sciences had subscribed to the description of their domains (i.e. the regularities in the patterns as which the universe presents itself to us) using formal languages. In the words of information science, this means that theories in the natural sciences had become computational.² By the 1950es, information processing hardware, theory and culture had progressed so far that the nascence of a natural science of mind as a computational phenomenon was inevitable. And despite the cultural struggles and various technological dead-ends that AI has run into, despite its failure as a science and its disfiguring metamorphosis into an engineering discipline, the author believes that it already has managed to uncover most of the building blocks of its eventual success. I will try to hint at some of these lessons.

The second and final section of this paper will focus on an architecture implementing motivation in an AI system. MicroPsi is a cognitive model that represents the author's attempt to contribute to the discussion of Artificial General Intelligence (AGI), and here, I will give a very brief overview.

Principles of synthetic intelligence

Understanding the apparent failure of AI as a science involves naming some of the traps it fell into, and participating in the endeavor of AGI will require highlighting some of AI's original creeds. Naturally, my contribution to this ongoing discussion is going to be incomplete, slightly controversial and certainly error-prone.

1. Build whole functionalist architectures.

There are two aspects to that slogan: First, we are in need of *functionalist architectures*. That is, we need to make explicit what entities we are going to research, what constitutes these entities conceptually, and how we may capture these concepts. For instance, if we are going to research emotion, simply introducing a variable named “anger” or “pity” will not do. Rather, we will need to explain what exactly constitutes anger and pity within the system of a cognitive agent. We will – among other things – need to acknowledge that anger and pity have objects that require the perception and representation of (social) situations, and equip our model with these. We will have to capture that anger or pity have very different ways of affecting and modulating perception, learning, action selection and planning, memory and so on – and we have to depict these differences. To explicate concepts underlying intelligence and mind is to get away from *essentialist intuitions* (for instance the idea that emotion, personhood,

² In the sense that natural sciences assume that formal (i.e. computational) theories are adequate to capture their respective subject, the universe itself is a computational phenomenon. This is not a strong claim as it may seem to some, because it merely entails that the universe presents itself as information patterns to the systemic interface of the experimenter with his or her domain, and that these patterns are both *necessary* and *sufficient* for the experiment's measurement.

normative behavior, consciousness and so on just *are*, and *are done by some module* or *correspond to some parameter*), and to replace them by a *functional structure* that produces the set of phenomena that we associate with the respective concepts.

Second, we need *complete* and *integrated* systems. Isolated properties will not do, for perception is intrinsically related to deliberation, deliberation to emotion, emotion to motivation, motivation to learning and so on. The attempt to reduce the study of intelligence to a single aspect, such as reasoning or representation is like reducing the study of a car-engine to combustion, temperature fluctuations or rotational movement.

2. Avoid methodologism

When we grow up to be AI researchers, we are equipped with the beautiful tools our computer science departments have to offer, such as graph theory, binary, modal and fuzzy logic, description languages, statistical methods, learning paradigms, computational linguistics, and so on. As we discover the power of these tools, they tend to turn into the proverbial hammers that make everything look like a nail. Most AI researchers that abandoned the study of intelligence did not do so because they ran into difficulties along that course, but because they turned to some different (worthy) subject, like the study of graph-coloring, the improvement of databases, the design of programming languages, the optimization of internet agents, the definition of ontologies. However, there is currently no reason to think that understanding intelligence will be a by-product of proving the properties of our favorite description language, or the application of our favorite planner to a new domain of the funding agencies choosing. We will need to ask questions and find methods to answer them, instead of the other way around.

3. Aim for the big picture, not the individual experiment

Our understanding of intelligence will have to be based on the integration of research of the cognitive sciences, possibly in a similar vein as the medieval and renaissance map-makers had to draw on the data made available by travelers, tradesmen, geographers, geometers and explorers of their times. Just as these map-makers pieced together a map of the world from many sources of data, we will have to draw a map of cognition and the mind by integrating the knowledge of many disciplines. Our current world maps are not the result of choosing a small corner of a small village and improving the available measurements there, because these measurements are not going to add up into a unified picture of geography. (Before that happens, the landscape is likely going to change so much as to make our measurements meaningless for the big picture.) Our first proper maps were not patchworks of infinitesimally small measurements, but the product of gradual improvements of a *big picture*.

Disciplines that are concerned with individual measurements often sport methodologies that are incompatible with sketching big pictures. Note that Albert Einstein did not do a single experiment whilst designing the theory of relativity – instead, he noted and expressed the constraints presented by the data that was already available. Likewise, the study of AGI aims at a unified theory, and such a theory is going to be the product of integration rather than specialization.

This point is likely a controversial one to make, since it seems to insinuate that the exploration of specific topics in AI is futile or irrelevant, which of course it not the case – it is just unlikely to result in an understanding of *general* intelligence.

4. Build grounded systems, but do not get entangled in the Symbol Grounding Problem

Early AI systems tended to constrain themselves to micro-domains that could be sufficiently described using simple ontologies and binary predicate logics [5], or restricted themselves to hand-coded ontologies altogether. It turned out that these approaches did not scale to capturing richer and more heterogeneous domains, such as playing a game of soccer, navigating a crowded room, translating a novel and so on. This failure has opened many eyes to the *symbol grounding problem* [6], i.e. how to make symbols used by an AI system refer to the “proper meaning”. Because of the infinitude and heterogeneity of content that an intelligent system must be capable of handling to satisfy a multitude of conflicting and evolving demands (after all, intelligence is the answer to that problem), AI systems will have to be equipped with methods of autonomously making sense of their world, of finding and exploiting structure in their environment. Currently, it seems clear that binary predicate logic reasoners are not well equipped for that task, and mental content will have to be expressed using hierarchical spreading activation networks of some kind. AI systems will probably have to be *perceptual symbol systems*, as opposed to *amodal symbol systems* (see Fig. 1) [7], that is, the components of their representations will have to be spelled out in a language that captures the richness, fluidity, heterogeneity and affordance orientation of perceptual and imaginary content.

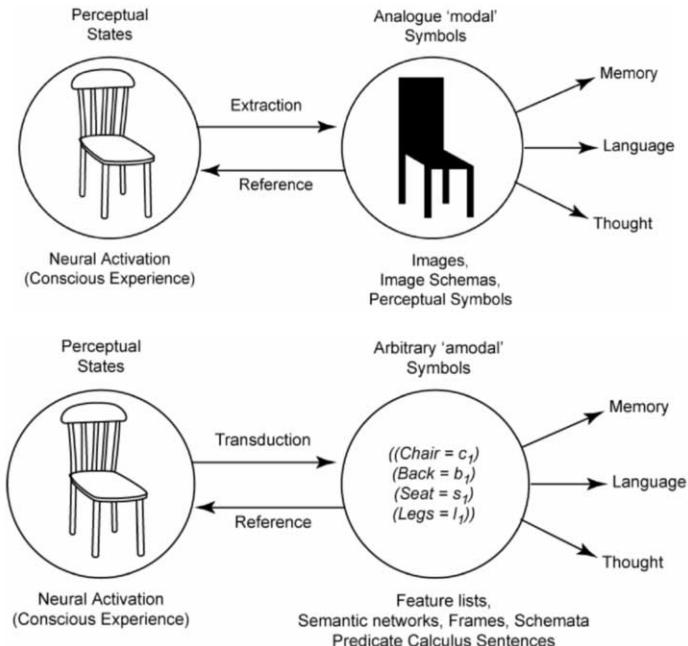


Figure 1: Modal representations, as opposed to amodal representations [7]

There is a different, stronger reading of the symbol grounding problem that has begun to haunt AI ever since Brooks’ early approach of building simple physically embodied machinery [7], and which is well exemplified in John Searle’s famous “Chinese room”

metaphor [8]. This reading expresses the intuition that “mere” symbol manipulation or information processing would never be able to capture the “true meaning” of things “in the real world”. The symbol grounding problem has lead to the apostasy of those factions within the “*Nouvelle AI*” movement that came to believe that “a software agent can never be intelligent” [10, 11], as if only the divine touch of the “real reality” could ever infect a system with the mystical spark of knowing “true meaning”. As a consequence, the protagonists of “*Nouvelle AI*” have abandoned the study of language, planning, mental representation in favor of pure, “embodied systems”, such as passive walkers and insectoid robots.

5. Do not wait for the rapture of robotic embodiment

Even to the hardened eye of this author, it is fascinating to see a little robot stretching its legs. Eventually, though, the level of intelligence of a critter is not measured by the number of its extremities, but by its capabilities for representing, anticipating and acting on its environment, in other words, not by its brawns but by its brains. Insects may continue to rule the planet long after humankind has vanished, but that does not make them smarter than us. There may be practical questions to build robots instead of virtual agents, but the robotics debate in AI is usually not about practicality:

Unfortunately, a lot of research into AI robots is fueled by the *strong sense* of “meaning” originating in a Searle style conception of the *Symbol Grounding problem*. This sense of meaning, however, can itself not be grounded! For any intelligent system, whether a virtual software agent or a physically embodied robot (including us humans), the environment presents itself as a set of *dynamic patterns* at the systemic interface (for instance, the sensory³ nerves). For all practical purposes, the universe is a pattern generator, and the mind “makes sense” of these patterns by encoding them according to the regularities it can find. Thus, the representation of a concept in an intelligent system is not a pointer to a “thing in reality”, but a set of hierarchical constraints over (for instance perceptual) data. The encoding of patterns that is represented in an intelligent system can not be described as “capturing true meaning” without the recourse of epistemologically abject realist notions; the quality of a world model eventually does not amount to how “truly” it depicts “reality”, but how adequately it encodes the (sensory) patterns.⁴

Even though the advocates of *Strong Symbol Grounding* are mistaken, and there is no epistemological reason why the set of patterns we associate with our concept of a physical universe (i.e. “real things”) and that we feed into our AI model should not originate in an artificial pattern generator (such as a virtual world), there are practical difficulties with purely virtual agents: Virtual environments tend to lack richness of presentation, and richness of internal structure.

Where experimenters specify virtual environments, they usually encode structures and details with certain pre-specified tasks and ontologies in mind, thereby restricting the AI agent situated in such an environment to the re-discovery of these tasks and

³ Note that the perceptual input of a system is completely made up of sensory input, for it can perceive its output only insofar it is provided by additional sensors. So, without loss of generality, sensory data stemming from sensor-actor coupling of a system are just a specific sub-class of sensory data in general. This is by no means a statement on how an AI system should treat sensor-actor coupling, however.

⁴ The adequacy of an encoding over the patterns that represent an environment can be measured in terms such as completeness, consistency, stability, sparseness, relevance to a motivational sub-system and computational cost of acquisition.

limited ontologies and depriving it of opportunities for discovery and invention. Hand-crafted virtual environments (such as virtual soccer [12] or role-playing game worlds) are probably much too simplistic to act as a benchmark problem for AGI. Limited real-world problems, such as robotic soccer or the navigation of a car through a desert, suffer from the same shortcoming. If we take our agents from the confines of a virtual micro-world into the confines of a physical micro-world, the presented environment still falls short on establishing a benchmark that requires AGI.

On the other hand, there are virtual environments in existence that sport both structural and presentational richness to a degree comparable to the physical and social world, first among them the World Wide Web. Even the ocean of digitized literature might be sufficient: Humankind's electronic libraries are spanning orders of magnitude more bits of information than what an individual human being is confronted with during their lifetime, and the semantics of the world conceptualized in novels and textbooks inherits its complexity from the physical and social environment of their authors. If it is possible for an intelligence system to extract and encode this complexity, it should be able to establish similar constraints, similar conceptual ontologies, as it would have while residing in a socially and physically embedded robotic body.

Robots are therefore not going to be the singular route to achieving AGI, and successfully building robots that are performing well in a physical environment does not necessarily engender the solution of the problems of AGI. Whether robotics or virtual agents will be first to succeed in the quest of achieving AGI remains an open question.

6. Build autonomous systems

As important as it is to integrate perception, memory, reasoning and all the other faculties that an intelligent system employs to reach its goals is integration of goal-setting itself. General intelligence is not only the ability to reach a given goal (and usually, there is some very specialized, but non-intelligent way to reach a singular fixed goal, such as winning a game of chess), but includes the setting of novel goals, and most important of all, about exploration. Human intelligence is the answer to living in a world that has to be negotiated to serve a multitude of conflicting demands. This makes it a good reason to believe that an environment with fixed tasks, scaled by an agent with pre-defined goals is not going to make a good benchmark problem for AGI.

The motivation to perform any action, such as eating, avoiding pain, exploring, planning, communicating, striving for power, does not arise from intelligence itself, but from a motivational system underlying all directed behavior. In specifying a motivational system, for instance as a set of conflicting drives, we have to make sure that every purposeful action of the system corresponds to one of its demands; there is no reason that could let us take behavioral tendencies such as self-preservation, energy conservation, altruistic behavior for granted – they will have somehow to be designed into the system (whereby ‘somehow’ includes evolutionary methods, of course).

7. The emergence of intelligence is not going to happen all by itself

While the proposal of AGI or *synthetic intelligence* is based on a computational monism,⁵ dualist intuitions are still widespread in western culture and in the contemporary philosophy of mind, and they are not going to give in without a fight. Because a naked ontological dualism between mind and body/world is notoriously hard to defend, it is sometimes covered up by wedging the popular notion of *emergence* into the “explanatory gap” [13]. Despite the steady progress of neuroscience and computational models of neural activity, there is an emergentist proposal that assumes so-called “*strong emergence*”, which proposes that the intelligent mind, possibly including human specifics such as social personhood, motivation, self-conceptualization and phenomenal experience, are the result of non-decomposable intrinsic properties of interacting biological neurons, or of some equally non-decomposable resonance process between brains and the physical world. Thus, “*strong emergence*” is basically an anti-AI proposal.

Conversely, “*weak emergence*” is what characterizes the relationship between a state of a computer program and the electrical patterns in the circuits of the same computer, i.e. just the relationship between two modes of description. In that sense, emergent processes are not going to “make intelligence appear” in an information processing system of sufficient complexity. We will still need to somehow (on some level of description) implement the functionality that amounts to AGI into our models.

This brief summary of principles of synthetic intelligence does not answer the main question, of course: How do we capture the functionality of Artificial General Intelligence? – In cognitive science, we currently have two major families of architectures, which seem to be hard to reconcile. One, the classical school, could be characterized as *Fodorian Architectures*, as they perceive thinking as the manipulation of a language of thought [14], usually expressed as a set of rules and capable of recursion. Examples, such as ACT [15] and Soar [16] are *built* incrementally by adding more and more functionality, in order to eventually achieve the powers inherent to general intelligence. The other family favors distributed approaches [17, 18] and *constraints* a dynamic system with potentially astronomically many degrees of freedom until the behaviors tantamount to general intelligence are left. This may seem more “natural” and well-tuned to the “brain-level” of description, because brains are essentially huge dynamical systems with a number of local and global constraints imposed on them, and the evolution of brains from mice-sized early mammalians to *homo sapiens* has apparently not been a series of incremental functional extensions, but primarily a matter of scaling and local tuning. Yet many functional aspects of intelligence, such as planning and language, are currently much harder to depict using the dynamical systems approach.

The recent decade has seen the advent of several new architectures in AI, which try to combine both approaches in a *neuro-symbolic* fashion, such as Clarion [19], LIDA [20], the MirrorBot [21] and the author’s own MicroPsi [22], which will briefly be introduced on the remaining pages.

⁵ Computational monism itself amounts just to the subscription of contemporary materialism. Cartesian matter (‘*res extensa*’) sports unnecessary intrinsic properties, such as locality and persistence, which get in the way when doing contemporary physics. Today’s matter is not the same wholesome solid as it used to be in Laplace’s time and day; now it is just a shifty concept that we apply to encode the basic regularities in patterns presented to us by our environment.

The Psi theory and the MicroPsi architecture

MicroPsi [22, 23] is an implementation of Dietrich Dörner's *Psi theory* of mental representation, information processing, perception, action control and emotion [24] as an AI architecture. MicroPsi is an attempt to embody the principles discussed above:

1. MicroPsi aims at explaining intelligence by a **minimal orthogonal set of mechanisms** that together facilitate perception, representation, memory, attention, motivation, emotion, decision-making, planning, reflection, language. These features are not explained as parameters or modular components, but in terms of the **function of the whole system**; for instance, emotions are explained as specific *configurations* of cognitive processing rather than as given parameters; behavior is not the result of pre-defined goal directed routines, but of a demand-driven motivational system and so on.
2. An integrated cognitive architecture will require the recourse to **methods from many disciplines**; MicroPsi originates in theories of problem solving in psychology and integrates ideas from gestalt theory, motivational psychology and experiments in the study of emotion. Also, it has learned a lot from cognitive modeling paradigms and from representational strategies and learning methods in AI.
3. The facets of cognition are not seen as separate modules that could be understood, tested and implemented one by one – rather, they are aspects of a broad architecture. The model **combines a neuro-symbolic theory of representation with a top-down/bottom-up theory of perception, a hierarchical spreading activation theory of memory with a modulation model of emotion, a demand/drive based theory of dynamic physiological, cognitive and social motivation with a model of the execution and regulation of behaviors**.
4. Representations in MicroPsi are always **grounded in environmental interaction** or abstractions thereof. It does not matter, however, if the environment is simulated or physical.
5. The difficulty of providing a rich pre-programmed environment vs. the limitations that come with robot engineering have lead to **both simulation worlds and robotic experiments** for MicroPsi agents. At the current stage of development, we seem to learn much more from simulations, though.
6. MicroPsi agents are **autonomous**, their behavior is governed by a **set of primary urges** which determine motives which in turn give rise to intentions. All behavior, including cognitive exploration and social interaction, can be traced back to one or more primary urges.
7. There are many aspects of intelligence that MicroPsi does not address well yet. However, we do not believe that these will be automagically spring into existence with gradual improvements of the learning, representation or interaction modes of the model. We think that the **deficits of MicroPsi** highlight specific mechanisms, for instance for perceptual integration, language and self-monitoring, that we have not sufficiently understood to implement them. MicroPsi might propose some interesting answers, but more importantly, it helps to detail a lot of **useful questions**.

Representation in the Psi theory

The most basic elements in Dörner's representations are threshold elements, which are arranged into groups, called *quads*. These are made up of a central neuron, surrounded by four auxiliary neurons acting as gates for the spreading of activation through the

network. A network of quads amounts to a semantic network with four link types, called *SUB*, *SUR*, *POR* and *RET*. *SUB*: stands for “has-part”. If an element *a* has a *SUB*-link to an element *b*, it means that *a* has the part (or sometimes the property) *b*. *SUR* is the inverse relation to *SUB* and means “is-part”. If *a* is *SUR*-linked to *b*, then *a* is a part (or sometimes a property) of *b*. *POR* (from latin *porro*) is used as a causal (subjunctive), temporal or structural ordering relation between adjacent elements. If *a* has a *POR*-link to *b*, then *a* precedes (and sometimes leads to or even causes) *b*. *RET* (from latin *retro*) is the inverse relation to *POR*. If there is a *RET*-link between *a* and *b*, then *a* succeeds (and sometimes is caused by *b*).

Quads make up perceptual schemas and frames: Individual quads stand for concepts. If they are *SUB/SUR* linked, a partonomic (has-part/is-part) relationship is expressed. The lowest level of such a partonomic tree is made up by *perceptual* neurons and *motor* neurons. They provide the grounding of the system’s representations, since they are directly linked to the system’s environment.

In MicroPsi, quads are extended to cover more basic relations and are expressed by *concept nodes*. In addition to the basic *POR/RET* and *SUB/SUR* links, they also offer link types for taxonomic and labeling relationships.

Object schemas are organized as parts of situational frames. The world model of the system is established, reinforced or changed by *hypothesis based perception* (“hypercept”). Hypercept is a paradigm that might be described as follows:

- Situations and objects are always represented as hierarchical schemas that bottom out in references to sensory input.
- Low-level stimuli trigger (bottom-up) those schema hypotheses they have part in.
- The hypotheses thus activated heed their already confirmed elements and attempt (top-down) to get their additional elements verified which leads to the confirmation of further *SUB*-hypotheses, or to the rejection of the current hypothesis.
- The result of hypercept is the strongest activated (matching) hypothesis.

At any time, the system *pre-activates* and *inhibits* a number of hierarchical schema hypotheses, based on context, previous learning, current low-level input and additional cognitive (for instance motivational) processes. This pre-activation speeds up the recognition by limiting the search space.

Hypercept is not only used on visual images, but also on inner imagery, memory content, auditory data and symbolic language.

The current situational frame is stored as the head element of a growing protocol chain, which is formed by decay and re-arrangement into long-term memory.

Behavior Execution

A neuron that is not part of the currently regarded cortex field is called a *register*. Neural programs are chains of registers that call associators, dissociators, activators and inhibitors. (These “calls” are just activations of the respective elements.) In the course of neural execution, elements in the cortex field are summarily linked to specific registers which are part of the executed chain of neurons. Then, operations are performed on them, before they are unlinked again.

Dörner describes a variety of cognitive behaviors for orientation, anticipation, planning, memory retrieval and so on, often along with possible implementations. [24, 25]

Motivation and Emotion

During action execution, the system establishes motives based on a set of primary urges which are hard-wired. Currently, these urges consist of demands for *fuel*, *water*, *intactness*, *affiliation*, *competence* and *certainty*.

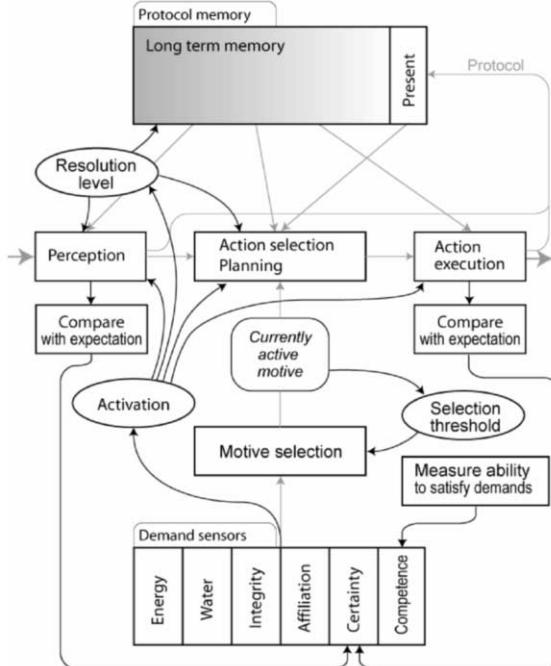


Figure 2: Psi architecture

Fuel, water and intactness are examples of physiological needs. Affiliation is a *social urge* – to satisfy it, the system needs *affiliation signals* from other agents. Thus, Psi agents may reward each other. *Competence* measures the ability to reach a given goal and the ability to satisfy demands in general (coping potential). The urge for *certainty* is satisfied by successful exploration of the environment and the consequences of possible actions, and it is increased by violations of expectations. Competence and certainty are *cognitive urges*. Together they govern explorative strategies.

Every increase of an urge creates a negative reinforcement signal, called *displeasure signal*; conversely, a decrease of an urge results in a *pleasure signal*. These signals are used to strengthen links in the current protocol and thus enable reinforcement learning of behavior. At any time, the system evaluates the urge strengths and, based on an estimate of the competence for reducing individual urges, determines a *currently active motive*. This motive pre-activates memory content and behavior strategies and is used in determining and executing a plan to achieve it.

To adapt the cognitive resources to the situation at hand, the system's activity is influenced by a set of *modulators*. Within the Psi theory, a configuration of modulator settings (together with appraisals of certainty, competence and current pleasure/displeasure status) is interpreted as an emotional state [25].

The MicroPsi Framework

MicroPsi has been developed in an AI context and offers executable neural representations, multi-agent capabilities and visualization tools. The author's group has used it to model perceptual learning [26], the evolution of motivational parameters in an artificial life setting and as an architecture for controlling robots.

The framework consists of the following components: a graphical editor for designing executable spreading activation networks (which make up the Psi agent's control structures and representations), a network simulator (integrated with the editor and monitoring tools to log experiments), an editor and simulator for the agent's environment and a 3D viewer which interfaces with the simulation of the agent world.

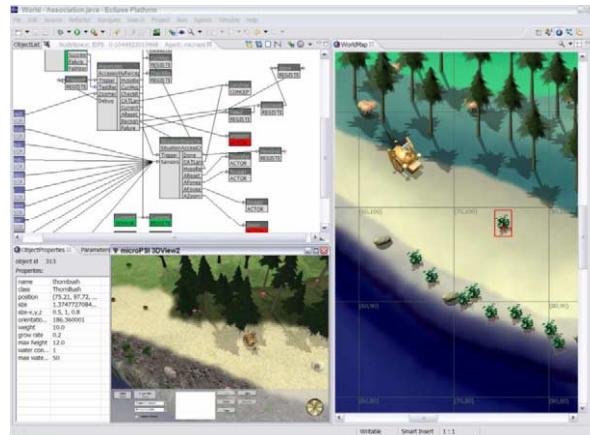


Figure 3: MicroPsi network editor and agent environment

MicroPsi has matured into a runtime environment for various cognitive modeling tasks. Among other things, we are using it for

- **Building agents according to the Psi theory.** These agents are autonomous systems with a set of innate urges, situated in a simulated environment, where they perceive objects using a simplified *hypercept* approach. Perceptual content is represented as partonomic schema descriptions and arranged into protocols, which are later retrieved to generate plans to satisfy the agent's urges.

- **Performing neural learning using hybrid representations.** To connect higher, gradually more abstract layers within MicroPsi network representations to real-world sensory input, we are setting up a matrix of foveal sensor nodes, which correspond to pixels in the camera of a robot. By moving the foveal field through the camera image, the image is scanned for salient features. Using backpropagation learning, we are training it to identify edge segments in the camera image, which in turn make up the lowest layer in an instantiation of *hypercept*.

- **Evolving motivational parameter settings in an artificial life environment.** Here, groups of MicroPsi agents jointly explore their territory and cooperate to find and defend resources. Suitable cooperative behaviors are evolved based on mutations over parameters for each urge and modulator in accordance to the given environment.

- **Implementing a robotic control architecture using MicroPsi.** A simplified neurobiological model of behavior and perception of mice in a labyrinth is mimicked using Khepera robots that are embodied MicroPsi agents.

Since its beginnings in 2001, the MicroPsi framework and the associated cognitive architecture have come a long way. Even though MicroPsi is far from its goal – being a broad and functional model of human cognition and emotion – it fosters our understanding and serves as a valuable tool for our research.

Acknowledgments

This work has been supported by the University of Osnabrück. I am indebted to Ronnie Vuine, who vitally contributed to the technical framework and the MicroPsi agents, to Matthias Füssel and Daniel Küstner, who built the environment simulator, to David Salz who is responsible for 3D visualization components, to Markus Dietzsch, who is performing the artificial life experiments, and to Daniel Weiller and Leonhardt Laer, who are in charge of robotic experiments. Also, I want to thank two anonymous reviewers for their constructive criticism.

References

- [1] Jackson, F. (1982). Epiphenomenal qualia. *Philosophical Quarterly* 32: 27-36
- [2] Block, N. (2001). Paradox and Cross Purposes in Recent Work on Consciousness. In Dehaene and Naccache: Special Issue of *Cognition*, Vol. 79, The Cognitive Neuroscience of Consciousness, 197-219
- [3] Chalmers, D. (1996): The Conscious Mind. New York: Oxford University Press
- [4] Searle, J. R. (1992): The Rediscovery of the Mind, MIT Press, Cambridge
- [5] Dreyfus, H. L. (1992): What Computers still can't do. A Critique of Artificial Reason. MIT Press
- [6] Harnad, S. (1990). The symbol grounding problem. *Physica D*, 42, 335-346.
- [7] Barsalou, L.W. (1999). Perceptual symbol systems. *Behavioral and Brain Sciences*, 22, 577-609.
- [8] Brooks, R. A. (1991): Intelligence Without Reason, IJCAI-91
- [9] Searle, J. R. (1980): Minds, brains, and programs. *Behavioral and Brain Sciences* 3 (3): 417-45
- [10] Pfeifer, R., Stieler, W. (2007): „Es geht um den physikalischen Prozess.“ Technology review, online at <http://www.heise.de/tr/artikel/95539>, Oct 1st 2007
- [11] Pfeifer, R., Bongard, J. (2006): How the body shapes the way we think. MIT Press
- [12] Noda, I (1995). Soccer Server: A Simulator of Robocup. In Proceedings of AI symposium 1995, Japanese Society for Artificial Intelligence.
- [13] Stephan, A. (1999). *Emergenz: Von der Unvorhersagbarkeit zur Selbstorganisation*. Dresden Univ. Press
- [14] Fodor, J. A. (1975). *The Language of Thought*, Cambridge, Massachusetts: Harvard University Press.
- [15] Anderson, J. R. (1990). *The adaptive character of thought*. Hillsdale, NJ: Erlbaum.
- [16] Laird, J. E., Newell, A., & Rosenbloom, P. S. (1987). Soar: An architecture for general intelligence. *Artificial Intelligence*, 33(1), 1-64.
- [17] Rumelhart, D. E., McClelland, J. L. and the PDP Research Group (1986): Parallel Distributed Processing, (Vols. 1&2), Cambridge, Massachusetts: MIT Press
- [18] Smolensky, P., Legendre, G. (2005): The Harmonic Mind. From Neural Computation to Optimality-theoretic Grammar, Vol. 1: Cognitive Architecture, MIT Press
- [19] Sun, R. (2003) A Detailed Specification of CLARION 5.0. Technical report.
- [20] Franklin, S. (2007). A foundational architecture for Artificial General Intelligence. In Advances in Artificial General Intelligence: Concepts, Architectures and Algorithms, Proceedings of the AGI workshop 2006, ed. Ben Goertzel and Pei Wang: 36-54. Amsterdam: IOS Press.
- [21] Borst, M., Palm, G. (2003): Periodicity Pitch Detection and Pattern Separation using Biologically Motivated Neural Networks. In: Proc. 6. Tübinger Wahrnehmungskonferenz, 52
- [22] Bach, J. (2003). The MicroPsi Agent Architecture. Proceedings of ICCM-5, International Conference on Cognitive Modeling, Bamberg, Germany, 15-20
- [23] Bach, J. (2008): PSI – An architecture of motivated cognition. Oxford University Press.
- [24] Dörner, D. (1999). *Bauplan für eine Seele*. Reinbeck
- [25] Dörner, D., Bartl, C., Detje, F., Gerdes, J., Halcour, (2002). *Die Mechanik des Seelenwagens. Handlungsregulation*. Verlag Hans Huber, Bern
- [26] Bauer, C. (2005): Kategorielernen im MicroPsi-Agenten. Diploma Thesis, Technische Univ. Berlin

Language Processing in Human Brain

Alexander BORZENKO ¹
Proto-mind Machines Inc., Toronto, Canada

Abstract. Human brain is exceptionally complex and simple at the same time. Its extremely composite biological structure results itself in human everyday behavior that many people might consider rather simple than complex. In our research we will concentrate on the ways how a human brain can processes English and other human natural languages because taken in general sense the ability to speak English or other human languages is only serious distinguishing feature that rises humans over the rest of the world making a human an intellectual being. On the purpose of our research we consider natural language as naturally formed symbolic system completely independent of these symbols' physical nature that is a little more general than a common natural language definition. The principles of natural language processing in human brain are most important for us if we want to build equally powerful artificial general intelligence. We start with the features of human brain neurons and neural networks, and step by step create a computer model of human brain networks that is able to process and generate a reasonable speech. We can't give a detailed explanation of human brain functionality in this short article. Moreover, it is not our goal, and such research is not complete yet. The main result of our research is revealing the principles how tiny single neurons working together can produce intellectual-like behaviour that exhibits itself in proper speech comprehension and generation in accordance with current context.

Keywords. Neuron, brain, neural network, intellect, language, AGI, human, mind.

1. Neurophysiologic Background

Intellect is a direct property of brain cerebral activity according to the Nobel Prize laureate R. Sperry [1]. From other side, human intellect is bound to natural languages that humans speak. Natural languages spoken by humans are in essence the only language system that provides comprehensive description of the world. It is obvious that the features of neural networks of human brain restrict the natural language basis. Moreover the basic operation set of natural language depends on universal methods of data transformation in real neural networks. We will show that taking the general features of human brain as base we can design a virtual device that has human intellectual abilities like English speech understanding and producing relevant speech answers. Everything can be built over English language then, even, for instance, such complicated abstract theories like theory of intellect [1] or constructive mathematics

¹ E-mail: alex.borsen@gmail.com

theory [2] because there is no other language or extra language means in the basics of these theories save English language.

Let us consider some general facts about human brain structure and neuron physiology. These facts would not be mentioned here if they didn't complete a base of our brain model. For the beginning, our model consists of a huge number of mutually linked independent information units without any central control like natural brain does. Those elementary information units of human brain are neurons. Neurons are connected to each other through their axons and dendrites that cover the large areas of human brain according to other Nobel Prize winner Santiago Ramon Cajal [3] who discovered this feature about a century ago. We are interested in two main neuron structures of a human brain: Neocortex and Cerebellum that both have a great similarity from the information point of view [4]. The number of neurons in these human brain structures is huge that is of great importance for us (our statistics will work perfect with such significant amount of data). Current estimates suggest that there may be on the order of 10^{10} - 10^{12} neurons and of 10^{16} synapses per person. Each neuron has about 1000 connections on average [5] forming a kind of regular or, sometimes, granulated structure.

Natural neuron consists of cell body, number of dendrites, and one axon (generally we are not interested in internal neuron structure). Most of neurons consist of a cell body plus one axon and many dendrites. The axon is a protuberance that delivers neuron output signals to connections with dendrites of other neurons or with neuron cell bodies. Dendrites are other protuberances that provide plenty of surface area, facilitating connection with the axons of neurons. Dendrites often divide a great deal, forming extremely bushy dendrite trees. Axons divide to some extent but far less than dendrites. We don't reflect physical neuron structure in our model making the following generalization – our model elementary unit has many input signals and only one consolidated output signal that can be passed to many other model units.

There are several types of signals transmitting from one neuron to others as well as different types of data in human brain neurons [6, 7, 8]. We will abstract of the physical nature of mentioned processes taking in a view only their general external information features, in particular an impulse manner of signals.

Usually neuron does nothing active unless some function (we assign the letter Ψ to it) of collective influence of all neuron inputs reaches a threshold level. Whenever that threshold level is reached, the neuron produces an output in the form of a short pulse that proceeds from the cell body, down the axon, and into the axon's branches. When this happens, the neuron is said to fire. From the external point of view the general manner of neuron firing seems to be stochastic except the case when neuron keeps firing during a certain period after it fired the first time. Sometimes neuron does not fire even if its threshold measured or estimated in the previous experiments is reached. Sometimes it fires spontaneously having no visible external factors to fire. This means that function Ψ belong to a stochastic type. In addition, the neuron spontaneous activity (also of stochastic type) plays an important role in brain information processes, and it should be considered in our model through the features of function Ψ .

Axons influence dendrites over narrow gaps called synapses that are of a complex thin structure itself. Stimulation at some synapses encourages neurons to fire. Stimulation at others discourages or blocks neurons from firing. A complex stimulation provides answer that is hard to predict without exact information about states of neuron synapses and their dynamics. A lot of research was done to reveal the details of human brain synapse and neurons behavior [9, 10, 11, 12, 13, 14]. Nowadays the most of

neuroscience researches confirm each other in general. There is mounting evidence that learning takes place in the vicinity of synapses and has something to do with the degree to which synapses translate the pulse traveling down one neuron's axon into dendrites of the next neuron or neurons. These results mean that function Ψ is not constant but changes itself to support learning. However we will not create an exact model of synapse functionality. In our model we will use only general synapse features that were just mentioned above describing them through the function Ψ .

Brain neuron behavior was generalized in a simple information model [15, 16] which would help us in further analysis. Let E_k^t be an event that some neuron has its synapse of number k activated at the time t . Then, supposing that zero time corresponds to the last time when function Ψ has changed itself, a probability P of the neuron firing can be symbolically described as following

$$P = \Psi(E^t, E^0, U, M, N), N < M, \quad (1)$$

where E^t and E^0 are vectors of synapse activity at time t and 0 respectfully, U is a probability that a neuron changes its reaction during one second due to the phenomenon of spontaneous activity, M is a total number of this neuron synapses, and N is a minimal number of active neuron synapses that are necessary to reach the neuron's threshold. In other words, neuron can make decision to fire if only N of its M synapses are in the states they were at zero time.

The neuron synapse function (we will use the letter Ψ for it) plays a key role in brain neuron functioning (e.g. in converging external synapse activity into neuron firing with a respect to previous history of neuron activity in a time-spatial form) [30]. It is obvious that function Ψ is extremely complex and requires a powerful computer or even some advanced computer system to make an exact model even for a single neuron. Instead of using that approach we make only one natural assumption about Ψ :

Principle 1. If there is a configuration of dendrite activity (regardless of its type) and the neuron is active (e.g. firing) the function Ψ modifies itself in the way that the neuron should fire on this particular or similar dendrite activity configuration in future.

This fundamental principle can be named as *principle of activity maintenance*.

This function Ψ behavior leads to a recently confirmed fact that the same image repeatedly shown to human inspires activity of the same neurons in his brain. In its turn it leads to more complicated behavior when these activated neurons belong to one of specialized zones of the human brain, in particular, to the speech zone of the brain. In this case the human brain generates speech as a result of intellectual processing its input data. Principle 1 presumes that some physical process happens when brain memorizes information, and this process happens in the vicinity of neuron contacts (synapses mainly) because it is a place where activity signals of both contacting neurons meet. According to [11] it is probably the only place where such process occurs.

The next principle of simplifying Ψ deals with a neuron spontaneous activity that was discovered and confirmed by many independent researches a long time ago. Spontaneous activity seems to be a mere noise producing by a neuron but really it is a fundamental feature. The second principle of brain functioning is:

Principle 2. There is a finite probability that any brain neuron fires independently of its current external configuration, history of firing, and the current states of its dendrite and axon inputs.

Further we will refer to this principle as a *principle of neuron spontaneous activity*.

In general our understanding of natural neuron functioning contains nothing more than:

- Neuron has two states: active when it fires and passive when it does nothing visible from the outside,
- Neuron can recognize the similarity of current synapse activity of other neurons connected to it and can react with firing on similar configurations of current synapse activity,
- Whenever active the neuron has a power to share its efforts with other neurons in order to activate or to suspend the activity of some other neurons including itself (actually knowing nothing about possible consequences of this activity),
- Neuron thoroughly follows principles 1 and 2.

Paying an undeniable respect to the researchers devoted their time to discover the details of natural neuron physiology nevertheless we have to admit that those details give us almost nothing about understanding the structure of the human intellect foundation.

Our aim is to prove that principles 1 and 2 lead to intellectual behavior if we speak about humans. In order to prove a memory phenomenon in neural networks of human brain we construct a mathematical model of natural neural network using principles 1 and 2. The deductions from this model form the basis for analysis of natural languages in a human manner.

This abstract model of natural language processing in relation to the respective human brain methods helps us to build more sophisticated models and check the different aspects of natural language processing in a human brain.

Finally, we build a complete abstract model - *General Sense Linker* (GSL) - of human brain methods for English language processing and synthesizing through generalization of some intermediate models that we will develop and also through direct implementation of model of the object that function Ψ deals with. GSL learns English like a child who forms proper speech ability by himself using his/her parents' and teachers' lessons and examples. So, in order to do this, GSL should be provided with necessary conditions and examples for the automatic creation of output reactions on a contextual base.

After learning, GSL becomes almost fluent in English (certainly in the domain it was taught). This means that GSL's natural language responses on given natural language stimuli can be predicted in a current context by taking the similar human responses as a base. At the same time this GSL's behavior serves as a proof of mentioned above simple principles 1 and 2 of brain functioning because no extra fundamental features are used in GSL model save activity maintenance and spontaneous activity principles.

2. Natural Neuron network model

Let us consider a group of mutually connected neurons that we call *column*. All column neurons comply with the principles 1 and 2. We accentuate this fact by using italic font when mentioning the object *column*. In a general case the internal *column* neuron connections are supposed to be of some restricted chaos type that makes columns' information properties vary. Extern *column* connections consist of the set of column neurons' connections to the neurons that don't belong to the *column*. We divide extern

column connections onto input and output parts (that parts can have connections in common sometimes). We consider the following virtual experiment in order to estimate a stability of column neuron reactions.

Suppose we have a neuron with number p (p -neuron) reacted with firing on some spatial configuration of activity coming from the external neurons to the dendrites of *column* neurons. In more precise, “spatial activity” means that we have a combination of external neuron signals constantly coming to our column neurons during some critical time period. This period is supposed be long enough to let column neurons to process these signals.

Formula (2) below shows probability x_p^t of the event that a *column*’s p -neuron keeps its proper reaction on the concrete image I_m exposed to the *column* exactly t seconds ago. By other words, p -neuron loses this reaction at the second with number t if the image I_m has been exposed at a zero time point. The reason why this p -neuron changes its reaction can be explained through the *activity maintenance principle*: this neuron spontaneous activity occurred and this neuron merely amplifies its active dendrite ties when any other image was exposed to the *column* at this time point (like the results of research [28]).

Supposing that every *column* neuron needs an exact signal configuration on its dendrites to recognize the image (or by other words, dendrite activity configuration) we have the following equation for probabilities x_p^t in the connection to other neural network parameters.

$$x_p^t = \left(\prod_{r=1}^{t-1} (1-x_p^r) \right) \cdot \{U + (1-U)[1 - \prod_{i \in Q(p)} (1-x_i^{t-1})]\}, \quad (2)$$

where U is a probability that a *column* neuron changes its reaction during one second as it was defined in formula (1). Naturally, probabilities x_p^0 equal zero. The set $Q(p)$ is designated to the variety of numbers of *column* neurons that provide p -neuron with relatively significant influence.

Let us take a look at figure 1 with a diagram of the numerical evaluation of the equation (2) under assumption that no neuron in a *column* has advantage over others, q equals 182 (we supposed it to be the estimated average size of human brain neural “column”) and U equals 10^{-7} (this probability means that 10,000 human brain neurons of Neocortex change their internal memory every second on average).

We need some additional model that can summarize the results of image remembering and recognition in our *columns* to create a consolidated reply like a human brain that generates

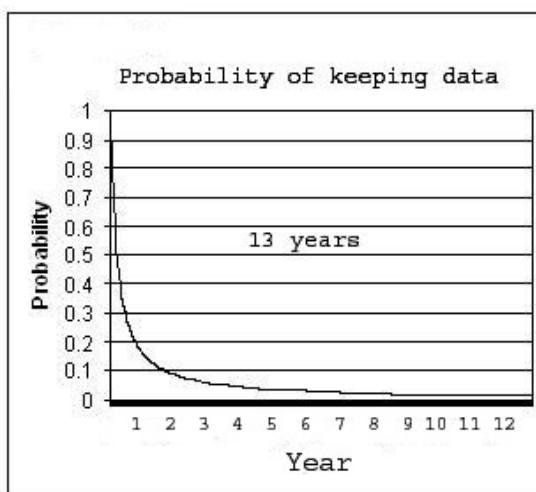


Figure 1. Probability of keeping the data in the Model

consolidated answers. Fortunately we know that there is a brain Neocortex part with a plenty of giant pyramidal and motor neurons which generate signals resulting in speech. In this part there are outgoing peripheral brain nerves that control speech [11].

These pyramidal neurons collect and summarise signals of other neurons through their giant ramified dendrites covering the vast regions of a human brain. Many *column* neurons of some types with long horizontal-spread axons help them. Such pyramidal neurons regulate human speech facilities directly through generating signals to the throat and other muscles that are involved in a speech producing.

We introduce nothing new in pyramidal firing estimation. Formula that provides us with the means for estimating a probability of pyramidal neuron firing through vectors E^t of its synapse activity (formula (1)) is

$$\sum_{k \in M} A(E_k^t, E_k^{0_k}) * W_k > \lambda,$$

where $A(\cdot, \cdot)$ is a function of two arguments that returns 1 if its arguments are equal, and 0 otherwise; λ is some threshold, and W_k are some weights associated to correspondent synapses (these weights are not constants, they vary according to the principles 2 and 3).

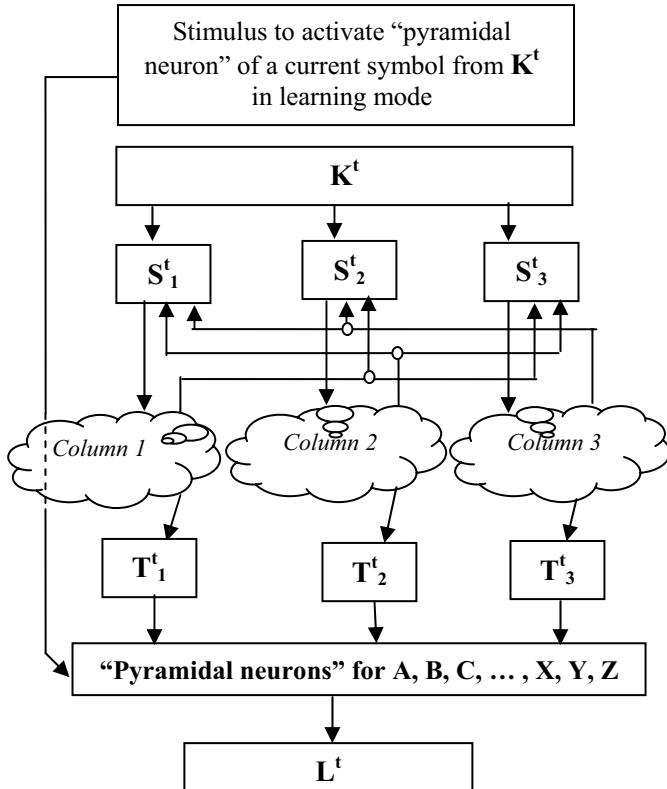


Figure 2. General Principle of Natural Language Phrase Processing in the Model

Unfortunately, a *human-like intellectual behavior* is a vague concept. Everybody understands it but can't express it formally at the same time. Broadly speaking, human is able to learn in any possible knowledge domain either quantum physics or car renting or almost any other field. From the external point of view this ability reveals itself in *generating relevant reactions on an appropriate input* in most cases. In more general, every generated English phrase should be consistent and should correspond to its predecessors. By other words, speech should be interconnected. We will believe that we have successfully proved that our abstract model has a human-like intellect if we will be able to show a similar behavior of our model.

We will consider many *columns* concurrently as a *column* system united by innumerable mutual ties of chaotic type (as it is shown formally by clouds on figure 2). This *column* system will be a core of our neuron network model. Statistical characteristics of the chaos are critically important as usual when any neuron network model is built. We start with the simplest probability distributions (like normal) keeping in the mind to find the most effective ones through experiments with our computer model.

As far as English language is concerned we will consider here only English letter and grammar symbol sequences $\{K^t\}$ as a brain model input no matter how they are delivered to brain either through vision, hearing or other relevant channel. Physically any peripheral neurons could deliver the consequences of signals that correspond $\{K^t\}$ through their ramified axons. Moreover we don't suppose to have functional level dependence here. Statistical one is enough for us. Our model accepts $\{K^t\}$ symbol by symbol having all information about the current symbol $C(K^t, t)$ processed before accepting the next symbol.

In its turn, $\{L^t\}$ stays for English letter or grammar symbol output of our model. We correlate it to the most active "pyramidal neuron" at the time (fig.2). Here the superscript index t corresponds to the time interval when actual symbol/letter was generated (or was entered in the case of $\{K^t\}$) in our neural *column* system.

The form of input doesn't matter for us because regardless of external symbol form the peripheral nerve system delivers information to brain as a combination of dendrite impulse activity. So what we can be sure in is that this dendrite activity correlates statistically with input symbols, and we need nothing more to know about input/output in our investigation.

We consider two modes of our model functioning: learning and speaking. At the first mode we have $\{K^t\}$ not empty, so our model has its current symbol anytime. At the second mode $\{K^t\}$ is empty, and we suppose that current symbol is $C(L^t, t-1)$ (model listens to its own "speech").

Let us number all neural *columns* using any appropriate method and let S_i^t be an input of any particular *column* with number i at the time point t , and then let T_i^t be the according *column* output that is a combination of *column* neuron axon activity. We also consider T_i^t as some numeric measure of intensity of summarized *column* neuron activity in a relevant phrase context (the most frequent case is that there is not more than one active neuron in a *column* at any time point). In many cases just a numeric sum as measure is sufficient enough.

In the relation to our *columns* each variable S_i^t means that there are some synapses on i -*column* neurons those synapses' weights are correlated somehow to letters and symbols those the model percepts and those are described by variable K^t . Formally S_i^t

includes some valuable subset of K^t and feedbacks of previous neuron states of i -*column*.

All ties on the figure 2 provide not instant data transfer because natural axons and dendrites have different lengths and thickness, and sometimes because of different chemical environment in real neuron networks, feedback ties and because of the general inertia of a neuron functioning. These features are confirmed by many researches, for example, recent developments [25, 26, 27]. Action potentials travel at up to 10 m/s speed in unmyelinated axons and up to 120 m/s speed in myelinated axons [5]. Taking in a view these signal transferal delays, each T_i^t is a complete mess of *columns*' neuron signals (in the form of impulses) if we consider them as an input sequence of some network neuron. One particularity of this "mess" is extremely important. *Column* will repeat this "mess" (configuration of outputs of its neurons) with the same statistical characteristics exactly if this *column* input is the same because *column*'s signal "mixing" algorithm statistically depends on dendrite-axon ties, length and thickness of dendrite and axons and some other constant factors that has been indirectly confirmed by many experiments mentioned in the first paragraph.

Finally, we constitute a simple but effective method to create a generalized output L^t of "pyramidal neurons". We select a letter or a symbol of that "pyramidal neuron" which has currently a maximum activity level among all others as actual model output leaving the real algorithm for further investigation.

Nevertheless we use efferent and afferent signals in our model (the latter is a prototype for stimulus signal in fig. 2) in the learning mode. We suppose that efferent nerves have their corresponding afferent partners or the stimulation of a "pyramidal neuron" matching the current symbol can be done through efferent influence on receptor activity [31]. In our model this "afferent" direct or indirect stimulus provides additional activity to "pyramidal neuron" of a current letter when we teach our model.

3. COMPUTER-ADDED EXPERIMENT

It took about a year of intensive interactive communication with computer implementation of our abstract model (GSL) in order to achieve expected results. During this period the necessary statistics in the model were completed. The method that was used to teach the model plays a secondary role and therefore is listless here. The text material that was used to teach GSL consists of some child books, scientific articles, many dialog fragments of the type you can find in literature and Internet chats, on-line scientific, sport and politics news, and even thesaurus articles. A lot of direct interaction was done with GSL at the final stages of learning when reliable answers of GSL became common.

The experiment has proven that diversity of input texts and dialog topics leads to better results. Certainly the final model is not a comprehensive and complete artificial intelligence because of lack of knowledge and some data input/output stream simplifications (we had only one input stream and one output stream controlled by high level model monitor). Fortunately it was not our aim to create an artificial intelligence.

Bellow there are some samples of a dialogue with GSL model at the different stages of learning process (**G**: stays for free speech generation by GSL, **Q**: means a question, provided by teacher, and **A**: stays for GSL's answer to the question above).

GSL learning process passed through several stages. Each stage required its own settings and learning method specific modifications. Accordingly, the model behavior

was improved stage by stage starting with exact answers to exact simple questions without taking the current context into account at primary stages, and to context speech generating at intermediate stages, and to free form answers to general questions in a complete conformity with context at final stages. Actually all these three stages are interconnected and differ only in a degree of context involving that in its turn is a result of amount of data that were accumulated in the model. Let us consider them consequently one by one.

Primary Stages (1,000,000 columns)

Q: *What's your name?*

A: *Alex*

Q: *How old are you?*

A: *I am a young man*

Q: *Who are you?*

A: *I'm a scientist*

At this stage the model was directly taught all these question-answer couples. It took several rehearsals for each question-answer couples before the model began to react properly.

Intermediate Stages (3,500,000 columns)

G: *He took no notice of this*

G: *He looked across of my companion who was also an elderly*

G: *I was away in a cab and down the platform, with what you had better give me much more clever than average man*

G: *I think that was to details of glance into the fire*

The model was not taught anyone of these sentences at any stage. No answers given by GSL were entered in the model in the exact form they were received. GSL generated them using its general knowledge collected throughout learning period in the context created by examiner especially for each sample. Also GSL computer model has read several books of famous English-speaking writers of last centuries. At the second intermediate stage it is common that all four sentences are of a correct syntax with some semantic disturbances that can be explained by a lack of knowledge in the model.

Final Stages (6,100,000 columns)

Q: *What is a sea floor?*

A: *A floor is generally a lower horizon*

Q: *What do you think about the mystery island colonists?*

A: *Poor wreckers! They were in a great danger*

Last answers look quite correct if we take in account that the context was the old trilogy of famous French child writer (in English translation) about underwater travel, oceans and isolated islands.

It should be especially mentioned that number of *columns* involved by teaching the model through dialogues and readings is not in a linear statistical dependence of processed data volume. The process of *column* number growth (our model has ability to create *columns* automatically when it is necessary) slows down with any portion of data that contains really new facts. For instance, the data volume processed at final

stages is about ten times greater than similar amount at the intermediate stages. But at the same time the column amount in GSL model was only doubled (find these numbers above).

Table 3 contains a table with the learning stages against approximate numbers of neuron *columns* which activity is above average level, durations of the stages and intensity of new *column* neuron activation.

Table 3. Active *column* dynamics

Stage	Amount of columns	Relative Duration	Average Intensity
<i>Primary</i>	1,000,000	1	1.00
<i>Intermediate</i>	3,500,000	5	0.70
<i>Final</i>	6,100,000	14	0.43

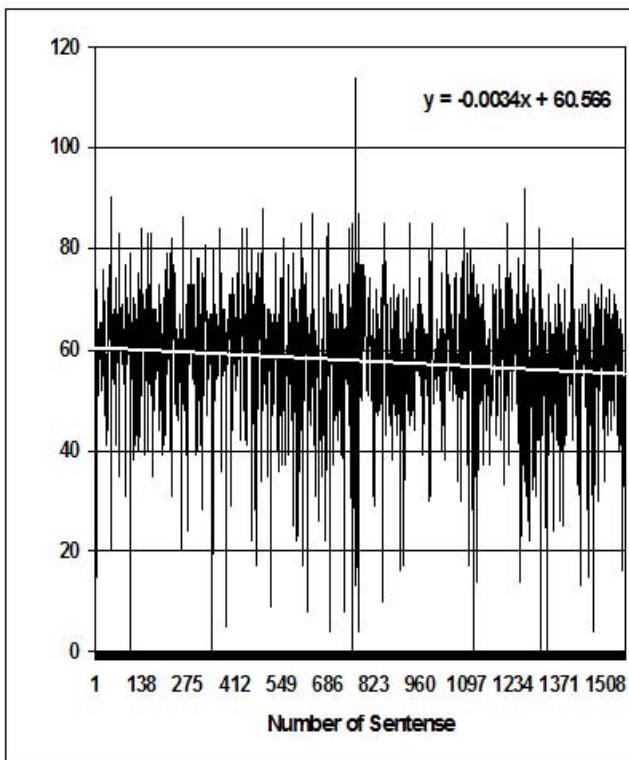


Figure 3. *Column* Involving Trend

model of natural brain neuron network that was created in order to explain how human brain works when processing the natural languages. Nevertheless GSL can be used as a language processor with not-symbolic internal knowledge representation. It is pretty effective to generate answers in a real time.

Figure 3 illustrates the trend of the intensity of new *column* neuron activation against the number of processed sentences of some brief novel. The trend shows that intensity (y-axis) drops very slow but stable. Intensity is measured in average number of newly involved neurons (number of neurons per sentence length * 100) against the sentence amount.

When analyzing the results of experiments with *General Sense Linker* we keep in mind that *GSL* is NOT artificial intelligence (there is a long way to find out even what is a natural intelligence [29] saying nothing about artificial one). *GSL* is a

4. CONCLUSIONS

We discussed neither what a human intellect is nor how it could be built from not intellectual parts. It was enough for us that in spite of revealing itself in many different ways a human intellect always uses a simple ability of our minds to give natural answers to natural language questions in accordance with the questions' common sense and current context. We have shown this fundamental intellectual feature of our virtual device GSL that we created on the base of principles 1 and 2 and broad-known facts of human brain structure and physiology using them in a more or less constructive manner.

We have created GSL device that is capable to learn and speak English, and this device consists of a variety of independent comparably simple units which work without any synchronization and supervision like neurons in a natural brain. They are self-organizing and self-control units similar the brain neurons. Macro blocks of them that we called *columns* show the memory abilities, and we proved that *columns* can keep information as long as years at least.

GSL provides *columns* with input data and incorporates the results of *column* information processes into generalized output working similar the brain pyramidal neurons. So GSL device has a lot of features in common with human brain structures. For instance, knowledge is spread among the total set of neurons. So destroying some group of neurons doesn't result in destroying GSL's knowledge, it just becomes less solid.

Certainly it requires time to elaborate details and to persuade us that this device works having the same principles in its base as human brain has, and therefore explains how the latter works. But it can't stop us from using those principles and GSL device itself for many practical applications.

At the next stage of our research we will deal with visual input channel. Human vision is a complicated highly specialized apparatus that is tightly collected to the human general intelligence. So this research will use already created theory involving simple primary visual image recognition methods. We will show how video data that are not compact and grammar-structured like speech can be processed by our final GSL model having a similar result – generating the relevant answers to appropriate video images taken in a current context.

References

- [1] Sperry R.W. "A modified concept of consciousness". In: Psychological Review, vol. 76, N 6, 1969, p.332-336.
- [2] Markov A. "An approach to constructive mathematical logic". Logic, Methodology and Philosophy of Sciences, III, 1968, Amsterdam.
- [3] Santiago Ramon Cajal. "New Ideas on the Structure of the Nervous System in Man and Vertebrates", 1894, Bradford Books, MIT Press.
- [4] Nieuwenhuys R., Donkelaar H., Nicholson C. "The Central Nervous System of Vertebrates", 1998, Springer-Verlag.
- [5] R.W. Williams, K. Herrup. "The control of neuron number". Annual Review of Neuroscience, 11, 1988, 423-453.
- [6] Paul Katz (Editor) "Beyond Neurotransmission. Neuromodulation and its Importance for Information Processing", Oxford University press, 1999.
- [7] Han, X. and Jackson, M. B. "Electrostatic interactions between the syntax in membrane anchor and neurotransmitter passing through the fusion pore". Biophysics, 2005. Letter. 88: L20-22.

- [8] Power, J.M., Oh M.M. and Disterhoft, J.F. "Metrifonate decreases sI (AHP) in CA1 pyramidal neurons in vitro". *J. Neurophysiol.* 85, 2001: 319-322.
- [9] Idan Segev "Single neuron models: oversimple, complex and reduced". In: *Trends in Neurosciences*, Vol. 15, No. 11, 1992, p.414-421
- [10] Mojarradi M., Blinkley D., Blalock B., Andersen R., Ulshoefer N., Johnson T., and L. Del Castillo "A miniaturized neuroprosthes is suitable for implants into the brain". *IEEE Transactions on Neural systems and Rehabilitation Engineering*, 11, 2003:1534-4320.
- [11] Milner P. "Physiological Psychology". Holt, Rinehart and Winston, Inc. New York, 1970.
- [12] Luria A. "The origin and cerebral organization of man's conscious action". In: *Proceedings of the 19 congress of psychology*, 1971, London.
- [13] Peter H. Lindsay, Donald A. Norman. "Human Information Processing". Academic Press, New York and London, 1972.
- [14] Jones, T., Greenough, W. T. "Ultra structural evidence for increased contact between astrocytes and synapses in rats reared in a complex environment". *Neurobiology of Learning & Memory*, 65 (1), 1996, 48-56.
- [15] Borzenko A. "Formal Neuron System for the Natural Language Analysis". In: *IEEE Neural Networks Proceedings*, 1998, p.2561-2564.
- [16] Borzenko A. "Associative Recognition of Signals by a Network of Formal Neurons". In: *Automatics and Mechanics*, No. 1, 1985, Moscow, Science, p. 95-100.
- [17] Eliot C. Bush, John M. Allman. "The Scaling of White Matter to Gray Matter in Cerebellum and Neocortex". *Brain, Behavior and Evolution*, Vol.61, N 1, 2003.
- [18] Jan Woogd, Mitchell Glickstein. "The Anatomy of the Cerebellum". *Neurosciences*, September 1998, Vol. 21, № 9.
- [19] Kaas J.H. "The organization of Neocortex in mammals: Implications for theories of brain function". *Annual Review of Psychology*, 38, 1987:129-151.
- [20] Hubel D.H. "Eye, Brain and Vision". *Scientific American Library*, No. 22, 1995, WH Freeman, NY. p. 70.
- [21] Hubel D.H., Wiesel T.N. "Brain and Visual Perception". Oxford University Press, 2005.
- [22] Greenough, W. T., Black, J. E. "Induction of brain structure by experience: Substrates for cognitive development". In M. Gunnar & C. Nelson (Eds.), *Minnesota Symposia on Child Psychology*. Vol. 24, 1992, *Developmental Behavioral Neuroscience*, p. 155-200.
- [23] Pavlov I.P. "Conditioned reflexes". London, Routledge and Kegan Paul, 1927.
- [24] Tran H, Brunet A, Grenier JM, Datta S.R, Fornace A.J Jr., DiStefano P.S, Chiang L.W, Greenberg M.E. "DNA repair pathway stimulated by the fork-head transcription factor FOXO3a through the Gadd45 protein". *Science*; 296, 2002: 530-534.
- [25] Soleng A, Raastad M, Andersen P. "Conduction latency along CA3 hippocampal axons from rat". *Hippocampus*, 13(8), 2003:953-61
- [26] Shimada Y, Horiguchi M, Nakamura A. "Spatial and temporal properties of interocular timing differences in multifocal visual evoked potentials". *Vision Res.*, Feb, 2005; 45(3):365-71.
- [27] Hammond J., Fischer S., Valova I. "A parallel algorithm for growing, unsupervised, self-organizing maps utilizing specialized regions of influence and neuron inertia". *IASTED International Conference on Circuits, Signals and Systems*, California, October, 2005.
- [28] French D.A., Gruenstein E.I. "An integrate-and-fire model for synchronized bursting in a network of cultured cortical neurons". *Journal of Computational Neuroscience*, Springer Netherlands, Issue Volume 21, Number 3, pp. 227-241, 2006
- [29] Minsky M. "The society of mind", Simon & Schuster paperbacks, 1986.
- [30] Katsunori Kitano, Tomoki Fukai "Variability vs. synchronicity of neuronal activity in local cortical network models with different wiring topologies, *Journal of Computational Neuroscience*, pp. 237-250, 2007.
- [31] Timo Jarvilehto (1999) Role of Efferent Influences on Receptors in the Formation of Knowledge, *Integrative Physiological and Behavioral Science*, April-June, Vol.34, No.2, 90-100.

Toward Logic-Based Cognitively Robust Synthetic Characters in Digital Environments¹

Selmer BRINGSJORD

`selmer@rpi.edu`

Andrew SHILLIDAY, Joshua TAYLOR, Dan WERNER, Micah CLARK

`{shilla, tayloj, werned, clarkm5}@rpi.edu`

Ed CHARPENTIER and Alexander BRINGSJORD

`{charpe, bringa}@rpi.edu`

Rensselaer AI & Reasoning (RAIR) Lab

Department of Cognitive Science

Department of Computer Science

Rensselaer Polytechnic Institute (RPI)

Troy NY 12180

Abstract. With respect to genuine cognitive faculties, present synthetic characters inhabiting online virtual worlds are, to say the least, completely impaired. Current methods aimed at the creation of “immersive” virtual worlds only avatars and NPCs the *illusion* of mentality and, as such, will ultimately fail. Like behaviorism, this doomed approach focuses only on the inputs and outputs of virtual characters and ignores the rich mental structures that are essential for any truly realistic social environment. While this “deceptive” tactic may be suitable so long as a human is in the driver’s seat compensating for the mental deficit, truly convincing autonomous synthetic characters must possess genuine mental states, which can only result from a formal *theory of mind*. We report here on our attempt to invent part of such a theory, one that will enable artificial agents to have and reason about the beliefs of others, resulting in characters that can predict and manipulate the behavior of even *human* players. Furthermore, we present the “embodiment” of our recent successes: Eddie, a four year old child in *Second Life* who can reason about his own beliefs to draw conclusions in a manner that matches human children his age.

Keywords. virtual characters, cognition, logic, theory of mind

¹We are grateful for funding from IARPA and IBM; this funding has partially supported the R&D described herein. We are also indebted to Konstantine Arkoudas for many contributions. Finally, thanks are due to Mike Schoelles for his work on SimBorgs that provided a starting point for automating conversation in *Second Life*.

1. The Problem

Your avatar in the current version of a massively multi-player online (MMO) virtual world (e.g., *Second Life*) is directly tethered to your key strokes, and is nothing more than an automaton controlled directly by what your fingers do. This is great as far as it goes, but the fact remains that your avatar is completely bereft of the cognitive things that make you you, and is in fact devoid of even computational correlates of the things that make you you. He/she doesn't speak or move autonomously, doesn't have any memories, doesn't have any beliefs (none at all, and therefore none of precisely the kind that are distinctive of persons, e.g., beliefs about the beliefs others have about your own beliefs²), doesn't know anything, and therefore certainly can't hold in mind a model of other creatures he/she encounters in such a virtual world.

In laying a foundation for our initial research, we sought to make this problem more concrete: We not only studied the state of the art in AI, but had our laboratory, if you will, enter into it. We thus know that while Rensselaer graduate Joshua Taylor has an avatar in *Second Life*, and also in Nintendo's Wii system (see Figure 1), the fact of the matter is that the real Joshua Taylor has all sorts of beliefs, intentions, goals, and desires that the avatar does not.



Figure 1. Joshua Taylor, Computer Science MS Holder, “Avatared” in Nintendo’s Wii System (left) and *Second Life* (right)

The same fundamental problem arises when we are talking not about avatars, but about NPCs (non-player characters). They are just as cognitively empty, and as a result can invariably be spotted in digital environments as mere shells.

If space permitted, we could show, in detail, that all synthetic characters, to this point in time, whether avatars or NPCs, are indeed primitive. (For some examples, see Figure 2.) Even book-length treatments of how to build synthetic characters are impoverished relative to what we are engineering. For example, Figure 3, taken from [2], models synthetic characters using finite state automata. In such models, there is no knowledge and belief, no reasoning, no declarative memories, and no linguistic capacity.

2. Behaviorism Isn’t the Answer

In the end, given the stone cold emptiness of avatars and NPCs, simple engineering tricks to make these digital creatures seem as if they have genuine mentality isn’t going to work.

²One of the traditional marks of personhood has long been held to be that they can have such so-called *third-order* beliefs. See [1].



Figure 2. Sample Synthetic Characters. Worst to best, in our eyes: Top-left, The Legend of Zelda; SC spits text upon entering room. Top-right, Chrono Trigger; tree-branching conversations. Middle-left, Might & Magic VI (Shopkeepers). Middle-right, Superfly Johnson from Daikatana; behavior scripting, attempts to follow player and act as a sidekick. Bottom-left, Galatea – Interactive Fiction award winner for Best NPC of 2000 (text-based). Bottom-right, Sims 2. Thanks are due to Marc Destefano for these examples and the snapshots.

One can try, through surface-level tricks (such as the “chirping” in *The Sims*, used to make it *seem* as if characters are genuinely conversing), to make the observable behavior of synthetic characters such that these characters *seem* to have knowledge, beliefs, memories, and so on, even when they don’t have these things, but this strategy, ultimately, will always fail. Any approach that treats avatars as empty shells, as puppets, will fail for the same reasons that behaviorism failed: if focuses on inputs (stimuli) and outputs (responses) and ignores everything in between. A rich mental life, however, can only be understood, predicted, and explained (indeed, it can only be *had*) on the assumption that there are rich mental structures mediating inputs and outputs. That was the central tenet of the “cognitive revolution,” and unless we want a form of AI as bankrupt as behaviorism, we must subscribe to it when building synthetic characters. We must thus build synthetic characters with rich mental structures capable of representing (at least computational correlates of) beliefs, knowledge, perceptual input, and so on. And we further argue that we need to do that in a rigorous way, one that specifically capitalizes on the logical properties of mental states, as investigated by logicians, technical philosophers, theoretical cognitive scientists, and so on (rather than by building, say, neural networks and other numerical, non-declarative systems). A very deep kind of *logicist* engineering is thus, at least by our lights, needed. For us, the engineering must be in the mold of logic-based AI [3,4,5,6,7] and computational cognitive modeling [8]. This logic-based work stands in what we see as interesting, stark contrast to brain-based approaches in AI and cognitive science (e.g., to [9]).³

³While we don’t discuss the issue herein, it may be that hybrid approaches bringing together top-down logic-based techniques with bottom-up neuroscience-oriented techniques are worth pursuing. Sun’s [10] is directly relevant here.

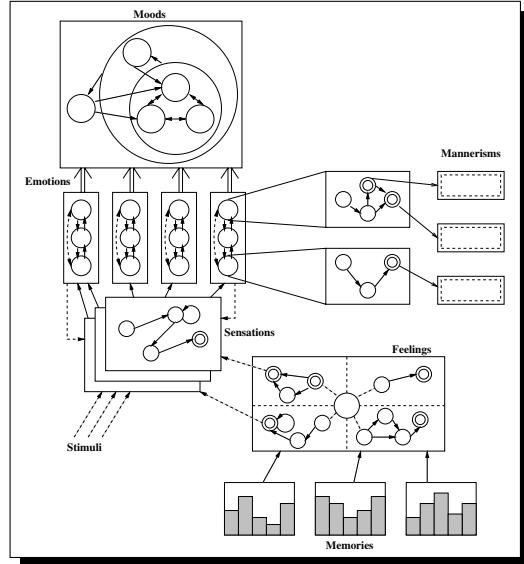


Figure 3. Impoverished FSA-Based Formalism for Synthetic Characters (from Champadard 2003).

3. Our Long-Term Objective

In order to lay a foundation for engineering cognitively robust synthetic characters in massively multiplayer online (MMO) virtual worlds (e.g., *Second Life*) and other digital environments, we are in the process of codifying, to an unprecedented degree, the principles of “common-sense psychology” (CSP) in the form of an implementable logico-mathematical theory. This theory must include rigorous, declarative definitions of all the concepts central to a theory of mind — concepts like lying, betrayal, evil, and so on. The implementation of this theory will be one that artificial agents can deploy in order to understand and predict psychological aspects of the behavior of other artificial agents, and in order to be genuine stand-ins for human beings. No such understanding or prediction is possible today.

Our methodology can be understood along the lines of Lewis’s prescription:

Collect all the platitudes regarding the causal relations of mental states, sensory stimuli, and motor responses. Add also all the platitudes to the effect that one mental state falls under another... Perhaps there are platitudes of other forms as well. Include only the platitudes which are common knowledge amongst us: everyone knows them, everyone knows that everyone else knows them, and so on. [11, p. 256]

Of course, Lewis here is concerned with the construction of a *complete* theory of CSP, one that would be capable of explaining virtually all aspects of human behavior. It is

We also leave aside the economic side of the present work. Surmounting the impoverished nature of synthetic characters, making them, as we say, *cognitively robust*, would have great economic value. Millions of people are willing today to spend hard-earned money to merely control “puppets” in MMO virtual worlds. Billions would, we suspect, spend such money to play in virtual worlds populated by *bona fide* digital psyches. A small but important part of our efforts includes study of the economic and business dimension of MMO virtual worlds and synthetic characters, because we see our R&D as being bound up with the realities of the marketplace. Our study in this area, led by Alexander Bringsjord, is left aside in the present document.

questionable whether the construction of such a theory is a practical possibility, and whether it could be carried out in the absence of a complete physical theory. Even if such a theory is practically possible, it is doubtful whether practical reasoning could be performed on its basis. Therefore, at least initially, we are building a micromodel of CSP.

4. The Theory of Mind Theory

The ability to ascribe mental states to others and to reason about such mental states is indispensable for social communication. All social transactions — from lying and detecting lying, to engaging in commerce and negotiating, to making jokes and empathizing with other people's pain or joy — require at least a rudimentary mastery of CSP. Today's synthetic characters lack such a facility, and hence essentially suffer from autism, which is literally the disease afflicting that part of the brain responsible for reasoning about the mental states of others. An inability to attribute mental states to other characters is “tantamount to not differentiating between the world of objects (with physical states) and the world of persons (with mental states)” [12, p. 65]. A grasp of CSP is particularly important for agents who are trying to manipulate the behavior of other agents. The ability to manipulate through lying, for example, requires sophisticated reasoning about the beliefs of other agents. More generally, social interaction has been described as a form of “social chess” [13], whereby one agent, for instance, “may wish by his own behavior to change the behavior of another; but since the social animal is himself reactive and intelligent, the interaction soon becomes a two-way argument where each ‘player’ must be ready to change his tactics — and maybe his goals — as the game proceeds.” The principles and techniques that neurobiologically normal humans deploy in order to understand, predict, and manipulate the behavior of other humans are collectively referred to as a *theory of mind* [14]. What we are now starting to do is to engineer part of the theory that artificial agents could use to understand, predict, and manipulate the behavior of other agents, and in order to be veridical avatars for human beings, or to be autonomous intellects in their own right.

One quick way to make our project clearer is to briefly turn specifically to lying, a speck in the overall theory we seek, but nonetheless informative to consider.⁴ One well-known definition of lying within analytic philosophy is

$L \text{ lies to } D =_{\text{def}} \exists p \text{ such that}$

1. either L believes that p is not true or L believes that p is false; and
2. L asserts p to D [16, 152].

where

L asserts p to $D =_{\text{def}}$ L states p to D and does so under conditions which, he believes, justify D in believing that he, L , not only accepts p , but also intends to contribute causally to D 's believing that he, L , accepts p [16, 152].

As can be clearly seen by inspection, even if these definitions are only approximately correct, any human D who believes that he or she is being lied to, must have beliefs about the beliefs the liar has about her (i.e., D 's) beliefs, and at the heart of the matter is propositional content.

⁴Clark has extended, formalized, and implemented this definition in what he calls “the lying machine[15].”

5. Near-Term Goal: The Crucial Pair of Demos

In the near term, we are working toward progressively more elaborate versions of two key demonstrations (both of which will be given at AGI2008). The first (the first two versions of which we have already achieved; see below) will subject our system to the litmus test of theories of mind: passing a false-belief test. Experiments with false beliefs were first carried out by Wimmer and Perner[17]. In a typical scenario, a child is presented with a story in which a character *A* places an object (e.g., a marble, teddy bear, cookie, etc.) in a certain location l_1 , say in a particular kitchen cabinet. Then *A* leaves, and during his absence a character *B* removes the object from its original location l_1 and puts it in a different location l_2 (say, a different cabinet). The child is then asked to predict where *A* will look for the object when he gets back — the right answer, of course, being the original location, l_1 . (Very young children ($\approx \leq 4$) don't produce this answer; they do not have a theory of the mind of others.) We are putting artificial agents to a similar test. To pass the test successfully, the agent will be required to make the correct prediction, by deploying our CSP model.

Though we have much work to do, some primitive versions of this first demo have been implemented in connection with *Second Life*. A snapshot from one of our demos is shown in Figure 4. To take a look at a couple of the full videos, visit the link at the bottom of http://www.cogsci.rpi.edu/research/rair/asc_rca.



Figure 4. Snapshot of Demo in SL.

Our demo in *Second Life* is achieved with help from an automated theorem prover, Snark, coupled with procedures for generating and executing AppleScript (a scripting language from Apple Inc.). When Snark discovers a proof that agent *s* will perform action *a*, a procedure written in Common Lisp produces the code that, when executed, simulates keystrokes in *Second Life*, thereby enabling control of an avatar. Figure 5 shows the proof associated with “Ed’s” immature response in the false-belief task carried out in *Second Life*. In this case, *s* is the autonomous synthetic character Eddie, and the action *a* is the selection of the container that the bear has been moved to (so this is an example of failure in the task).

Our second demonstration is in the area of natural language processing; here again an early version is available at the url given above. (See Figure 6 for a snapshot from

```

SNARK-USER 14 >
(in-immature-scenario
(prove '(t-retrieve subject
teddybear
?c)
:answer '(looks-in ?c)))
 $\vdash$ 
(Refutation
(Row 1
(or (not (person ?x)) (not (object ?
y))
(or (not (containsin ?x ?y)) (not (in ?y ?
z))
(bel-in ?x ?y ?z))
assertion)
(Row 2
(or (not (person ?x))
(not (containsin ?x ?y))
(not (object ?y))
(not (w-retrieve ?x ?y))
(not (bel-in ?x ?y ?z))
(t-retrieve ?x ?z ?y))
assertion)
(Row 4
(person subject)
assertion)
(Row 6
(container c2)
assertion)
(Row 7
(object teddybear)
assertion)
 $\vdash$ 
(=Row 8
(in teddybear c2)
assertion)
(Row 9
(w-retrieve subject teddybear)
assertion)
(Row 10
(not (t-retrieve subject teddybear ?
x))
negated-conjecture
Answer (looks-in ?x))
(Row 11
(or (not (person ?x)) (bel-in ?x
teddybear c2))
(rewrite (resolve 1 8) 6 7))
(Row 12
(bel-in subject teddybear c2)
(resolve 11 4))
(Row 28
(t-retrieve subject teddybear c2)
(rewrite (resolve 2 25) 9 7 6 4))
(Row 10
false
(resolve 10 28)
Answer (looks-in c2)))
:PROOF-FOUND
SNARK-USER 15 > (answer t)
(LOOKS-IN C2)

```

Figure 5. Snapshot of False-Belief Task Demo in *Second Life*.

the video.) The goal in the second demonstration is to engineer a virtual character in an MMO virtual world capable of carrying out a conversation based upon certain knowledge and belief. In the demo, this character (BinDistrib) is in conversation with an avatar (CrispyNoodle). When the avatar (CrispyNoodle) speaks, the text in question is read by BinDistrib. For our work here we are leveraging our prior work in the machine reading area [18], an overview of which we now provide.



Figure 6. Snapshot of Conversational Demo in *Second Life*.

5.1. Machine Reading of Text in Second Life

In our approach, text to be read is expressed in logically controlled English, which is translated into multi-sorted logic (MSL). The resulting information in MSL, in conjunction with other background knowledge expressed in MSL, is automatically reasoned over in proof-theoretic and model-based fashion. Our approach to machine reading is a three-phase one:

Phase 1 English texts are rephrased in logically controlled English — i.e., a proper subset of full English that can be unambiguously translated into a formal logic. In the past, we have

made use of Attempto Controlled English (ACE) [19,20], a logically controlled English with a fixed, definite clause grammar and a user-defined vocabulary.⁵ However, we are now moving to CELT [22,23,24].

Phase 2 Discourse representation structures (DRSs) are automatically generated from the controlled English. DRSs are a syntactic variant of first-order logic for the resolution of unbounded anaphora. Their use in the interpretation of text is a central element of discourse representation theory [25,26].

Phase 3 The DRSs are automatically translated into MSL. (We have a number of translators in our lab for going from MSL to straight first-order logic (FOL), using long-established theorems [27].) As a DRS is equivalent to a quantified first-order formula, the translation to FOL is not conceptually difficult. Algorithms for performing such translations are provided by Blackburn [28], among others.

We have used this three-phase process in developing machine reading capability in systems we have built with sponsorship from the DoD. In connection with Slate, one of these systems, the three phases can be viewed from a high-level perspective as shown in Figure 7.

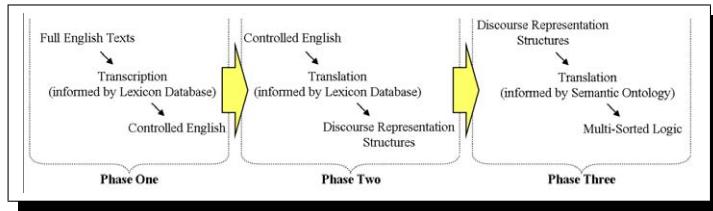


Figure 7. Slate's Reading Process

As we have indicated, in prior logic-based NLP R&D, we have used Attempto Controlled English (ACE), but we are now moving to CELT. CELT is potentially more powerful than ACE because it leverages the Suggested Upper Merged Ontology (SUMO), which incorporates useful knowledge such as geometric axioms. In the current version of our demo, a sentence like

Micah puts the teddy bear in the box.

becomes in CELT's translation to FOL the following formula:

```

(exists
  (?box ?event ?teddy_bear)
  (and
    (instance Micah Human)
    (attribute ?box Box)
    (agent ?event Micah)
    (instance ?event Putting)
    (instance ?teddy_bear Artifact)
    (destination ?event ?box)
    (patient ?event ?teddy_bear)))

```

⁵Phase 1 is currently a manual operation, but techniques developed by Mollá & Schwitter [21] may allow for at least partial automation of this phase.

6. On Relevant Prior Work

It is important to distinguish our approach from other related research efforts:

1. We are not be doing research on BDI (belief-desire-intention) logics in the standard knowledge-representation tradition [29,30]. Most of that work has focused on codifying *normative* reasoning (e.g., about knowledge) in modal logics, with little attention paid to practical concerns such as computational efficiency. We are more interested in a *descriptive* model of reasoning about mental states, rather than a prescriptive model, and one that is computationally tractable.
2. We are not doing cognitive science. While our insights will be obviously coming from human CSP, we will not be particularly concerned with arriving at a cognitively plausible theory. Our aim is not to construct a computational theory which explains and predicts actual human behavior. Rather, our aim is to build artificial agents which are more interesting and useful by enabling them to ascribe mental states to other agents, to reason about such states, and to have, as avatars, states that are correlates to those experienced by corresponding humans. This might result in an interesting computational simulation of a fragment of human CSP, but our primary concern is engineering in the service of entertainment and gaming, not science. In particular, our effort should not be seen as an endorsement of the “theory-theory” position in the current debate between simulation theorists and folk-psychology theorists in cognitive science [31].

Some prior research and development in our own case provides a foundation for the proposed work. For example, with support from NSF, we designed and built synthetic characters (named Ralph and JR9000) that interacted with Bringsjord to teach AI. In addition, the primitive synthetic character known simply as E, was demoed at the 2005 *GameOn!* conference. The associated paper won the “Best Paper” award: [32] In the style of the definition of lying provided above, E is based upon a fully declarative, and fully formal definition of evil.⁶

7. The Main Obstacle: Meaning Mathematized

It’s hard to miss the fact that we are sanguine about our attempt to engineer cognitively robust synthetic characters. In the sober light of AI’s failure to live up to predictions made by its founders, do we really maintain that we can meet with success? Do we not at least see some *seemingly* insuperable barriers out there ahead of us? Actually, we do see a serious hurdle ahead, one that we currently confess we do not have the means to overcome. We do not have the space to carefully describe this hurdle, which is quite technical, but we give here a synopsis of both the challenge, and our approach to meeting it.

The problem is that, to this point, formal logic and formal methods in computer science have provided inadequate mathematical frameworks for pinning down the *meaning* of the propositional attitudes central to our objectives. What does it mean for a person to believe, know, desire, fear, hope, etc. — where this meaning is provided in fully formal fashion that can be rendered in computational form? This question has not been answered. There have been *attempted* answers, but they are clearly inadequate, and logic-based AI is only fooling itself when it takes the current answers to be anywhere near ac-

⁶A copy of the paper can be found at <http://kryten.mm.rpi.edu/GameOnpaper.pdf>.

ceptable. For example, Bringsjord and Konstantine Arkoudas have long been convinced that the appropriation of possible-worlds semantics for supposedly specifying the meaning of *Knows* and *Believes* has been more a marriage of convenience than anything else. For *Necessarily* and *Possibly*, the two operators used in standard modal logic [33], this sort of semantics makes very good sense. For example, it seems eminently sensible to understand $\Box\phi$ (a formula in standard modal logic traditionally used to express that ϕ is logically necessary) to be saying that, in all possible worlds, or in all possible histories, ϕ holds. The standard correlate for knowing, though, is almost laughably forced. What intuitive sense does it make to say (e.g., with [34]) that an agent knows ϕ just in case in all possible worlds epistemically accessible for this agent, ϕ holds?

In the course of the project described herein we are going to need to invent a new logical system for the propositional attitudes at the heart of the robust characters we seek. Some of the first steps in this direction have been taken, and we now say a few words about our system.

Our approach provides a formal semantics for epistemic operators like *Believes* and *Knows* on the basis of *epistemic maps*. Such maps are a particular kind of graph in which the objects of such operators are propositions situated within a web of interconnected inference. Inference is allowed to be deductive, inductive, abductive, analogical, or (a mode that readers are unlikely to be familiar with) creative. Figure 8 shows a very simple epistemic map \mathcal{M} corresponding to a situation in which agent A believes, on the strength of a number of other, subsidiary beliefs (all of which, appearing within rectangles, are perceptually based), that Smith's cat is paralyzed with fear up in an oak tree on Smith's property. In this case, only one inference has taken place, an abductive one. Where ϕ represents in first-order logic that Smith's cat is stymied in an oak tree, the formula $\psi = \text{Believes}_A \phi$ is true on this map (written $\mathcal{M} \models \psi$), but because the strength of ϕ is only at level 1 ("probable") in the continuum from -4 to 4 of strength factors in our approach (a continuum derived directly from [35]), and this level doesn't provide a sufficiently strong justification to move A 's belief into the category of knowledge, $\mathcal{M} \not\models \text{Knows}_A \phi$.

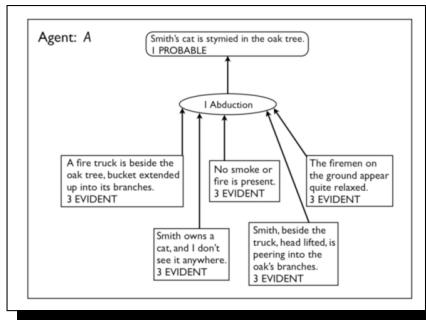


Figure 8. Simple Epistemic Map Re. Smith's Cat

We conclude with a final point about our approach. As is well known, ordinary substitution into extensional contexts leads to error. For example, though agent B may know that John Le Carré is the author of *A Perfect Spy*, it doesn't follow from this and the fact that Le Carré and David Cornwall are one and the same that B knows that David Cornwall wrote this novel. (After all, B may not have heard or seen the string 'David

Cornwall' in his entire life.) Any acceptable epistemic logic must respect such blocked inference. In our approach, as shown in Figure 9, an agent A wouldn't believe that B knows that David Cornwall wrote *A Perfect Spy*, even if A believes that B believes that John Le Carré wrote this book. The mechanical reason for this is that epistemic maps can themselves contain epistemic maps, but, in general, propositions holding in a map \mathcal{M} that contains another map \mathcal{M}' ($calM \supset \mathcal{M}'$) do not get "injected" into \mathcal{M}' . (On the other hand, for all such pairs of epistemic maps, and all formulae δ , if $\mathcal{M}' \models \delta$, then $\mathcal{M} \models \phi$.) Put in terms of Figure 9, the maps shown there don't satisfy the proposition that A believes that B believes that Cornwall wrote the novel in question.

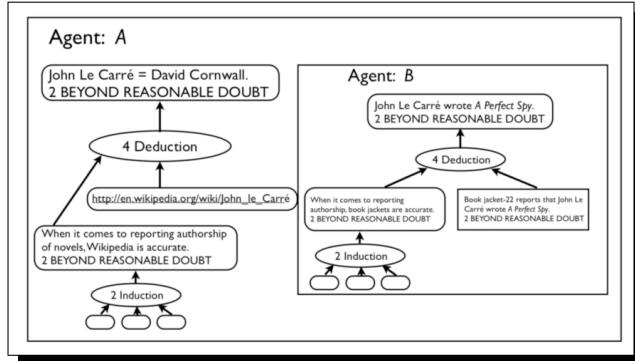


Figure 9. Opaque Context in Our Approach

References

- [1] S. Bringsjord, *Abortion: A Dialogue*. Indianapolis, IN: Hackett, 1997.
- [2] A. Champandard, *AI Game Development*. Berkeley, CA: New Riders, 2003.
- [3] S. Bringsjord, K. Arkoudas, and P. Bello, "Toward a general logicist methodology for engineering ethically correct robots," *IEEE Intelligent Systems*, vol. 21, no. 4, pp. 38–44, 2006.
- [4] S. Bringsjord and D. Ferrucci, *Artificial Intelligence and Literary Creativity: Inside the Mind of Brutus, a Storytelling Machine*. Mahwah, NJ: Lawrence Erlbaum, 2000.
- [5] N. Nilsson, "Logic and Artificial Intelligence," *Artificial Intelligence*, vol. 47, pp. 31–56, 1991.
- [6] S. Bringsjord and D. Ferrucci, "Logic and artificial intelligence: Divorced, still married, separated...?," *Minds and Machines*, vol. 8, pp. 273–308, 1998.
- [7] S. Bringsjord and D. Ferrucci, "Reply to Thayse and Glymour on logic and artificial intelligence," *Minds and Machines*, vol. 8, pp. 313–315, 1998.
- [8] S. Bringsjord, "Declarative/logic-based cognitive modeling," in *The Handbook of Computational Psychology* (R. Sun, ed.), Cambridge, UK: Cambridge University Press, forthcoming.
- [9] J. Hawkins, *On Intelligence*. New York, NY: Holt Paperbacks, 2005.
- [10] R. Sun, *Duality of the Mind*. Mahwah, NJ: Lawrence Erlbaum Associates, 2001.
- [11] D. Lewis, "Psychophysical and Theoretical Identifications," *Australasian Journal of Philosophy*, vol. 50, pp. 249–258, 1972.
- [12] U. Frith, "Cognitive explanations of autism," *Acta Paediatrica Supplement*, vol. 416, pp. 63–68, 1996.
- [13] N. Humphrey, *Consciousness regained*. Oxford University Press, 1984.
- [14] D. Premack and G. Woodruff, "Does the chimpanzee have a theory of mind?," *Behavioral and Brain Sciences*, vol. 4, pp. 515–526, 1978.
- [15] M. Clark, *Cognitive Illusions and the Lying Machine*. PhD thesis, Rensselaer Polytechnic Institute (RPI), 2008.

- [16] R. M. Chisholm and T. D. Feehan, “The Intent to Deceive,” *Journal of Philosophy*, vol. 74, pp. 143–159, March 1977.
- [17] H. Wimmer and J. Perner, “Beliefs about beliefs: Representation and constraining function of wrong beliefs in young children’s understanding of deception,” *Cognition*, vol. 13, pp. 103–128, 1983.
- [18] S. Bringsjord, K. Arkoudas, M. Clark, A. Shilliday, J. Taylor, B. Schimanski, and Y. Yang, “Reporting on some logic-based machine reading research,” in *Proceedings of the 2007 AAAI Spring Symposium: Machine Reading (SS-07-06)*, (Menlo Park, CA), pp. 23–28, AAAI Press, 2007.
- [19] N. E. Fuchs, U. Schwertel, and R. Schwitter, “Attempto Controlled English (ACE) Language Manual, Version 3.0,” Tech. Rep. 99.03, Department of Computer Science, University of Zurich, Zurich, Switzerland, 1999.
- [20] S. Hoefler, “The Syntax of Attempto Controlled English: An Abstract Grammar for ACE 4.0,” Tech. Rep. ifi-2004.03, Department of Informatics, University of Zurich, Zurich, Switzerland, 2004.
- [21] D. Mollá and R. Schwitter, “From Plain English to Controlled English,” in *Proceedings of the 2001 Australasian Natural Language Processing Workshop*, (Macquarie University, Sydney, Australia), pp. 77–83, 2001.
- [22] A. Pease and J. Li, “Controlled english to logic translation,” forthcoming.
- [23] A. Pease and C. Fellbaum, “Language to logic translation with phrasebank,” in *Proceedings of the Second International WordNet Conference (GWC 2004)* (P. Sojka, K. Pala, P. Smrz, C. Fellbaum, and P. Vossen, eds.), (Masaryk University, Brno, Czech Republic), pp. 187–192, 2004.
- [24] A. Pease and C. Fellbaum, “An english to logic translator for ontology-based knowledge representation languages,” in *Proceedings of the 2003 IEEE International Conference on Natural Language Processing and Knowledge Engineering*, (Beijing, China), pp. 777–783, 2003.
- [25] H. Kamp and U. Reyle, *From Discourse to Logic: Introduction to Model-theoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Springer, 1 ed., 1993.
- [26] H. Kamp and U. Reyle, “A Calculus for First Order Discourse Representation Structures,” *Journal of Logic, Language and Information*, vol. 5, pp. 297–348, 1996.
- [27] M. Manzano, *Extensions of First Order Logic*. Cambridge, UK: Cambridge University Press, 1996.
- [28] P. Blackburn and J. Bos, “Working with Discourse Representation Theory: An Advanced Course in Computational Semantics.” Forthcoming.
- [29] M. E. Bratman, *Intention, Plans, and Practical Reason*. CSLI Publications, 1999.
- [30] M. Wooldridge, *Reasoning About Rational Agents*. MIT Press, 2000.
- [31] M. Davies and T. Stone, eds., *Folk Psychology: The Theory of Mind Debate*. Blackwell Publishers, 1995.
- [32] S. Bringsjord, S. Khemlani, K. Arkoudas, C. McEvoy, M. Destefano, and M. Daigle, “Advanced synthetic characters, evil, and E,” in *Game-On 2005, 6th International Conference on Intelligent Games and Simulation* (M. Al-Akaidi and A. E. Rhalibi, eds.), pp. 31–39, Ghent-Zwijnaerde, Belgium: European Simulation Society, 2005.
- [33] B. F. Chellas, *Modal Logic: An Introduction*. Cambridge, UK: Cambridge University Press, 1980.
- [34] R. Fagin, J. Halpern, Y. Moses, and M. Vardi, *Reasoning About Knowledge*. Cambridge, MA: MIT Press, 2004.
- [35] R. Chisholm, *Theory of Knowledge*. Englewood Cliffs, NJ: Prentice-Hall, 1966.

A Cognitive Substrate for Natural Language Understanding

Nicholas L. Cassimatis, Arathi Murugesan and Magdalena D. Bugajska
Human-level Intelligence Laboratory, Rensselaer Polytechnic Institute, Troy

Abstract. Our goal is to understand human language use and create systems that can use human language fluently. We argue that to achieve this goal, we must formulate all of the problems for language use from morphology up to pragmatics using the same *cognitive substrate* of reasoning and representation abilities. We propose such a substrate and described systems based on it. Our arguments, results with real-world systems and ongoing work suggest that the cognitive substrate enables a significant advance in the power of cognitive models and intelligent systems to use human language.

Keywords. Natural language, cognitive substrate.

Introduction

Enabling computers to use and understand natural language would greatly increase our ability to model human language use and the power of the human-computer interface. In this paper we will refer to the goal of creating computers that can use and understand natural language as human-level language use (HLU). Achieving HLU will require serious advances in the state of the art in artificial intelligence and cognitive modeling. One consequence of these advances will be inferential and representational computational abilities that can be applied to other interface modalities.

The following is a (simplified) enumeration of computational tasks involved in HLU:

1. Infer phonemes, morphemes and words being uttered from an acoustic signal.
2. Infer the syntactic structure the words were intended to form.
3. Given knowledge about the world, people and the specific participants of a conversation, infer the goal of the speaker which generated his utterance.
4. Given a goal, produce an utterance that achieves it.
5. Articulate the utterance.

In this paper, we will focus on the goal of inferring syntactic structure and speaker intention, partly because we suspect the insights required to solve these problems will enable fairly straightforward solutions to the other problems.

1. Current AI is not adequate for HLU

Let us now examine the state of the art in artificial intelligence and evaluate whether it is adequate to the task of solving these problems. Many of the individual

observations we will make are obvious to many. They are mentioned here to consider their ramifications for achieving HLU.

1.1. Parsing and intention recognition are closely related

The goal of recognizing the syntax of a sentence and recognizing the speakers' intent are closely related. Consider the sentence:

“Show me the weather for the Red Sox game.”

Does “for the Red Sox game” modify “the weather” or “show”? That is a syntactic question; but to answer it you have to infer that the speaker intends you to show him the weather forecast involving the Red Sox game (“for ...” modifies “the weather”) and does not intend you to show him the weather in order to somehow facilitate the Red Sox game (“for ...” modifies “show”, as in “show me the weather for the second time”).

1.2. Modern algorithms for inference and parsing are incompatible

The algorithms used in AI for parsing are incompatible with those used in inference. For now, we consider “modern” mainstream AI algorithms and later discuss algorithms based on more structured formalisms such as scripts and frames. Probabilistic and statistical approaches are popular in AI today, with Markov chain Monte Carlo (MCMC) methods being perhaps the most popular. However, in parsing research, MCMC is shunned for parsers of probabilistic context-free grammars. This is because MCMC (and other algorithms for propagating beliefs in graphical models) require that all of the relevant state variables in a problem be identified before inference proceeds. However, in parsing we do not have all the state variables in advance. We are simply given a string of words and it is our job to infer which phrases those words were intended to utter. For MCMC to be used to find the most likely parse of a sentence, it must be given the possible parses of a sentence to begin with. For example, to decide whether “the dog” form a noun phrase or whether, as in “the dog food”, they do not, MCMC would need to have both possibilities specified in advance. But that is precisely the job of a parser. Even if you had all (or most) of the possible parses, the size of the corresponding graphical model MCMC would be operating over would probably be prohibitively large. (More on this later). Thus, devotees of the statistical approach in the parsing community must use a different inference algorithm than their kin in the rest of AI.

The upshot of all this is that it is very difficult to guide parsing using perceptual information or world knowledge because the “general” inference algorithms for those are different than the algorithms used by the parsing community.

1.3. Dynamism and scalability

The root cause of the tension between PCFG parsers and MCMC is that the latter are not dynamic¹ (i.e., they do not enable unforeseen objects) to be inferred and do not scale well as the number of state variable increase. These problems are not specific simply to probabilistic inference algorithms but also apply, e.g., to SAT-based search

¹ Dynamic Bayesian Networks are not dynamic in the present sense because they assume that all the state variables are fixed in advance and make inference about how these unfold through time.

algorithms. Recent approaches come closer to addressing these problems but do not address them fully. For example, [1] only deals with the case where new objects are perceived, not when they are actually inferred. [2] does not enable new objects to be introduced at all, but only delays the representation of grounded relations involving those objects until they are needed.

1.4. World knowledge and scalability

The amount of world knowledge and the consequent size of the inference problem are enormous. Consider again our example sentence: “Show me the weather for the Red Sox game.” To retrieve a weather forecast, one requires a time and a location. Neither of these is explicitly specified in the sentence. The Red Sox could be playing away from Boston, the speaker might not even care about the weather for the game itself, but for the reception of his television satellite, say, in St. Louis, as he watches the game. Which game? The next game, the last game, or the game for which the speaker has tickets? Finally, almost anything can be relevant to answering this question. If marketing for a new kind of egg white product is being debuted during a Red Sox game in two weeks and the speaker is a commodity trader, he might be referring to that game. Thus, quite literally, the price of eggs in China could be relevant to even the most prosaic utterance.

This example illustrates that for any given utterance a wide variety of knowledge, involving the speaker’s history and goals, typical intentions of speakers, the behavior of objects and event can all be relevant to using language.

What this means is that the state/search/constraint space for the HLU problem is enormous. To illustrate, let us assume very conservatively, that people know on the order of one million facts (as does CYC), that each fact is expressed with about 4 open variables that can each range over 100 objects and that there are 1 billion seconds in a lifetime. This amounts to 100 billion trillion (10^{23}) concrete constraints. That is well beyond the scope of all general inference algorithms available today (and, we suspect, ever).

The problem is even made worse to the extent that understanding a sentence requires an understanding of multiple people’s mental states. For example, if the listener knows that the speaker thinks a glass of water is a glass of gin, then he will understand who “the man with the gin” refers to. Each person’s beliefs thus become a factor in the understanding of an utterance, which adds at least an integral multiple to the calculations in the last paragraph. It is worse, however. The listener’s beliefs about the speaker’s beliefs about the listener’s beliefs about ... can potentially be relevant. The regress is infinite and, formally at least, so is the state/constraint/search space.

HLU will thus require inference algorithms that are extremely scalable.

Integration

One solution is to abandon general inference algorithms for approaches based on larger structures such as scripts or frames. However, these approaches, *by themselves*, have proven to be too rigid and inflexible. When a situation changes slightly, a structure that worked might, *by itself*, not work.

What is needed is a combination of both approaches: the speed of structured approaches along with the flexibility of more general approaches.

Most researchers who have taken integration seriously have taken a “modular” approach. They build systems that have, for example, a syntax module that outputs

possible parses to other (e.g., semantic and pragmatic) modules that settle on the most likely parse and attach a meaning to it. The problem with this approach is that information from each module is needed to constrain information in the others. The flow of information needs to be deeper and more frequent than the modular approach makes easy to achieve.

So far, we have enumerated many grounds for pessimism. However, we at least have come to a deeper understanding of what the obstacles. We need better inference (more dynamic, scalable, structured and flexible) and a way to tightly integrate syntactic, semantic and pragmatic inference. The next section outlines an approach.

In the preceding discussion, we argued that two key steps to achieving HLU are developing common sense inference algorithms that are scalable, flexible and dynamic and closely integrating inference required for all aspects of language use. In our work, we are addressing these two tasks using the Polyscheme cognitive architecture [3] and the notion of a cognitive substrate [4].

A cognitive substrate for integrating reasoning about language with common sense reasoning

Every step of understanding a sentence can be influenced by a wide variety of factors. Deciding which sense of a word to use in parsing a sentence, for example, can be influenced by the perceived state of the world, statistical regularities in word use, the beliefs and goals of the people involved in the conversations and knowledge about how people and the rest of the world work.

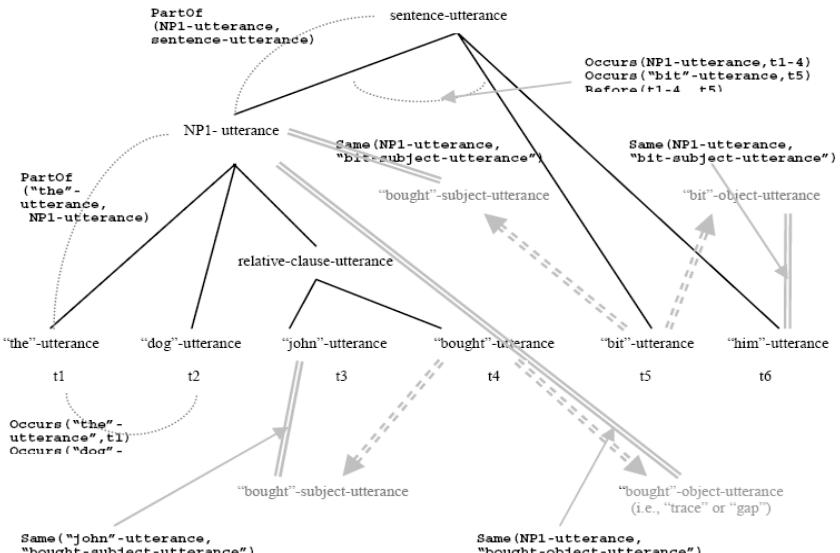


Figure 1: Syntactic structure expressed using substrate representations.

Our goal is to formulate each of these kinds of constraints on language interpretation using the same “language” or “framework”. By so doing, language understanding becomes not several inference problems (e.g., parsing, intention recognition, etc.) with algorithms for solving each encapsulated within separate

modules, but a single inference problem in which every aspect of the problem can constrain every other.

To exploit this fact computationally, we needed to find a way to represent the superficially very different realms of cognition, e.g., syntax and people's beliefs, using the same basic conceptual elements. We have achieved this by showing how to reduce these conceptual fields into the same basic cognitive substrate [4]. This substrate includes reasoning and representation abilities for time, space, causality, identity, parthood and desire. For example, [5] has shown how the problem of syntactic parsing can be reformulated straightforwardly using the same concepts used to formulate constraints on physical actions by agents. Figure 2 shows how to reduce a grammatical rule in HPSG to this substrate. Figure 1 shows an example of a sentence being described in terms of substrate concepts.

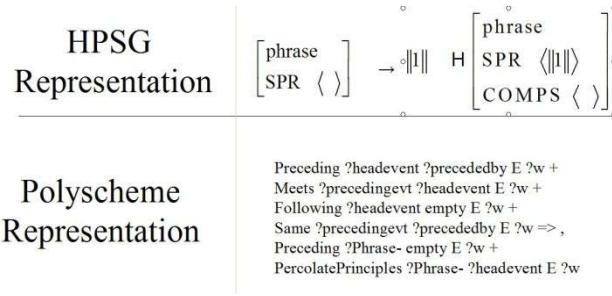


Figure 2: An HPSG rule formulated using substrate concepts.

Table 1 illustrates Cassimatis' [6] mapping between elements of syntactic structure and substrate concepts.

We have also shown [7] how to reduce representations of beliefs to this same basic substrate of concepts. For example, we can represent belief using a combination of identity and counterfactuality. To say John believes Fred is Honest, we say that in the counterfactual world where John is like Fred (*Same(John,Fred,world)*), that Fred is honest (*Honest(Fred,w)*).

These kinds of reductions to substrate representations allow all aspects of language understanding to be represented within the same language, making it easier to let each influence the other.

Table 1: Mapping of syntactic structures onto syntactic structures.

Grammatical Structure	Cognitive structure
Word, phrase, sentence	Event
Constituency	Meronymy
Phrase structure constraints	Constraints among (parts of) events
Word/phrase category	Event category
Word/phrase order	Temporal order
Phrase attachment	Event identity
Coreference/binding	Object identity
Traces	Object permanence
Short- and long-distance dependencies	Apparent motion and long paths.

2. Implementing a substrate in Polyscheme

A substrate that enables all the factors involved in using language to mutually constrain each other must conform to the algorithmic requirements described above. It must integrate the best characteristics of several classes of algorithms by being fast, flexible, scalable and dynamic. In this section we briefly outline how this integration is achieved in the Polyscheme cognitive architecture and direct readers who want more detail to other sources [3, 4, 8].

The fundamental idea behind Polyscheme is that many different kinds of algorithms can be implemented as sequences of attention fixations. These attention fixations in Polyscheme each concentrate multiple computational resources on a single aspect of the world. When algorithms are executed as sequences of attention fixations, integrating these algorithms is as simple as interleaving these sequences.

This enables hybrid algorithms that combine the benefits of multiple algorithms. For example, consider the tension between a general search algorithm and a script-based reasoner. The search algorithm is general and can adapt to many changes in circumstances but is very slow because meaningful state spaces are prohibitively large. The script-based reasoner will work quickly and effectively when the appropriate script exists for a particular situation. However, slight changes in a situation can invalidate a script and we cannot expect each of these changes to have been anticipated in advance. Our goal is to be able to achieve the generality and flexibility of general search without sacrificing the speed of more structured reasoning algorithms.

In Polyscheme, we can achieve this by implementing each algorithm as a sequence of attention fixations. We can look at applying a script as a special case of search in which at each step the operator is chosen from the script. If, when applying a script a problem occurs at a particular step, we can begin to choose operators that a more general search algorithm would choose. The result will be that the elements of a script that are not appropriate for a particular situation will be repaired by the integrated search algorithm. Figure 3 illustrates this process.

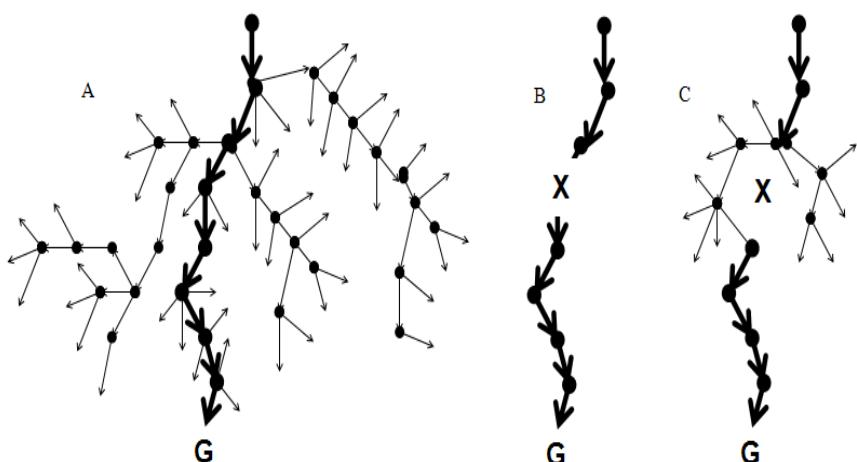


Figure 3: A hybrid (C) of systematic, general, flexible but slow search (A) with fast but rigid case-based reasoning (B).

In ongoing work, we have used this approach to address the tension alluded to above between PCFG parsers and more general probabilistic inference. The fundamental issues that prevent general inference algorithms from parsing PCFGs is that they cannot in the middle of inference add new objects to their reasoning. PCFG parsers constantly do this (e.g., when they infer that a determiner and a noun imply the existence of a noun phrase). Our approach has been to treat PCFG parsing as a maximization problem. We want to find the parse of a sentence with the highest probability of being the parse intended by the speaker. This has led us to implement WalkSAT-inspired algorithm in Polyscheme that maximizes the probability of parses. By integrating this algorithm with a rule matcher we have been able to represent the parsing problem using relational first-order representations instead of propositionally. This also enables objects and relations to be instantiated in memory only when they are needed. Because nonlinguistic relationships can be captured using the rule system, the algorithm automatically takes into account nonlinguistic information when making linguistic inferences. From its point of view, there is no difference between the two types of constraints. Finally, because each step of our algorithm is implemented by a focus of attention that includes information from memory or perceived from the world, this approach enables an embodied approach to sentence processing all the benefits of modern PCFG and probabilistic inference research.

This kind of integration in Polyscheme enables suitable cognitive substrate to be implemented. As alluded to before, the substrate implementation we have so far has enabled a unified and integrated account of physical, epistemic and syntactic inference.

3. Implemented systems

In our preliminary work combining parsing and general inference, we have been able to formulate grammatical rules and world knowledge in the same substrate and develop a system that used the world knowledge to disambiguate the meaning of a sentence. For example, with knowledge about mechanical devices and the relevant syntax and semantics, our parser could find the right sense of “bug” in “the bug needs a new battery”. Confirming the power of the substrate approach, nothing special needed to be done to the grammar rules or the world knowledge to make them interact. It was a zero-effort natural consequence of formulating each within the substrate.



Figure 4: The person cannot see everything the robot can from its perspective.

The last section described how basing a system on a substrate implemented in Polyscheme enables language understanding to be facilitated by rich, embodied inference. This approach was used to enable human-interaction with a mobile robot. As is typically done in Polyscheme, a focus of attention was used to implement inference algorithms for semantic understanding and inference about the world. This focus of attention concentrated several computational resources, including a perceptual input from a video processing system. The result was that visual and spatial information was used to infer a person's perspective and use all that information was used to disambiguate natural language references. For example, in one situation (illustrated in Figure 4), a robot could see two cones (A and B) while it could see that the person it was interacting with only ever saw one cone (B). When the person said "the cone", the robot was able to use perceptual information and spatial inference to infer that the robot was looking at cone B. This required no special encoding of semantic information. The inference was simply a byproduct of Polyscheme's ability to implement a cognitive substrate in which several kinds of constraints (linguistic, epistemic and spatial) could be given a unified formulation.

4. Conclusions

The arguments we have presented, the results with real-world systems we have just described and ongoing work demonstrate that a cognitive substrate and a suitable inference mechanism enable natural language capabilities that will greatly improve the interface between humans and computers.

References

- [1] B. Milch, B. Marthi, D. Sontag, S. Russell, D. L. Ong, and A. Kolobov, "BLOG: Probabilistic Models with Unknown Objects," presented at IJCAI-05, Edinburgh, Scotland, 2005.
- [2] P. Domingos and M. Richardson, "Markov Logic Networks," *Machine Learning*, vol. 62, pp. 107-136, 2006.
- [3] N. L. Cassimatis, "Integrating Cognitive Models Based on Different Computational Methods," presented at Twenty-Seventh Annual Conference of the Cognitive Science Society, 2005.
- [4] N. L. Cassimatis, "A Cognitive Substrate for Human-Level Intelligence," *Artificial Intelligence Magazine*, vol. 27, 2006.
- [5] A. Murugesan and N. L. Cassimatis, "A Model of Syntactic Parsing Based on Domain-General Cognitive Mechanisms," presented at 8th Annual Conference of the Cognitive Science Society, Vancouver, Canada, 2006.
- [6] N. L. Cassimatis, "Grammatical Processing Using the Mechanisms of Physical Inferences," presented at Twentieth-Sixth Annual Conference of the Cognitive Science Society, 2004.
- [7] P. Bello and N. L. Cassimatis, "Developmental Accounts of Theory-of-Mind Acquisition: Achieving Clarity via Computational Cognitive Modeling," presented at 28th Annual Conference of the Cognitive Science Society, Vancouver, Canada, 2006.
- [8] N. L. Cassimatis, J. G. Trafton, M. Bugajska, and A. C. Schultz, "Integrating Cognition, Perception, and Action through Mental Simulation in Robots," *Robotics and Autonomous Systems*, vol. 49, pp. 13-23, 2004.

The China-Brain Project

Building China's Artificial Brain Using an Evolved Neural Net Module Approach

Hugo de GARIS^{1,5}, TANG Jian Yu^{1,2}, HUANG Zhiyong³, BAI Lu¹, CHEN Cong¹,
CHEN Shuo¹, GUO Junfei¹, TAN Xianjin¹, TIAN Hao¹, TIAN Xiaohan¹,
WU Xianjian¹, XIONG Ye¹, YU Xiangqian¹, HUANG Di⁴

¹ *Brain Builder Group, International School of Software, Wuhan University,
Wuhan, Hubei, China. Email: profhugodegaris@yahoo.com*

² *Computer Science School, Hubei University of Economics, Wuhan, Hubei, China*

³ *Computer Science School, Wuhan University, Wuhan, Hubei, China*

⁴ *Computer Science School, University of Geosciences, Wuhan, Hubei, China*

⁵ *(from early 2008) Brain Builder Group, Institute of Artificial Intelligence,
Department of Computer Science, Xiamen University, Xiamen, Fujian, China*

Abstract. Prof. Hugo de Garis has recently received a 3 million RMB, 4 year grant to build China's first artificial brain, starting in 2008, that will consist of approximately 15,000 interconnected neural net modules, evolved one at a time in a special accelerator board [1] (which is 50 times faster than using an ordinary PC) to control the hundreds of behaviors of an autonomous robot. The approach taken in building this artificial brain is fast and cheap (e.g. \$1500 for the FPGA board, \$1000 for the robot, and \$500 for the PC, a total of \$3000), so we hope that other brain building groups around the world will copy this evolutionary engineering approach.

Keywords. Artificial Brains, Evolved Neural Net Modules, Brain Architects, Accelerator Board

1. Introduction

This paper shows how the first author and his research team build *artificial brains* [2], and how we plan to build the first artificial brain in China, now that we have been funded for it. We define an *artificial brain* (A-Brain) to be a “network of neural network modules” (10,000–50,000 of them), each of which is evolved quickly in special electronic programmable (FPGA based) hardware, then downloaded into the memory of a PC, and interconnected according to the designs of human “BAs” (“Brain Architects”). The neural signaling of the artificial brain is performed by the PC in real time (defined to be 25Hz per neuron). Such artificial brains can be used for many purposes, e.g. controlling the behaviors of autonomous robots.

However, there is a major problem with the above approach (if one does not use accelerator boards), namely the slowness, using PCs, of evolving individual neural network modules. Typically, it can take many hours, or even a day to evolve a single neural net module on a PC. Obviously, evolving several tens of thousands of such modules using only a PC to build an artificial brain will not be practical. Before such A-Brains can be built with this approach, it is necessary to find a way to accelerate the evolution of such a large number of neural net (NN) modules. This we have done. It is now possible for us to execute the evolution of neural net modules in hardware, and achieve a speedup factor (relative to ordinary PC evolution speeds) of about 50 times. We use a Celoxica company's FPGA (Field Programmable Gate Array) electronic accelerator board (containing a 3-megagate FPGA, i.e. Xilinx's Virtex II programmable chip) to accelerate the evolution of neural network modules. In this paper we describe how we build artificial brains (using the accelerator board, which is an essential component in our method). The remaining contents of this paper are as follows. Section 2 provides an overview of how we evolve our neural network modules. Section 3 describes the "Celoxica" electronic board that we use to accelerate the neural net module evolution. Section 4 explains the so-called "IMSI" (Inter Module Signaling Interface), i.e. the software used to allow modules to send and receive signals between themselves. Section 5 describes the properties of the robot that our A-Brain is to control. Section 6 talks about the brain-robot interface. Section 7 gives an example of a multi-module architecture to illustrate how we build our A-Brains. Section 8 presents some examples of single module evolution. Section 9 concludes.

2. Evolving Neural Net Modules

This section gives a brief description of the approach that we use to evolve our neural network (NN) modules that become components for building artificial brains. We use a particular neural net model called "GenNet" [3]. A GenNet neural network consists of N (typically N = 12-20) fully connected artificial neurons. Each of the N^2 connections has a "weight", represented as a signed, binary fraction, real number, with p (typically p = 6-10) bits per weight. The bit string chromosome used to evolve the N^2 concatenated weights will have a length of $N^2(p+1)$ bits. Each neuron "j" receives input signals from the N neurons (i.e. including a signal from itself). Each input signal S_{ij} (a binary fraction) is multiplied by the corresponding connection weight W_{ij} and summed. To this sum is added an external signal value E_j . This final sum is called the "activation signal" A_j to the neuron "j".

$$A_j = \sum_{i=1}^N W_{ij} S_{ij} + E_j$$

This activation value is fed into a sigmoid function g that acts as a "squashing" function, limiting the output value S_j to have a maximum absolute value of 1.0

$$S_j = g(A_j) = \frac{A_j}{|A_j| + 1.0}$$

Each neuron of a neural network module has a weighted connection to (usually) a single output neuron, whose output signal is considered to be the output signal for the whole module. This output signal $S(t)$ is compared to a target (desired) signal value

$T(t)$ for some number (e.g. 100) “ticks” (where a “tick” is defined to be the time taken for all neurons in the module to signal once). The “fitness function” used in the genetic algorithm (GA) to perform the evolution of the neural net module is usually defined as follows.

$$f = \frac{1}{\sum_{t=1}^{100} (T(t) - S(t))^2}$$

3. The Celoxica Board

The aims of lowering the price of high-speed evolution, and of achieving higher performance in evolving hardware led us to use FPGAs (Field Programmable Gate Arrays). FPGAs are specially made digital semiconductor circuits that are often used for prototyping. The several million logic gates in modern FPGAs (e.g. Xilinx’s Virtex II chip) make it possible to have multiple copies of the same electronic sub circuit running simultaneously on different areas of the FPGA. This parallelism is very useful for a genetic algorithm. It allows the program to process the most time costly weight calculations in parallel, and this can speed up the overall evolution by a factor of about 50 times, as we have determined experimentally in earlier work. We chose the Celoxica FPGA board for our project. “Celoxica” is the name of a UK company [1]. Our Celoxica board (an RC203) costs about \$1500. With such a board, a design engineer is able to *program* electrical connections on site for a specific application, without paying thousands of dollars to have the chip manufactured in mass quantities. We currently use an RC203 FPGA board for our experiments, which is a desktop platform for the evaluation and development of high performance applications. The main FPGA chip is a Xilinx Virtex II that can be configured without using an HDL (Hardware Description Language). Instead, it uses a much easier high-level “C-like” language called “Handel-C” (after Handel the composer) [4]. This language is very similar to ordinary C (i.e. with approximately an 80% overlap between the two languages), with a few extra features, particularly those involved with specifying which functions ought to be executed in *parallel*.

A Celoxica board attaches to a PC, with two-way communication, so that instructions can be sent to the board from the PC, and results coming from the board can be displayed on the PC’s screen.

One of the aims of this paper is to emphasize that using the Celoxica board makes brain-building practical, due to the considerable speedup factor in the evolution of the individual neural network modules used to make artificial brains. The value of this speedup factor is *critical* to this whole approach, which aims to make brain building *cheap*. If a substantial speedup factor can be achieved, it then becomes practical to evolve large numbers of neural net modules in a reasonable time, and to interconnect them to build artificial brains inside a PC.



Fig. 1 The “Celoxica” Board (\$1500)

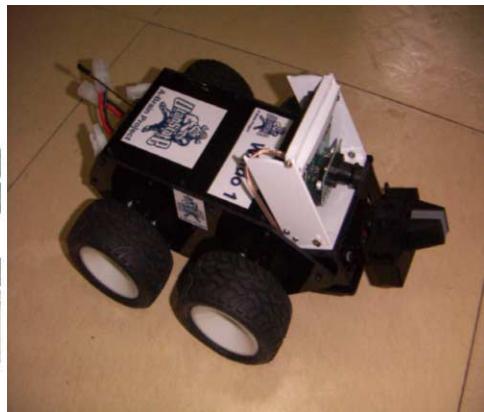


Fig. 2 Our Robot (\$1000)

4. Inter Module Signaling Interface (IMSI)

In order that each neural net module can calculate the strength of its output signal, it needs to know the strengths of all its input signals, including not only from “intra-module” connections, but also from “inter-module” connections, i.e. either from the “external world” (e.g. from sensors), or from the outputs of other modules. Each module therefore needs a look up table (LUT) which lists the sources of its external input signals (from sensors), and the integer I.D.s of the modules from which it receives their output signals. A similar lookup table is needed to specify to which other modules each module *sends* its output signal(s) to. One of the jobs of the BAs (Brain Architects) is then to specify the interconnections between the modules, and to enter them into these LUTs.

Special software was written for the PC, called “IMSI” (“Inter Module Signaling Interface”) which is used to calculate the neural signaling of each neuron in each module of the artificial brain. An ordinary PC is used to run the IMSI program, which calculates *sequentially* the output signal value(s) of each module, for all modules in the artificial brain. The IMSI code calculates the output neural signal value for each module in the artificial brain. It loops through each module sequentially, using its input LUT to find the signal values of its external inputs, as well as its internal signal values. A slightly more complex “activation function” A is then calculated according to the following formula

$$A_i = \sum_{(j=1, N)} W_{ij} * S_j + \sum_{(i=1, P)} E_i$$

where a W_{ij} is the weight value of the connection between neuron “i” and “j” in the module, S_j is the value of the signal strength on that connection, E_i is the value of an external neural signal, N is the number of neurons in the module, P is the number of external input signals for that module. The convention used above is that a signal travels from the “from” neuron “j” to the “to” neuron “i”. Each module has a table of its weight values W_{ij} (i.e. N^2 of them) and a list of its external input signals (i.e. P of them, where P usually varies from module to module). The IMSI software calculates the S_i

signal value for each module and stores it in its output signal register (“OSR”). This is done sequentially for all modules in the artificial brain. The IMSI then places the output signal value of each module into the input signal registers (“ISR’s) of those modules that the outputting module signals to. For example, let us assume that module M1432 sends its output signal value (0.328) to two other modules M3729 and M3606, then the IMSI would use the “Output Modules LUT” of M1432 to place the value 0.328 into one of the input signal registers in each of the modules M3729 and M3606.

Thus the output signals of a given clock cycle (i.e. the time taken for all modules to calculate their output signal values) become the input values at the next clock cycle. Each module also has a table of its internal neural signal values S_j , where j ranges from 1 to N , the number of neurons in the module. These N values are calculated in each clock cycle, to be used in the following clock cycle. In the IMSI code, two such “S tables” are used, in a “ping-pong-ing” style, e.g. table S_t (calculated in clock cycle “ t ”) is used to calculate table S_{t+1} (in clock cycle $t+1$), which in turn is used to calculate table S_{t+2} , which actually overwrites table S_t (hence the term “ping-pong-ing”).

Within each clock cycle “ t ”, the IMSI calculates for each module “ m ”, its output signal value S_m and updates its S table. The value S_m is transferred to “ m ’s” connecting modules. Thus all the data is in place for calculating the S_m values in the next clock cycle. Performing all these calculations is the job of the IMSI software. Some outputs of modules are not sent to other modules but to external “effectors”, e.g. to the motors of a robot, or to a transducer that generates a radio signal, etc. The IMSI deals with these special external outputs (i.e. not to other modules).

Actually, the IMSI has two roles. Certainly its main role is as described above, i.e. performing the neural signaling in the PC. Another important role is to allow the BAs (Brain Architects) to specify the connections between the modules. Thus for example, when BAs want to add a new module to the artificial brain, they will need to use IMSI to specify the module’s :-

- a) external inputs (from sensors)
- b) inputs from other modules (i.e. their integer I.D.s)
- c) outputs to other modules (i.e. their integer I.D.s)
- d) external outputs (to effectors)

As the size of the artificial brain grows, special book-keeping software is used to describe each module, e.g. its function, its fitness definition, its size (i.e. number of neurons), its evolutionary strategy, its connections with other modules, its position in the whole artificial brain design, etc.

5. The Robot

Fig. 2 shows a photo of our robot. It is controlled by a 2 way antenna at the back of the robot, having 4 wheels, a gripper at the front, and a “CMU-CAM2” programmable CCD camera. Visual (and other) data from the robot is transmitted via radio signals to the A-Brain in the PC, and vice versa. This robot cost less than \$1000 to build, so is not expensive. Its length is about 20 cms. This robot will be controlled by an artificial brain consisting of at least 15,000 evolved neural net modules that will allow the A-Brain to have many hundreds of pattern recognition circuits, and a similar number of decision circuits.

With any publicly funded research project, one needs to show results. An artificial brain “hidden” inside a PC is not visible. Even a large wall chart, consisting of

thousands of interconnected neural net modules, may impress people with its complexity, but would not have the visual impact of a moving robot that displays hundreds of behaviors, switching from one to another depending on circumstances both external and internal. *If the average person can remain intrigued for half an hour by such a robot and the A-Brain that controls it, then we state that the A-Brain has passed the “China Test”.*

6. The Brain-Robot Interface

Our research team has a robotics-mechatronics-electronic-engineering expert, who is working on the interface problem between the robot and the artificial brain in the PC. The CMU-CAM2 camera on the robot is programmable, using an assembler-like language which can be used to process the mega-pixel image that the camera generates. This language contains about 100 different statements. The “robot guy” of our team has the task of becoming an expert in this “camera language”, and then to provide the rest of the team, who are mostly BAs (Brain Architects) and EEs (Evolutionary Engineers) who design and evolve the many neural net modules for the A-brain, with a summary of what the camera outputs, as well as ideas on how these outputs can be translated (interfaced) with possible inputs to neural net modules. This mapping from *mega*-pixel camera images to *single digit* inputs to modules is what we call the robot-brain interface. This mapping, which we have yet to specify in detail, will be executed using normal computer code, in the PC (as will the IMSI), so our A-Brain is actually a “hybrid” system, consisting of evolved neural network modules, and standard high-level computer code. Having a clearer idea of the robot-brain interface will give the research team a better idea of what “first layer” neural net modules to suggest, that take as their inputs, the outputs from the interface mapping.

A type of reverse mapping, from the A-brain’s output modules to the motors of the wheels, the camera motor, and the gripper motor, is also needed. Numerical output signal values from neural net modules need to be translated into a form that is suitable for the various motors. Signals between the robot and PC are sent via 2-way radio antenna, so another level of interfacing is needed, i.e. between the antenna and to the motors, and between the effectors (wheels, camera, and gripper) and the antenna that sends signals to the PC. Once our robotist has finished his work, it will be easier for the BAs defining the modules to design them to interface appropriately with the hardware aspects of the project. Of course, the major part of the effort involved in the project remains the specification and evolution of the 10,000s of modules used in the A-Brain.

7. Multi-Module Architecture

This section describes a simple artificial brain architecture of several dozen modules, to illustrate the types of issues that arise when trying to design an artificial brain using our approach. For this simple brain (for the illustrative purposes of this paper) we do not use a real robot, but rather a conceptual “toy” robot, which can be modeled as a horizontal rectangular body, with four wheels, and a “V” shaped antenna, as shown in Fig. 3. When designing an artificial brain, certain common sense factors come into play. For example, one needs to answer certain basic questions, such as –

- What is the AB (Artificial Brain) to control, i.e. what is the “vehicle” that the AB controls? (In our case the vehicle is the conceptual robot of Fig. 3)
- What environment does the vehicle operate in?
- What are the behaviors of the vehicle that the AB needs to control?
- What are the inputs and outputs of the vehicle?

Once these basic questions are answered, the more detailed A-Brain design work can begin. In our simple *didactic* case, the answers we give to the above questions are the following. The environment in this simple conceptual model is a flat surface that contains vertical triangles and squares, that project onto the camera eye (that looks like a cone in Fig. 3). Sounds are also emitted from the environment of two different frequencies (high and low). The behaviors of the robot are simple. It can turn left slowly or quickly. It can turn right slowly or quickly. It can move straight ahead slowly or quickly, and it can stop. So there are 7 initial behaviors, to be controlled by the A-Brain. The inputs to the robot are a vertical triangle or square on the “retinal grid” of the eye (i.e. 2 possibilities), and a high or low frequency sound.

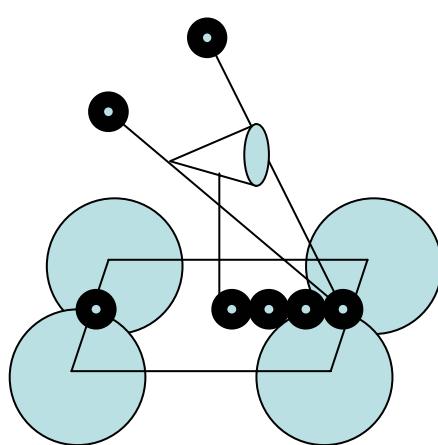


Fig. 3 The Conceptual Robot “Vehicle”

Frequency	Image	Position	Action
high	triangle	L > R	LF
high	triangle	L < R	RF
high	triangle	L = R	AF
high	square	L > R	RF
high	square	L < R	LF
high	square	L = R	LF
low	triangle	L > R	LS
low	triangle	L < R	RS
low	triangle	L = R	AS
low	square	L > R	RS
low	square	L < R	LS
low	square	L = R	LS

Fig. 4. Rules for Action Selection

The next set of questions asks how input stimuli map to output behaviors for the vehicle. In our case we place the robot in the context of a “story”, i.e. a description of the behaviors of the robot in a “coherent” context. This is rather vague, so it will be illustrated with a concrete example to clarify the concept. The “story” we provide is that the robot uses its detectors to see what the visual image is (i.e. is it a triangle or a square) and whether the sound has a high or low frequency.

To the robot brain, these input signals are interpreted as follows. If the sound has a low frequency, then that means that the source of the sound (assumed to be some object in the environment) is far away, so the robot need not react quickly, i.e. its motions can be slow. If the frequency of the sound is high, then the robot interprets this to mean the object creating the sound is close, so it has to react quickly, so its motions are fast, not

slow. Implicit in these interpretations is that a near object (high frequency sound) could be a “threat” (e.g. a predator), so the robot needs to react quickly.

If the image is a triangle, then the robot interprets it as being something “positive”, e.g. as prey, as food, as something to be approached. If the image is a square, then the robot interprets it as being something “negative”, e.g. as a predator, as dangerous, as something to be avoided. So the “story” in this case is “If the object detected in the retinal grid of the eye is dangerous, then flee.” “If it is prey, then approach it.” This mapping of sensor input to behavioral output makes sense in the context of the “story”, i.e. “eat but don’t be eaten”, which also makes biological sense. The robot has two other detectors (black circles in Fig. 3) on the tips of its “V” antenna. These are “Signal Strength Detectors” (SSDs). If the sound strength drops off as an inverse function of the distance between the robot and the source, then by having two such “SSDs” one can use them to locate the position of the source of the sound. For example, if the sound source lies to the “front-left” of the robot (i.e. as the robot faces the object, the object lies closer to the left SSD on the left branch of the antenna than the SSD on the right branch, then the signal strength detected will be stronger than that detected by the SSD on the tip of the right branch of the antenna. One can use this “signal strength difference” in the decision making as to which behavior the robot “chooses” to perform (i.e. the 7 of them). To give a concrete example, take the case of the sound having a high frequency, that the image is a triangle, and that the left SSD has a stronger signal than the right SSD (abbreviated to “L > R”). Then this combination of inputs maps to a given output in the form of a “rule”

IF (freq = high) & (image = triangle) & (L > R) THEN turn left fast (abbreviated to LF)

In light of the above story, this rule makes sense. The frequency is high, so the sound source is close. Therefore the reaction of the robot should be fast, hence the F in the behavioral selection (on the right hand side of the above rule). Since the image is a triangle, that means the object is a prey, to be approached. Since L > R, the object lies to the front-left of the robot, so to approach it, the robot should turn left, and quickly, i.e. the behavior selected is LF.

Once one understands this simple rule, then the rest of the rules listed in Fig. 4 are easily interpreted, e.g. in the *Action* column, an L means “left”, an R means “right”, an A means “approach”, an F means “fast”, an S means “slow”. For example, AS means “approach slowly”, LF means (turn) “left fast”. Most of the actions in Fig. 4 are turning behaviors. Once the robot has turned enough, so that the signal strength difference between the two SSDs on the V antenna is effectively zero, the robot then moves straight ahead, i.e. it approaches the source of the sound, or flees from it. The stop behavior creates complications, so we will ignore it in this simple didactic model. So effectively there are 6 behaviors. Now that we understand the “story”, i.e. the general behavior of the robot, we can now turn to a more detailed discussion of neural net modules that can implement these behaviors. These modules are now listed. We need –

- 2 detectors for the image, i.e. a “triangle detector”, and a “square detector”
- 1 SSD (signal strength detector) (2 copies, for the two branches of the V antenna)
- 2 frequency detectors (one for the high frequency, one for the low frequency)
- 3 difference detectors (i.e. L > R, L < R, L = R).
- 2 logic gates (“and”, “or”)

That's only 10 different modules, but it is enough for the didactic purposes of this section. These modules can be combined (i.e. interconnected) to form networks of

modules (i.e. a form of networks of (neural) networks), called “circuits” (or “sub-systems”), where the output(s) of some modules become the inputs of other modules, e.g. for the above rule, the following circuit could be used, as shown in Fig. 5.

Similar circuits can be made for the other 11 cases of Fig. 4. But we need not make 12 separate circuits similar to that in Fig. 5. That would be wasted effort. Instead we can use “or” gates to aggregate several cases. For example, Fig. 4 shows that there are 3 cases that give an LF output (or action). So make those 3 LF outputs become the inputs to a 3 input “OR” gate – and similarly for LS (3 cases), RF (2 cases), RS (2 cases). AF and AS only have single cases, so they don't need an “OR” gate. One need not replicate SSDs, nor difference detectors ($L > R$), ($L < R$), ($L = R$). One can simply have the outputs of these detectors branch to become inputs to multiple other circuits. Putting all these 12 (non redundant) circuits together would generate quite a complex circuit, which is not given in this section.

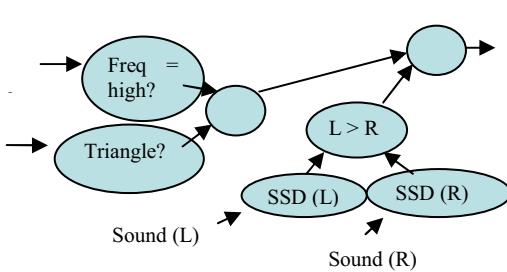


Fig. 5 Circuit for LF Rule

	<i>Left Wheel</i>	<i>Right Wheel</i>
LS	0.2	0.4
LF	0.2	0.8
RS	0.4	0.2
RF	0.8	0.2
AS	0.4	0.4
AF	0.8	0.8

Fig. 6 Control Signals to Wheels for Robot Turning

Similar circuits can be made for the other 11 cases of Fig. 4. But we need not make 12 separate circuits similar to that in Fig. 5. That would be wasted effort. Instead we can use “or” gates to aggregate several cases. For example, Fig. 4 shows that there are 3 cases that give an LF output (or action). So make those 3 LF outputs become the inputs to a 3 input “OR” gate – and similarly for LS (3 cases), RF (2 cases), RS (2 cases). AF and AS only have single cases, so they don't need an “OR” gate. One need not replicate SSDs, nor difference detectors ($L > R$), ($L < R$), ($L = R$). One can simply have the outputs of these detectors branch to become inputs to multiple other circuits. Putting all these 12 (non redundant) circuits together would generate quite a complex circuit, which is not given in this section.

How can we implement the LF, LS, AF, AS, RF, and RS? Common sense says that to get a 4 wheeled vehicle to turn left, one needs to make the right side wheels turn faster than the left side wheels. So use 3 more modules that output a constant signal of a low value (e.g. 0.2) and a constant signal of a middling value (e.g. 0.4), and a constant signal of a high value (e.g. 0.8). These different output signal values can be used to turn the wheels at different speeds. A high signal (0.8) will turn the wheel it controls quickly. A middling signal (0.4) will turn its wheel at a middling speed, etc. Fig. 6 gives the combination of control signals to make the robot turn appropriately. This brings the total of different modules now to 13. If the behavior LF is activated,

then it can send two output signals which become the inputs to modules that generate the 2 control signals for the wheels (i.e. 0.2 and 0.8). This simple circuit is shown in Fig. 7.

There are 6 different behaviors (LF, LS, AF, AS, RS, RS), with different combinations of control signal strengths to the wheels, so how to solve the problem that only one set of signals should be sent to the wheels at a time? With 6 circuits functioning simultaneously, it is possible to have several of the six having non negligible output values at the same time. This creates a conceptual problem. We want to have only one of these 6 to be active (e.g. a strong output value of 0.8) and the rest to be inactive (i.e. with weak output values of 0.2). Before proceeding further, it is interesting to note that the problem we are now discussing is fairly typical of the role of a brain builder or a “BA” (Brain Architect). It is analogous to the case of an electronic engineer who designs digital electronic circuits. There is a lot of creativity involved, and there may be “many ways to skin a cat” (i.e. many alternatives to solving a problem, some of them being more intelligent, or more efficient than others). Returning to the problem – how can only one of the 6 signals be strong and the other 5 weak? This sounds like a “winner takes all (WTA)” problem. So we suggest creating a circuit that has 6 inputs, and 6 outputs. If the i th input signal is the strongest of the 6 input signals, then the i th output signal is strong (e.g. 0.8) and all the other output signals are weak (e.g. 0.2), as shown in Fig. 8. How to design such a circuit using evolvable neural net modules? (One sees the scope for creativity here!)

The very idea of using a WTA circuit may not be the best way to solve the “unique signal problem” (i.e. ensuring that only one set of signals is sent to the wheel motors). An alternative might be to add the output signals of the 6 behaviors (i.e. LF + LS + AS + AF + RF + RS). This will generate some inaccuracies for a time, but so long as the behaviors don't change very fast from one to another, there will probably be time for the contributions of the 5 non active behaviors to drop to low values, leaving only the active behavior. (On the other hand a weak signal is about 0.2, so the sum of 5 weak signals is roughly equal to one strong one, so we still have a problem). So let us pursue the WTA approach. How to implement such a circuit? Trying to evolve a 6-input, 6-output circuit is probably too difficult, so our instinct is to use a “divide and conquer” approach, i.e. evolve simpler modules and then connect them to create the WTA circuit. Consider a “dominator” circuit, which has 3 inputs A, B, C, and one output O as shown in Fig. 9. If the input signal A (on the left) is the largest of the 3 input signals, then the output O takes the input value A, else, O takes a very low value (e.g. 0.1 or 0.05). With two such circuits we can combine them with a “greater than” circuit to create a WTA circuit (component) provided that the left most of the 6 inputs is the largest, as shown in Fig. 10. It should now not be too difficult to see that with 6 such components, we can construct a winner take all (WTA) circuit. For example, if the input B has the largest input signal strength, then if we swap the inputs A and B, then using the circuit of Fig. 10, we should get a strong output (e.g. 0.8) indicating that B was the strongest input. Similarly, if we swap A and C, then if C is the strongest, again the output O will be strong (else weak), and so on for the remaining 3 cases. With 6 such circuits, we can construct a full WTA.

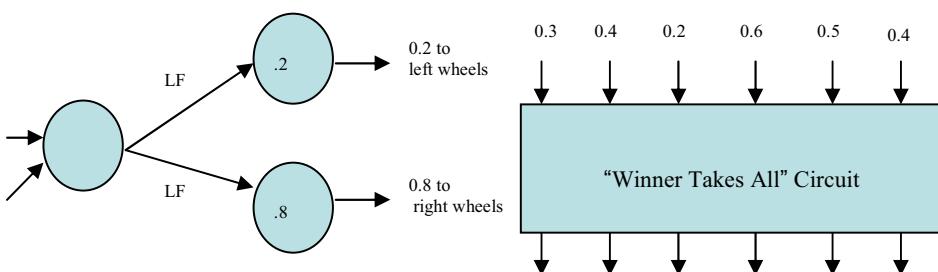


Fig. 7 Control Signals to Wheels for LF

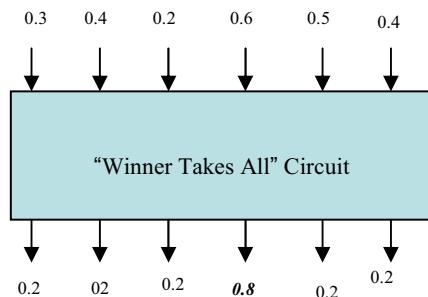


Fig. 8 Winner Takes All (WTA) Circuit

The 6 outputs from the WTA circuit (one of which will be strong (0.8), and the other five very weak (0.05) are fed into 6 circuits of the form shown in Fig. 7. The 6 “left wheel” output signals and the 6 “right wheel” output signals can be “paired”, i.e. 2 of the left wheel outputs have a value of 0.1, 2 have a value of 0.4, and 2 have a value of 0.8. Similarly with the right wheel outputs. But instead of having 6 circuits as in Fig. 7 we can simplify things a bit as in Fig. 11. Using the data of Fig. 4, an LS output needs to send a 0.2 signal to the left motor and a 0.4 signal to the right motor. An LF output sends signals 0.2 and 0.8 respectively to the left and right motors. Hence LS and LF both send a 0.2 signal to the left motor. By connecting the two left motor signals to an OR gate as shown in Fig. 11 if one of LS or LF is active from the WTA circuit, the OR gate will output a strong signal. In the case of the LS/LF “OR” gate, if one of the LS or LF is active, then the output of the OR gate is strong. This output connects to a new module called “Multigen”, which generates a multiple of outputs depending on which input has a strong signal. If the left input is strong (0.8) the output is a constant signal of 0.2 - if the middle input is strong, the output is a constant signal of 0.4 - if the right input is strong, the output is a constant of 0.8. Outputs go to the motors.

8. Single Module Evolution

This didactic section describes how a selection (a subset) of the modules discussed in the previous section can be evolved. The first author is currently writing a book entitled “Artificial Brains : An Evolved Neural Net Approach”, to be published by World Scientific (Singapore) early 2010. In this book will be the descriptions of the actual successful evolutions of many modules (probably many dozens) to give readers and potential brain builders a fairly thorough introduction as to how to evolve individual neural net modules, that is so critical to this approach of building artificial brains. In this section we give only two, because describing each one takes quite a bit of time. We start with a rather simple example, i.e. the Multigen module of Fig. 11. Remember its specification. It has 3 inputs and one output. If the left most input is high, and the others low, it outputs a constant signal of 0.2. If the middle input is high, it outputs a constant signal of 0.4. If the right most input is high, it outputs a constant signal of 0.8. How can one evolve such a module? We use a “multi-test” approach, which is a technique commonly used in our work. In this case there are 3 tests, or experiments, i.e. the 3 cases of input combinations (i.e. left high, middle high, right

high). Let us run each test (i.e. arrange to have the appropriate input signal values) for a given number of “ticks”, e.g. 33 ticks for each experiment. So the total number of ticks for the “multi-test” experiment will be 100 ticks. Fig. 12 shows the target output signal values for the “multi-test evolution”. To evolve this module, three experiments or tests are carried out on the same module, hence the word “multi-test”. For the first 33 ticks, 3 inputs (left to right) of value (0.8, 0.2, 0.2) are presented. For the second 33 ticks, the 3 inputs are (0.2, 0.8, 0.2), and for the third 33 ticks, the 3 inputs are (0.2, 0.2, 0.8). At the junctions of 33, 66 ticks, the internal neuronal signals are reset to zero, so that a “fresh” experiment can begin. The fitness definition is

$$f = \frac{1}{\sum_{t=1}^{100} (T(t) - S(t))^2}$$

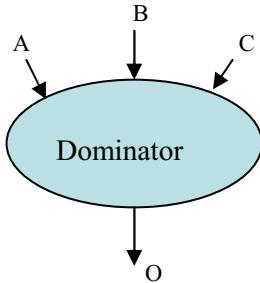


Fig. 9 Dominator Circuit

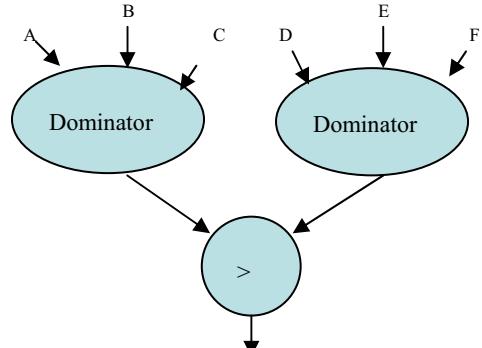


Fig. 10 Winner Takes All (WTA) Circuit Component

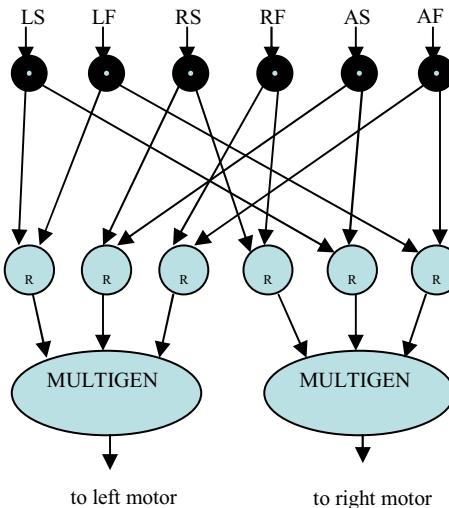


Fig. 11 Motor Control Circuit

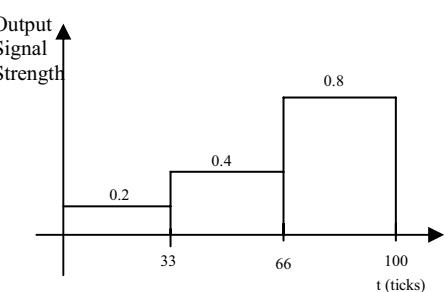


Fig. 12 Target Output Values for the Multigen Module

The $T(t)$ is the target (desired) output signal value. It takes the value 0.2 for the first 33 ticks, then 0.4 for the second 33 ticks, and 0.8 for the third 33 ticks. $S(t)$ are the actual output signal values from the evolving module. The closer the $S(t)$ values are to the target values $T(t)$, the higher the fitness score f . The fitness is a simple inverse of the sum of the squares of the differences between the T and S values for each tick t . For a second example, take the case of the frequency detector in Fig. 5. We create a training set of “positive” and “negative” examples. Let us assume that the environmental model in which the robot is situated contains only 2 frequencies, i.e. high and low, e.g. a high frequency with period of 20 ticks, and a low frequency with period 40 ticks. Assume the strengths of the sound signals received by the robot take the following form.

$$S(t) = A_i \sin(2\pi t/T)$$

Let the amplitude of the signal, i.e. A_i have 3 different values, 0.2, 0.5, 0.8. Hence there are 6 cases, i.e. 3 different possibilities for the amplitude, and 2 cases for the period T . There will be thus 3 positive and 3 negative examples, corresponding to the whether the period is small (high frequency) or large (low frequency). Use a 6 case multi-test evolution, in which the 6 cases are presented each for 80 ticks. For the positive cases, the target output is high i.e. 0.8, and for the negative cases the target output is low i.e. 0.2.

The above modules are fairly simple in their function and could be readily programmed conventionally in a high level computer language, so why bother evolving them? This next example, a pattern detector, can be evolved rather quickly, but it is not at all obvious how one might program it, and especially in a comparable time. We anticipate that there will be several thousand pattern detector modules in our 15000 module A-Brain, so we feel justified in our evolutionary engineering approach, when people challenge us with comments such as “Why don't you just program everything?” The CMU-CAM2 camera generates a mega-pixel image that can be data compressed to a much smaller number of pixels, each with its own grey level or color code. For didactic purposes, imagine this compressed grid has dimensions 10*10. Each pixel of the grid has an output signal with magnitude less than 1.0. Each such pixel value is sent to every neuron in a pattern detector module, so if the module has 16 neurons, there would be 16*10*10 (external) connections, and hence weights, to the module i.e. 1600. Imagine the module is to output a strong signal if the image contains a square (a positive example), and a weak signal if it contains a triangle (a negative example), i.e. it is a “square detector”.

A training set of images containing positive and negative examples is prepared, e.g. squares and triangles of different sizes and positions in the image. Each image is presented for 50 ticks, and the internal signals between the neurons in the module are cleared between each test, i.e. between the presentation of each image. The fitness function for this “multi-test evolution” is the inverse of the following formula

$$(T - P) \sum_t \sum_{(+ve \text{ tests})} (0.8 - S(t))^2 + P \sum_t \sum_{(-ve \text{ tests})} (0.1 - S(t))^2$$

where T is the total number of images, P is the number of positive (square) images, so $T-P$ is the number of negative (triangle) images. $S(t)$ is the strength at each tick of the output signal of the module. The double sums are over the 50 ticks (t), and the positive and negative test cases. The factors $(T - P)$ and P before the summation signs are used

to “balance” the evolution, i.e. to steer the evolution equally between the negative and the positive cases. Such pattern detector modules can often be evolved quickly using the Celoxica board, and in much less time than they could be conventionally programmed (assuming that one can even *think* of a way to program such cases quickly, which is often *not* the case.)

For more examples of single neural net module evolution, readers will have to wait until the first author’s book “Artificial Brains : An Evolved Neural Net Module Approach” is published early 2010.

9. Conclusions

The Celoxica FPGA board is capable of *speeding up the evolution of neural network modules (relative to that on a PC) by a factor of about 50 times*, depending on the size of the neural net being evolved. We believe this to be an important advance and an essential step when artificial brains, comprised of 10,000s of such modules are to be evolved *cheaply* over a reasonable time period, and then run in real time in interconnected form in an ordinary PC. Prior to the use of the Celoxica board, the evolution of a single neural network could take many hours on an ordinary PC, a fact that made brain building according to our PC-based approach quite impractical.

The underlying technology, i.e. the electronic evolution of large numbers of neural net modules, will make the production of 10,000s of evolved modules needed to build an artificial brain, practical. Then the *real* challenge of designing an artificial brain can begin, and will result hopefully in the creation a new research field, namely “Brain Building” or “Artificial Brains”.

This paper defined an artificial brain to be a network of evolved neural network modules. Each module is evolved quickly using a (Celoxica) electronic accelerator board (i.e. about 50 times faster than using a PC). The evolved weights of each neural net module are downloaded into the RAM memory of the PC. This is done thousands of times, one by one, up to a maximum of several 10,000s (which is the limit with which the PC can perform the neural signaling of all the (interconnected) modules sequentially in *real time*, where real time means the PC calculates the output signal value of every neuron in the whole A-Brain 25 times per second (25 Hz).

Once all the modules are downloaded into the PC’s memory, they are interconnected according to the designs of human “BAs” (Brain Architects). The interconnected network of neural networks is the A-Brain (Artificial Brain). When the PC is calculating each neuron’s output signal, it uses the IMSI’s LUTs to know where its inputs come from, and to where it is to send its output signals. Today’s PCs allow the *real time* neural signaling of A-Brains with up to a maximum of several 10,000s of modules. To observe a 15,000 module A-Brain controlling the hundreds of behaviors of a robot should give the impression to human observers that the robot “has a brain behind it”. Longer term, the research field of “Artificial Brains” should grow in size equivalent to those of NASA, ESA, etc.

References

- [1] Celoxica 2006, www.celoxica.com
- [2] Hugo de Garis, Michael Korkin, "The Cam-Brain Machine (CBM) : An FPGA Based Hardware Tool which Evolves a 1000 Neuron Net Circuit Module in Seconds and Updates a 75 Million Neuron Artificial Brain for Real Time Robot Control", *Neurocomputing*, Vol. 42, Issue 1-4, February, 2002, Elsevier. Special issue on Evolutionary Neural Systems, guest editor Prof. Hugo de Garis, <http://www.iss.whu.edu.cn/degaris/papers>
- [3] PPT notes for a masters level Artificial Brains course
<http://www.iss.whu.edu.cn/degaris/coursestaught.htm> Click on the CS7910 course.
- [4] The "Handel-C" High Level Language for Programming the Celoxica board, 2006, www.celoxica.com
- [5] Hugo de Garis, "Guest Editorial", *Neurocomputing*, Vol. 42, Issue 1-4, February, 2002, Elsevier. Special issue on Evolutionary Neural Systems, guest editor Prof. Hugo de Garis.
- [6] Hugo de Garis, Michael Korkin, "The CAM-Brain Machine (CBM) : Real Time Evolution and Update of a 75 Million Neuron FPGA-Based Artificial Brain", *Journal of VLSI Signal Processing Systems*, Special Issue on VLSI on Custom Computing Technology, Vol. 24, Issue 2-3, (2000), pp 241-262.
- [7] Hugo de Garis, Michael Korkin, Gary Fehr, "The CAM-Brain Machine (CBM) : An FPGA Based Tool for Evolving a 75 Million Neuron Artificial Brain to Control a Lifesized Kitten Robot", *Journal of Autonomous Robots*, Vol. 10, No. 3, May 2001.
- [8] Hugo de Garis, Michael Korkin, Felix Gers, Eiji Nawa, Michael Hough, "Building an Artificial Brain Using an FPGA Based CAM-Brain Machine", *Applied Mathematics and Computation Journal*, Special Issue on "Artificial Life and Robotics, Artificial Brain, Brain Computing and Brainware", Vol. 111, 2000, pp163-192, North Holland.
- [9] Hugo de Garis, Felix Gers, Michael Korkin, Arvin Agah, Norberto 'Eiji' Nawa, "CAM-Brain" : AT&T's Billion Neuron Artificial Brain Project : A Three Year Progress Report", *Artificial Life and Robotics Journal*, Vol. 2, 1998, pp 56-61.
- [10] Harik G.R., Lobo F.G., Goldberg D.E., "The Compact Genetic Algorithm", *IEEE Transactions on Evolutionary Computation*, Volume 3, Issue 4, Nov 1999, pp 287 – 297.

Cognitive Architectures: Where do we go from here?

Włodzisław DUCH^{a,1}, Richard J. OENTARYO^b, and Michel PASQUIER^b

^a*Dept. of Informatics, Nicolaus Copernicus University, Toruń, Poland*

^b*School of Computer Engineering, Nanyang Technological University, Singapore*

Abstract. Cognitive architectures play a vital role in providing blueprints for building future intelligent systems supporting a broad range of capabilities similar to those of humans. How useful are existing architectures for creating artificial general intelligence? A critical survey of the state of the art in cognitive architectures is presented providing a useful insight into the possible frameworks for general intelligence. Grand challenges and an outline of the most promising future directions are described.

Keywords: cognitive architectures, artificial general intelligence, neurocognitive models, intelligent agents.

1. Introduction

A long-term goal for artificial general intelligence (AGI) is to create systems that will exceed human level of competence in a large number of areas. There is a steady progress toward this goal in several domains, including recognition of specific patterns, memorization and retrieval of vast amount of information, interpreting signals and other types of numerical information, autonomous control, board games and reasoning in restricted domains. Yet even in lower level cognitive functions, such as object recognition or scene analysis artificial systems are still far behind the natural ones. Higher-level cognitive functions, such as language, reasoning, problem solving or planning, involve complex knowledge structures and are much more difficult to realize. Various types of memory stored by the brain facilitate recognition, association, semantic interpretation and rapid retrieval of large amounts of complex information patterns. At quite basic level organization of storage and retrieval of information in computers is completely different than in brains. Computing architectures are universal only in principle, in practice they always constrain information processing in specific ways. Computers are better in many tasks than brains, and vice versa, brains are better in many important tasks than computers. It is not clear at all whether cognitive architectures (CA) running on conventional computers, will reach the flexibility of the brain in lower or higher-level cognitive functions.

Traditionally higher cognitive functions, such as thinking, reasoning, planning, problem solving or linguistic competencies have been the focus of artificial intelligence (AI), relying on symbolic problem solving to build complex knowledge structure. These functions involve sequential search processes [1], while lower cognitive functions, such as perception, motor control, sensorimotor actions, associative memory recall or categorization, are accomplished on a faster time scale in a parallel way, without stepwise deliberation. Embodiment is a powerful trend in robotics and there is now a general agreement that the meaning of many concepts should be grounded in embodied, sensorimotor representations. While symbolic approximations that account for sensorimotor processes are certainly possible not much is known about their limitations, for example how deep the grounding of symbols should be, and how to achieve it through embodied cognition. Perhaps the dream of creating a General Problem Solver [2] may be realized with relatively minor extensions to symbolic cognitive architectures, while detailed understanding of animal behavior and creating flexible mobile robotic applications may require a distributed approach to embodied cognition. Analysis of existing cognitive architectures should facilitate understanding of limitations of different approaches. Many general ideas seem to explain everything but do not scale up well to real applications, therefore a clear notion what exactly AGI should do is necessary.

¹ Corresponding author, Dept. of Informatics, Nicolaus Copernicus Uni., Grudziądzka 5, Toruń, Poland, Google: "W. Duch".

2. Grand challenges for AGI

What should be required from an AI system to be worthy of the “Artificial General Intelligence” name? Artificial Intelligence has focused on many specific approaches to problem solving, useful for development of expert systems, neglecting its initial ambitious goals. One requirement for AGI, storing and manipulation of vast amount of knowledge, has been addressed by the Cyc project [3]. Started in 1984 a huge frame-based knowledge base has been constructed, but the list of its “potential applications” has not been replaced by real applications for decades. Perhaps the biggest mismatch between AI reality and popular expectations is in the language-related domains, for example in general purpose conversational systems, developed mostly in the form of various chatterbots by commercial companies and enthusiastic individuals. Restricted form of the Turing test [4] (the full test being too difficult to try), called Loebner Prize competition², has been won for almost two decades by chatterbots based on old template matching techniques, or more recently contextual pattern matching techniques. Such programs have no chance to develop real understanding of language and use it in meaningful dialogs or texts analysis, but may be used for stereotyped question/answer systems or “impersonation”. Carpenter and Freeman have proposed a “personal Turing test” [5], where a person tries to guess if the conversation is done with a program or a real personally known individual. Human behavior includes the ability to impersonate other people, and the personal Turing test may be an interesting landmark step on the road to general intelligence.

Another area that poses remarkable challenge to AI is word games, and in particular the 20-questions game. Word games require extensive knowledge about objects and their properties, but not about complex relations between objects. Different methods of knowledge representation may be used in different applications, from quite simple, facilitating efficient use of knowledge, to quite involved, needed only in deep reasoning. In fact simple vector-space techniques for knowledge representation are sufficient to play the 20-question game [6]. Success in learning language depends on automatic creation and maintenance of large-scale knowledge bases, bootstrapping on the resources from the Internet. Question/answer systems pose even more demanding challenge, and in this area a series of competitions organized at Text Retrieval Conference (TREC) series may be used to measure progress. Intelligent tutoring systems are the next great challenge, but there seem to be no clearly defined milestones in this field.

Feigenbaum [7] proposed as a grand challenge building a super-expert system in a narrow domain. This seems to go in a direction of specialized, rather than general intelligence, but one may argue that a super-expert without general intelligence needed for communication with humans is not feasible. Sophisticated reasoning by human experts and artificial systems in such fields as mathematics, bioscience or law may be compared by a panel of experts who will pose problems, rise questions, and ask for further explanations to probe the understanding of the subject. A good example of such challenge is provided by the Automated Theorem Proving (ATM) System Competitions (CASC) in many sub-categories. An interesting step toward general AI in mathematics would be to create general theorem provers, perhaps using meta-learning techniques that rely on specialized modules. Automatic curation of genomic/pathways databases and creation of models of genetic and metabolic processes for various organisms poses great challenges for super-experts, as the amount of information in such databases exceeds by far human capacity to handle it.

Defining similar challenges and milestones towards AGI in other fields is certainly worthwhile. The ultimate goal would be to develop programs that will advise human experts in their work, evaluating their reasoning, perhaps even adding some creative ideas. DARPA in the “Personal Assistants that Learn” (PAL) program sponsors a large-scale effort in similar direction. Nilsson [8] has argued for development of general purpose educable systems that can be taught skills needed to perform human jobs, and to measure which fraction of these jobs can be done by AI systems. Building one such system replaces the need for building many specialized systems, as already Allan Turing [4] has noted proposing a “child machine” in his classical paper. Some human jobs are knowledge-based and can be done by information processing systems, where progress may be measured by passing a series of examinations, as is done in such fields as accounting. However, most human jobs involve manual labor, requiring senso-motoric coordination that should be mastered by household robots or autonomous vehicles. The DARPA Urban Challenge competition (2007) required integration of computer vision, signal processing, control and some reasoning. It is still simpler than control of a humanoid robot, where direct interaction of robots with people will require an understanding of perception, controlling of attention, learning causal models from observations, and hierarchical learning with different temporal scales. Creation of partners or personal assistants, rather than complete replacements for

² See <http://www.loebner.net> for information on the Loebner competition.

human workers, may be treated as a partial success. Unfortunately specific milestones for this type of applications have yet to be precisely defined. Some ordering of different jobs from the point of view of difficulty to learn them could be worthwhile. In fact many jobs have already been completely automatized, reducing the number of people in manufacturing, financial services, printing houses etc. In most cases alternative organization of work is to be credited for reduction in the number of jobs (plant and factory automation, ATM machines, vending machines), not because of deployment of AI systems.

A detailed roadmap to AGI should thus be based on detailed analysis of the challenges, relationships between various functions that should be implemented to address them, system requirements to achieve these functions and classes of problems that should be solved at a given stage.

3. Cognitive architectures

Cognitive architectures are frequently created to model human performance in multimodal multiple task situations [1][9] rather than to create AGI. A short critical review of selected cognitive architectures that can contribute to development of AGI is provided below. Allen Newell in his 1990 book *Unified Theories of Cognition* [1] provided 12 criteria for evaluation of cognitive systems: adaptive behavior, dynamic behavior, flexible behavior, development, evolution, learning, knowledge integration, vast knowledge base, natural language, real-time performance, and brain realization. These criteria have been analyzed and applied to ACT-R, Soar and classical connectionist architectures [10] but such fine-grained categorization makes comparison of different systems rather difficult. Without going into such details we shall propose below a simpler taxonomy, give some examples of different types of cognitive systems that are currently under development, and provide a critique and some recommendations for better systems. Surveys on the system organization and working mechanisms of a few cognitive architectures that have already been published [11] were not written from the AGI point of view.

Two key design properties that underlie the development of any cognitive architecture are *memory* and *learning*. The importance of memory has been stressed from different perspectives in a few recent books [12]-[14]. Various types of memory serve as a repository for background knowledge about the world and oneself, about the current episode of activity, while learning is the main process that shapes this knowledge. Together learning and memory form the rudimentary aspects of cognition on which higher-order functions and intelligent capabilities, such as deliberative reasoning, planning, and self-regulation, are built. Organization of memory depends on the knowledge representation schemes. A simple taxonomy of cognitive architectures based on these two main features leads to a division of different approaches into three main groups (Fig. 1): *symbolic*, *emergent*, and *hybrid* models.

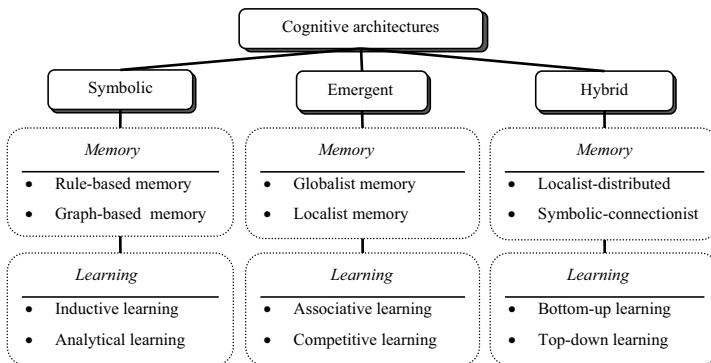


Fig. 1. Simplified taxonomy of cognitive architectures

Roughly speaking symbolic architectures focus on information processing using high-level symbols or declarative knowledge, in a classical AI top-down, analytic approach. Emergent architectures use low-level activation signals flowing through a network consisting of numerous processing units, a bottom-up process

relying on the emergent self-organizing and associative properties. Hybrid architectures result from combining the symbolic and emergent paradigms in one way or another. The memory and learning aspects of these three broad classes of approaches are investigated in details below.

3.1. Symbolic architectures

There is a strong link between the type of architecture and problems it is supposed to solve. The use of symbols as the key means to support information processing originates from the physical symbol system hypothesis [1], which has been motivated by research on memory and problem solving. A physical symbol system has the ability to input, output, store and alter symbolic entities, and to carry out appropriate actions in order to reach its goals. The majority of symbolic architectures utilize a centralized control over the information flow from sensory inputs through memory to motor outputs. This approach stresses the working memory executive functions, with an access to semantic memory for knowledge retrieval. *Rule-based* representations of perception-action memory in knowledge-based production systems embody the logical reasoning skills of human experts. *Graph-based* representations are typically encoded as directed graph structures comprising nodes for symbolic entities and their attributes, and edges for relationships among them. Main examples of this sort of knowledge representation are semantic networks and conceptual graphs (Sowa 1984), frames/schemata [16], and reactive action packages (RAPs) [17]. The underlying paradigms of these approaches remain very similar, and are sometimes even equivalent.

Substantial efforts have been made over the years to introduce *analytical and inductive* learning techniques to symbolic systems. The former aims at exploiting existing general/specific facts to infer other facts that they entail logically. Prominent examples include explanation-based learning (EBL) [18] and analogical learning [19]. Inductive machine learning, on the other hand, seeks to derive from specific facts or examples general rules which capture the underlying domain structure. Well-known examples of this kind include knowledge-based inductive learning (KBIL) [20] and delayed reinforcement learning [21]. Many ambitious cognitive architectures have been proposed and abandoned after a period of vigorous activity. Only those potential candidates for AGI that are still actively developed are reviewed below.

SOAR (*State, Operator And Result*) is a classic example of expert rule-based cognitive architecture designed to model general intelligence [1][22]. Based on theoretical framework of knowledge-based systems seen as an approximation to physical symbol systems, SOAR stores its knowledge in form of production rules, arranged in terms of operators that act in the problem space, that is the set of states that represent the task at hand. The primary learning mechanism in SOAR is termed chunking, a type of analytical EBL technique for formulating rules and macro-operations from problem solving traces [22]. In recent years many extensions of the SOAR architecture have been proposed: reinforcement learning to adjust the preference values for operators, episodic learning to retain history of system evolution, semantic learning to describe more abstract, declarative knowledge, visual imagery, emotions, moods and feelings used to speed up reinforcement learning and direct reasoning³. SOAR architecture has demonstrated a variety of high-level cognitive functions, processing large and complex rule sets in planning, problem solving and natural language comprehension (NL-SOAR) in real-time distributed environments⁴. At present SOAR architecture has not yet integrated all these extensions. A few additional ones, like memory decay/forgetting, attention and information selection, learning hierarchical representations, or handling uncertainty and imprecision, will also be useful. The design of the perceptual-motor systems within SOAR is fairly unrealistic, requiring users to define their own input and output functions for a given domain. It remains to be seen how well numerous problems that face such an extension can be handled using the existing architecture as a base.

EPIC (*Executive Process Interactive Control*) is a cognitive architecture for building computational models that subsume many aspects of human performance [23]. It aims at capturing human perceptual, cognitive and motor activities through several interconnected processors working in parallel, and to build models of human-computer interaction for practical purposes. The system is controlled by production rules for cognitive processor and a set of perceptual (visual, auditory, tactile) and motor processors operating on symbolically coded features rather than raw sensory data. Although EPIC is focused on multiple simple tasks in one experiment it has been connected to SOAR for problem solving, planning and learning, and the EPIC-SOAR combination has been applied to air traffic control simulation [23].

³ J. Laird, Current Soar Research, see <http://ai.eecs.umich.edu/people/laird/current-research.html>

⁴ For more references see the Soar community site: <http://sitemaker.umich.edu/soar/home>

ICARUS project [24] defines an integrated cognitive architecture for physical agents, with knowledge specified in the form of reactive skills, each denoting goal-relevant reactions to a class of problems. The architecture includes a number of modules: a perceptual system, a planning system, an execution system, and several memory systems. Concepts are matched to percepts in a bottom-up way and goals are matched to skills in a top-down way. Conceptual memory contains knowledge about general classes of objects and their relationships, while skill memory stores knowledge about the ways of doing things. Each comprises a long-term memory (LTM) and a short-term memory (STM). The LTM is organized in a hierarchical fashion, with the conceptual memory directing bottom-up, percept-driven inference and skill memory controlling top-down, goal-driven selection of actions. The acquisition of knowledge in ICARUS is achieved through hierarchical, incremental reinforcement learning, propagating reward values backward through time. Since skills comprise sub-skills, the system learns to calculate the value estimates over a stack of state-action pairs, instead of just a single pair as in the traditional reinforcement learning. This hierarchical processing strategy yields in turn a much faster learning rate than that of the standard reinforcement learning. The hierarchical memory organization and learning procedure have equipped ICARUS with the ability to focus its attention on a specific object or event in its sensor range and to reduce the reaction time and the search space requirements. Through its planning and memory modules, ICARUS is also able to incrementally learn new concepts in an efficient manner by constructing a feature tree that the system can comprehend. Relational reinforcement learning that gives priority to high-utility beliefs and rapidly finds most useful inferences handles well large memory hierarchies [25]. Interesting applications of this architecture to in-city driving, blocks world or free-cell solitaire have been demonstrated. One vital problem is the lack of concurrent processing to cope with asynchronous inputs from multiple sensors while coordinating resources and actions across different modalities. Issues related to uncertainty have also been largely ignored.

NARS (Non-Axiomatic Reasoning System) [26] project has been developed for over two decades. It is a reasoning system based on a language for knowledge representation, an experience-grounded semantics of the language, a set of inference rules, a memory structure, and a control mechanism, carrying out various high-level cognitive tasks as different aspects of the same underlying process. The non-axiomatic logic is used for adaptation with insufficient knowledge and resources, operating on patterns that have the “truth-value” evaluated according to the system’s “experience” with using these patterns. This approach allows for emergence of experience-grounded semantics, and inferences defined on judgments. While several working NARS prototypes of increasing complexity have been built they are solving only rather simple problems.

SNePS (Semantic Network Processing System) [27] is a logic, frame and network-based knowledge representation, reasoning, and acting system that went through over three decades of development. It stores knowledge and beliefs of some agent in form of assertions (propositions) about various entities. Each knowledge representation has its own inference scheme, logic formula, frame slots and network path, integrated in SNIP, the SNePS Inference Package. When a belief revision system discovers contradiction some hypotheses that led to the contradiction have to be unasserted by the user and the system removes all those assertions that depend on it. The SNePS Rational Engine controls plans and sequences of actions using frames and believing/disbelieving propositions. The natural language processing system works with English morphological analyzer/synthesizer and generalized augmented transition network grammar interpreter. SNePS has been used for commonsense reasoning, natural language understanding and generation, contextual vocabulary acquisition, control of simulated cognitive agent that is able to chat with the users, question/answer system and other applications. Interesting inferences have been demonstrated, but the program has not yet been used in a large-scale real application, for example a chatterbot.

3.2. Emergent Paradigm Architectures

Emergent cognitive architectures are inspired by connectionist ideas [28]. Their relation to the processes that take place in the brain may be rather distant. Processing elements (PEs) form network nodes that interact with each other in a specific way changing their internal states and revealing interesting emergent properties. There are two complementary approaches to memory organization, *globalist* and *localist*. The Multi-Layer Perceptron (MLP) and other neural networks based on delocalized transfer functions process information in a distributed, global way. All parameters of such networks influence their outputs. Generalization of learned responses to novel stimuli is usually good, but learning new items may lead to catastrophic interference with old knowledge [29]. The basis set expansion networks that use localized functions (such as Gaussians) are examples of localist networks; the output signals for a given input depend only on a small subset of units that

are activated. However, it should be remembered that a modular organization of globalist network will easily create a subgroups of processing elements that react in a local way [30].

The learning methodologies for emergent architectures are quite diverse [28][29]. *Associative learning* creates a mapping of specific input representation to specific output representation and in this sense remembers the reactions, heteroassociations or enables pattern completion (autoassociations). In this case, learning may be either guided directly by a set of correct “target” signals or indirectly by certain critic inputs, which correspond to the supervised and reinforcement learning paradigms in AI, respectively. In *competitive learning* all PEs compete to become active and learn in an unsupervised fashion. The simplest form of this procedure is the winner-takes-all (WTA) rule, which permits only one winning PE (or one per group) to learn at a time, while inhibiting others that lose the competition. Correlation-based learning using Hebb learning rule captures statistical properties of incoming signals creating an internal model of the environment [29]. A few emergent architectures that are still under active development are presented below. In applications to complex reasoning they have not yet reached the same level of maturity as symbolic architectures, but are much closer to natural perception and reasoning based on perceptions, rather than symbolic knowledge.

IBCA (*Integrated Biologically-based Cognitive Architecture*) is a large-scale emergent architecture that epitomizes the automatic and distributed notions of information processing in the brain [29]. The role of three regions in the brain is emphasized: posterior cortex (PC), frontal cortex (FC), and hippocampus (HC). The PC module assumes an overlapping, distributed localist organization that focuses on sensory-motor as well as multi-modal, hierarchical processing. The FC module employs a non-overlapping, recurrent localist organization in which working memory units are isolated from one another and contribute combinatorially (with separate active units representing different features). The HC module utilizes a sparse, conjunctive globalist organization where all units contribute interactively (not combinatorially) to a given representation. It permits rapid binding of all activation patterns across PC and FC (i.e., episodic memory), while reducing interference. The LEABRA learning algorithm includes error-driven learning of skills and Hebbian learning with inhibitory competition dynamics. In this framework, the PC and FC modules employ a slow integrative learning that blends many individual experiences to capture the underlying regularities of the world and to support sensory-motor activities. The HC module adds fast learning that retains and discriminates (instead of integrating) the individual experiences. This cooperation between HC and FC/PC reflects in turn the complementary learning paradigms in the brain.

A salient trait of IBCA is the knowledge-dependency merits (generalization, parallelism, flexibility) of content-specific distributed representation in the brain, missing in symbolic models. Complementary learning capacity resolves problems with knowledge consolidation, or transfer of short-term into long-term memory. Higher-level cognition (variable binding, chaining of operation sequences etc.) is a result of emergent power of activation-based processing (invocation, maintenance, and updating of active representations for self-regulation) in the FC module. These capacities have been validated in basic psychological tasks, such as Stroop test, dynamic sorting task, or visual feature binding [29]. However, the fine-grained structure of the architectural modules requires a large number of neurons to simulate cognitive functions, and raises the issues of system scalability. The current architecture is limited to learning the ANN weight parameters, but not the network local structure. Representation of emotions for motivation and setting goals, as well as motor coordination and timing, is still missing. While this type of architecture may be used to explain human behavior probed by psychological or psycholinguistic experiments no-one has yet demonstrated how to use it for tasks that require reasoning, where many symbolic architectures reach the human competence level.

Cortronics is a new emergent architecture that models the biological functions of the cerebral cortex and thalamus systems (jointly termed thalamocortex) in the human brain [14]. Its memory organization consists of modular feature attractor circuits called lexicons. Each lexicon comprises further a localist cortical patch, a localist thalamic patch, and the reciprocal connections linking them. Essentially, each lexicon implements a large stable set of attractive states called symbols, each represented by a specific group of neurons. The number of neurons overlapping between each pair of symbols is relatively small, and each neuron representing one symbol may be used to represent many symbols. Accordingly, knowledge in Cortronics takes the form of parallel, indirect unidirectional links between the neurons representing one symbol in a lexicon and those describing a symbol in another lexicon. Each such knowledge link is termed an item of knowledge, and the collection of all these links is called a knowledge base. The union of cortical patches of all lexicons constitutes in turn the entire cortex, while that of the thalamic patches of all lexicons forms only a portion of thalamus. A competitive activation of symbols of lexicons, called confabulation, is used for learning and information retrieval. Confabulation is carried out by every lexicon when appropriate knowledge link inputs, and operation command inputs, arrive at the lexicon at once. The states of involved neurons

evolve dynamically via the parallel, mutual interactions of these neurons, and the minority that ends up in the excited/active state denote conclusions, the symbol(s) that won the competition, or a null symbol, implying a “don’t know” answer. The model predicts or anticipates the next state, move or word that should follow. This is quite new architecture and it is not yet clear how it can be extended to create AGI, as confabulation is not sufficient for reasoning with complex knowledge. However, it is an interesting process involved in anticipation, imagination and creativity [32][33], a process at a shorter time scale than reasoning processes.

The **NuPIC** (*Numenta Platform for Intelligent Computing*) is an emergent architecture based on the Hierarchical Temporal Memory (HTM) technology, which is modeled on the putative algorithm used by neocortex [12]. Network nodes are organized in a hierarchical way, with each node implementing learning and memory functions. Hierarchical organization is motivated by the growing size of cortical receptive fields in the information streams that connect primary sensory cortices with secondary and higher-level association areas. This feature is also present in the IBCA architecture, where specific connectivity between different layers leads to growing and invariant object representation. HTM is unique in stressing the temporal aspect of perception, implementing memory for sequences of patterns that facilitate anticipation. Each level in the hierarchical network is trained separately to memorize spatio-temporal objects, and is able to recognize new, similar objects in a bottom-up/top-down process. The architecture has not yet been tested in larger applications, therefore it is hard to evaluate its potential and its limitations.

NOMAD (Neurally Organized Mobile Adaptive Device) automata are based on “neural Darwinism” theory [34]. Nomads, also known as Darwin automata, demonstrate the principles of emergent architectures for pattern recognition task in real time. They use many sensors for vision, range finders to provide a sense of proximity, priopreceptive sense of head direction and self-movement, artificial whiskers for texture sensing, artificial taste (conductivity) sensors. NOMAD is controlled by a large ($\sim 10^5$ neurons with $\sim 10^7$ synapses) simulated nervous system running on a cluster of powerful computers. The “Brain-Based Robotic Devices” that develop through behavioral tasks has elucidated a role of value systems based on reward mechanisms in adaptation and learning, importance of self-generated movement in development of perception, the role of hippocampus in spatial navigation and episodic memory (Darwin X-XI models), invariant visual object recognition (Darwin VI-VII), binding visual features of complex objects into a unified scene by neural synchrony due to the recurrent connections in visual pathways, implementation of concurrent, real-time processes. However, the emergence of higher-level cognition does not seem likely in this architecture.

A number of emergent architectures based on the global workspace theory of Baars [35] have been formulated in the last two decades, but very few reached implementation level. Shanahan has described a very simple implementation based on weightless neural network built from generalizing random access memory processing units and used to control simulated robot [36]. Other examples of architectures inspired by the global workspace theory are discussed in the hybrid systems subsection below.

Many other interesting emergent architectures have been discussed in recent years, but there is little experience with them due to the lack of software implementation to experiment with. Haikonen [37] has written a book outlining an approach to conscious machines and discussing cognitive architecture for robot brains. Anderson and his colleagues proposed the Erzatz brain project [38]. The autonomous mental development movement, motivated by the human mental development from infancy to adulthood, has been active for about a decade now [39], going in similar direction as the Edelman’s Darwin projects, and Brooks’ Cog project [40][41], that is creating a robotic agent for real-time interaction. Korner and Matsumoto [42] argue that cognitive architecture should control constraints that are used to select a proper algorithm from existing repertoire to solve a specific problem, or to create a new one if the stereotyped behaviors are not sufficient. This is very much in agreement with the meta-learning ideas in computational intelligence [43], where solving hard learning problems is done by learning which transformations should be composed to achieve the goal. The DARPA Biologically-Inspired Cognitive Architectures (BICA) program has already resulted in a several interesting proposals, such as the “TOSCA Comprehensive brain-based model of human mind” [44] written by a number of experts from leading USA institutions, which essentially came to the same conclusion.

3.3. Hybrid Paradigm Architectures

Given the relative strengths of the symbolic and emergent paradigms, it becomes clear that combining the two would offer a promising venue for developing a more complete framework for cognition [45]. Symbolic architectures are able to process information and realize high-level cognitive functions, such as planning and deliberative reasoning, in a way that resembles human expertise. However, the major issues in this approach

are the formulation of symbolic entities from low-level information, as well as the handling of large amount of information and uncertainty. Emergent architectures are better suited for capturing the context-specificity of human performance and handling many pieces of low-level information simultaneously. Yet their main shortcoming is the difficulty in realizing higher-order cognitive functions. The potential benefit of a combined approach is therefore to have each method address the limitations of the other, allowing creation of a complete brain architecture that covers all levels of processing, from stimuli to higher-level cognition.

Research in this area has led to many proposals of hybrid cognitive architectures, which can be roughly divided in two classes based upon the memory type of the constituent modules: *localist-distributed* and *symbolic-connectionist* [45]. The first class of hybrid architectures comprises a combination of localist modules (with each concept specified by one PE node) and distributed modules (with each concept represented by a set of overlapping nodes). In comparison, the second class involves a mixture of symbolic modules (i.e., rule- or graph-based memory) and connectionist modules (either of localist or distributed type). Correspondingly, hybrid architectures can be categorized into two main classes according to their direction of learning: *top-down* and *bottom-up learning* [46]. The former involves a transition of knowledge from explicit (accessible) conceptual level to implicit (inaccessible) sub-conceptual level, while the latter goes from sub-conceptual level to conceptual level. The top-down learning can be achieved by pre-coding a set of expert rules at the top level (localist/symbolic module) and allowing the bottom-level (distributed ANN) to learn by observing actions guided by the top-level [46]. Conversely, bottom-up learning may be accomplished by extracting or translating implicit knowledge coded by a bottom-level module into a set of conceptual rules [30][47]. A few examples of hybrid cognitive architectures follow, focused on the memory organizations, learning methodologies, and key strengths and issues.

ACT-R (*Adaptive Components of Thought-Rational*) [10] is a hybrid cognitive architecture and theoretical framework for emulating and understanding human cognition⁵. It aims at building a system that can perform the full range of human cognitive tasks and describe in detail the mechanisms underlying perception, thinking, and action. The central components of ACT-R comprise a set of perceptual-motor modules, memory modules, buffers, and a pattern matcher. The perceptual-motor modules basically serve as an interface between the system and the world. There are two types of memory modules in ACT-R: declarative memory (DM) and procedural memory (PM), which encode factual knowledge about the world and that about how to do things respectively. Both are realized as a symbolic-connectionist structures, where the symbolic level consists of productions (for PM) or chunks (for DM), and the sub-symbolic level of a massively parallel connectionist structure. Each symbolic construct (i.e., production or chunk) has a set of sub-symbolic parameters that reflect its past usage and control its operations, thus enabling an analytic characterization of connectionist computations using numeric parameters (associative activation) that measure the general usefulness of a chunk or production in the past and current context. Finally, the ACT-R buffers serve as a temporary storage for inter-module communications (excluding PM), while the pattern matcher is used to find a production in PM that matches the present state of the buffers.

ACT-R utilizes a top-down learning approach to adapt to the structure of the environment. In particular, symbolic constructs (i.e., chunks or productions) are first created to describe the results of a complex operation, so that the solution may be available without recomputing the next time a similar task occurs. When a goal, declarative memory activation or perceptual information appears it becomes a chunk in the memory buffer, and the production system guided by subsymbolic processes finds a single rule that responds to the current pattern. Sub-symbolic parameters are then tuned using Bayesian formulae to make the existing symbolic constructs that are useful more prominent. In this way chunks that are often used become more active and can thus be retrieved faster and more reliably. Similarly, productions that more likely led to a solution at a lower cost will have higher expected utility, and thus be more likely chosen during conflict resolution (i.e., selecting one production among many that qualify to fire). This architecture may be partially mapped on the brain structures. It has been applied in a large number of psychological studies, and in intelligent tutoring systems, but ambitious applications to problem solving and reasoning are still missing.

CLARIOn (*The Connectionist Learning Adaptive Rule Induction ON-line*) is a hybrid architecture that incorporates a distinction between explicit (symbolic) and implicit (sub-symbolic) processes and captures the interactions between the two [45]-[47]. The design objective is two-fold: to develop artificial agents for certain cognitive task domains, and to understand human learning and reasoning processes in similar domains. The CLARIOn architecture contains four memory modules, each comprising a dual explicit-implicit representation: action-centered subsystem (ACS), non-action-centered subsystem (NCS), motivational

⁵ See the ACT-R site at <http://act-r.psy.cmu.edu>

subsystem (MS), and metacognitive subsystem (MCS). Essentially, the ACS module serves to regulate the agent's actions, while NCS maintain the general system knowledge (either explicit or implicit). On the other hand, MS functions to provide a motivation/impetus for perception, action and cognition, while MCS monitor, direct and alter the operations of the other three modules. Each of these modules adopts a localist-distributed representation, where the localist section encodes the explicit knowledge and the distributed section (e.g. an MLP network) the implicit knowledge. CLARION also employs different learning methods for each level of knowledge. Learning of implicit knowledge is achieved using reinforcement learning methods such as Q-learning or supervised methods such as the standard back-propagation, both of which can be implemented using an MLP network. The implicit knowledge already acquired at the bottom level is then utilized to craft the explicit knowledge at the top level via a bottom-up learning. This can in turn be viewed as a rational reconstruction of implicit knowledge at the explicit level. Top-down learning may also be achieved by precoding/fixing some rules at the top level and allowing the bottom-level to accumulate knowledge by observing actions guided by these rules [46]. As such, the system's decision making that relies initially on the top level gradually becomes more dependent on the bottom level. Software is available for experimentation with CLARION. A lot of psychological data has been simulated with this architecture, but also a complex sequential decision-making for a minefield navigation task.

LIDA (*The Learning Intelligent Distribution Agent*) is a conceptual and computational framework for intelligent, autonomous, “conscious” software agent that implements some ideas of the global workspace (GW) theory [48]. The architecture is built upon a bit older IDA framework, which was initially designed to automate the whole set of tasks of a human personnel agent who assigns sailors to new tours of duty. LIDA employs a partly symbolic and partly connectionist memory organization, with all symbols being grounded in the physical world in the sense of Brooks and Stein [41]. LIDA has distinct modules for perception, working memory, emotions, semantic memory, episodic memory, action selection, expectation and automatization (learning procedural tasks from experience), constraint satisfaction, deliberation, negotiation, problem solving, metacognition, and conscious-like behavior. Most operations are done by codelets implementing the unconscious processors (specialized networks) of the global workspace theory. A codelet is a small piece of code or program that performs one specialized, simple task. The LIDA framework incorporates three new modes of learning into the older IDA model: perceptual, episodic, and procedural learning, which are all of bottom-up type. Perceptual learning concerns learning of new objects, categories, relations, etc, and takes two forms: strengthening or weakening of the base-level activation of nodes, as well as creation of new nodes and links in the perceptual memory. Episodic learning, on the other hand, involves learning to memorize specific events (i.e., the what, where, and when). It results from events taken from the content of “consciousness” being encoded in the (transient) episodic memory. Finally, procedural learning concerns learning of new actions and action sequences with which to accomplish new tasks. It combines selectionist learning (i.e., selecting from an obsolete repertoire) and instructionalist learning (i.e., constructing new representations), with functional consciousness providing reinforcements to actions. There is no doubt that this architecture may explain many features of mind, however, it remains to be seen how competence it will achieve in understanding language, vision, and common sense reasoning based on perceptions.

DUAL architecture [49] has been inspired by Minsky's “Society of Mind” [50] theory of cognition. It is a hybrid, multi-agent general-purpose architecture supporting dynamic emergent computation, with a unified description of mental representation, memory structures, and processing mechanisms carried out by small interacting micro-agents. As a result of lack of central control the system is constantly changing, depending on the environment. Agents interact forming larger complexes, coalitions and formations, some of which may be reified. Such models may be evaluated at different levels of granularity, the microlevel of micro-agents, the mesolevel of emergent and dynamic coalitions, and the macrolevel of the whole system and models, where psychological interpretations may be used to describe model properties. Micro-frames are used for symbolic representation of facts, while relevance or activation level of these facts in a particular context is represented by network connections with spreading activation that changes node accessibility. Links between microagents are based on their frame slots and weights control the influence of agents on each other's activity. DUAL architecture has been used in a number of projects: AMBR, a model of human reasoning that unifies analogy, deduction, and generalization, including a model of episodic memory; a model of human judgment; a model of perception, analysis of interactions between analogy, memory, and perception; understanding the role of context and priming effects for the dynamics of cognitive processes. This is certainly a very interesting architecture that is capable of explaining many cognitive phenomena. It is not clear how well it will scale up to real problems requiring complex reasoning, as nothing in this area has yet been demonstrated.

Polyscheme [51] integrates multiple methods of representation, reasoning and inference schemes in problem solving. Each Polyscheme “specialist” models a different aspect of the world using specific representation and inference techniques, interacting with other specialists and learning from them. Scripts, frames, logical propositions, neural networks and constraint graphs can be used to represent knowledge. A reflective specialist guides the attention of the whole system, providing various focus schemes that implement inferences via script matching, backtracking search, reason maintenance, stochastic simulation and counterfactual reasoning. High-order reasoning is guided by higher-level policies for focusing attention. Many problem solving algorithms use forward inference, subgoaling, grounding, representing alternate worlds and identity matching as their basic operations. Such operations are handled by specialists who are equipped with different representations but focus on the same aspect of the world, and may integrate also lower-level perceptual and motor processes. Thus Polyscheme may be used both in abstract reasoning and also in common sense physical reasoning in robots. It has been used to model infant reasoning including object identity, events, causality, spatial relations. This meta-learning approach combining different approaches to problem solving is certainly an important step towards AGI and common sense reasoning.

4CAPS architecture [52] has plausible neural implementation and is designed for complex tasks, such as language comprehension, problem solving or spatial reasoning. A unique feature is the ability to compare the activity of different 4CAPS modules with functional neuroimaging measures of brain’s activity. It has been used to model human behavioral data (response times and error rates) for analogical problem solving, human-computer interaction, problem solving, discourse comprehension and other complex tasks solved by normal and mentally impaired people. Its first operating principle, “Thinking is the product of the concurrent activity of multiple centers that collaborate in a large scale cortical network”, leads to the architecture based on a number of centers (corresponding to particular brain areas) that have different processing styles, for example Wernicke’s area is specialized for the associative retrieval/design, constructing and selectively accessing structured sequential and hierarchical representations. Each center can perform and be a part of multiple cognitive functions, but has a limited computational capacity constraining its activity. Functions are assigned to centers depending on the resource availability, therefore the topology of the whole large-scale network is not fixed. Although 4CAPS contains many interesting ideas it is not aimed at achieving intelligent behavior, but rather tries to model human performance; software written in Lisp is available for experimentation. See the discussion in [52] of other models that are aimed at explanation of behavioral data.

Shruti [53], biologically-inspired model of human reflexive inference, represents in connectionist architecture relations, types, entities and causal rules using focal-clusters. These clusters encode universal/existential quantification, degree of belief, and the query status. The synchronous firing of nodes represents dynamic binding, allowing for representations of quite complex knowledge and inferences. This architecture may have great potential, but after rather long time of development it has not yet found any serious applications to problem solving or language understanding.

The Novamente AI Engine is based on system-theoretic ideas regarding complex mental dynamics and associated emergent patterns, inspired by the *psynet* model [54] and more general “patternist philosophy of mind” [55]. Similarly as in the “society of minds” and the global workspace, self-organizing and goal-oriented interactions between patterns are responsible for mental states. Emergent properties of network activations should lead to hierarchical and relational (heterarchical) pattern organization. Probabilistic term logic (PTL), and the Bayesian Optimization Algorithm (BOA) algorithms are used for flexible inference. Actions, perceptions, and internal states are represented by tree-like structures. This is still an experimental architecture that is being developed, seems to be in a fluid state, and its scaling properties are not yet known.

4. Where do we go from here?

The previous sections has presented a number of very interesting models of cognition that have the potential to develop general intelligence. Many excellent projects have already been formulated, some have been developed over many decades, while others are just starting. So far cognitive architectures are used in very few real-world applications. Grand challenges, as discussed in section two, and smaller steps that lead to human and super-human levels of competence should be formulated to focus the research. Extending small demonstrations in which a cognitive system reasons in a trivial domain to larger-scale applications, for example generating results that may be of interest to experts, or acting as an assistant to human expert, is one important direction. Without a set of demanding test problems it is very hard to evaluate new projects, compare their capabilities and understand their limitations. Integrative models of human performance are of

great interest in the defense and aerospace industries. A recent project on the Agent-Based Modeling and Behavior Representation (AMBR) Model Comparison resulted in quantitative data comparing the performance of humans and cognitive architectures in a simplified air traffic controller environment [56]. Some efforts have been expended on the evaluation of software agents and several proposals in this direction has been put forth during the 2007 AAAI Workshop “Evaluating Architectures for Intelligence” [57]. Ideas ranged from using in-city driving environment as a testbed for evaluating cognitive architectures, to measuring incrementality and adaptivity components of general intelligent behavior.

Perhaps a measure of “cognitive age” could be established, with a set of problems that children at a given age are able to solve. Problems should be divided into several groups: e.g. vision and auditory perception, understanding language, common-sense reasoning, abstract reasoning, probing general knowledge about the world, learning, problem solving, imagination, creativity. Solving all problems from a given group that children at some age are able to solve will qualify cognitive system to pass to the next grade in this group of problems. It should be expected that some systems will show advanced age in selected areas, and not in the others. For example, solving problems requiring vision may require addition of specialized computer vision modules, while mathematical reasoning in many reasoning systems may be fairly advanced comparing to children. Experts in human intelligence largely agree to the original Gardner’s proposal [58] that seven kinds of intelligence should be distinguished: logical-mathematical, linguistic, spatial, musical, bodily-kinesthetic, interpersonal and intrapersonal intelligence, perhaps extended by emotional intelligence and a few others.

General world knowledge is fairly difficult to collect and could be probed using a question/answer system. If a 5-year old child could get all the answer to general questions from an avatar controlled by some cognitive architecture one should assume that the mental age of the control system in this respect is at least 5. Knowledge bases in cognitive systems are usually quite limited and require very different kind of organization and knowledge representation methods. Huge CyC knowledge base is an exception [3], and using it to construct large knowledge bases suitable for other cognitive systems is certainly worth the effort.

Such analysis should certainly help to understand what type of intelligence may be expected from embodied cognitive robotic projects and what the limitations of symbolic approaches are. Brooks has made a good point that elephants do not play chess [40], and expressed hope [41] that a robot with integrated vision, hearing and dexterous manipulation controlled by large scale parallel MIMD computer “will learn to ‘think’ by building on its bodily experiences to accomplish progressively more abstract tasks”. His Cog project based on grounding the meaning of concepts in deep embodiment has many followers although after 15 years it has stayed at the level of reactive agent and there are no good ideas how to extend it to higher cognitive levels. While behavioral intelligence in robotic devices may be difficult to achieve without embodiment experiences with this approach in the last two decades are not very encouraging for AGI. Elephants are intelligent, but cannot learn language or be personal assistants. It is also possible that ideas on which cognitive architectures are based are not sufficient to solve the problems in computer vision or language and more specific models of some brain functions are needed.

The survey presented above showed several trends that will probably dominate in the research on cognitive architectures. First, the number of hybrid architectures is already quite large, biological inspirations are becoming increasingly important and this will lead to domination of BICA architectures. Even hard core symbolic architecture proponents base now further extension of their architectures on inspirations from the brain (see link in footnote 3). They focus on the role of cortex and limbic system, but completely neglect the regulatory role of the brain stem which may provide overall meta-control selecting different types of behavior. Second, there may be many BICA architectures, but several key features need to be preserved. Different types of memory are certainly important, as has been already stressed by several symbolic, emergent and hybrid architectures. Processing of speech or texts requires recognition of tokens, or mapping from sounds or strings of letters to unique terms; resolving ambiguities and mapping terms to concepts in some ontology; and a full semantic representation of the text, that facilitates understanding and answering questions about its content. These three steps are roughly based on several kinds of human memory.

First, recognition memory that helps to focus quickly attention when something is wrong, for example a strangely spelled word that could be a misspelling, a foreign word, personal name, or an attempt to avoid spam filters. This may be implemented by simple neural networks without hidden layer or by correlation matrix memories [32]. The role of recognition memory has also been largely forgotten.

Second, there is a need for semantic memory that serves not only as hierarchical ontology, but approximates spreading activation processes in real brains, and thus activates various types of associations providing background knowledge that humans use for token to concept mapping and disambiguation. Unfortunately large-scale semantic memories that contain both structural properties of concepts (chairs have

legs, seat, etc) and their relations and associations (chair – table, sit, etc) and could be used in computationally efficient way do not exist. While significant progress has been made in drawing inspirations from neuroscience in analysis of auditory, visual and olfactory signals much less has been done at the higher cognitive function level. Although neurocognitive approach to linguistics has been formulated as “an attempt to understand the linguistic system of the human brain, the system that makes it possible for us to speak and write, to understand speech and writing, to think using language ...” [59], in practice it has been used only to analyze specific linguistic phenomena. A practical algorithm to discover these “pathways of the brain” has been introduced recently [60], opening the way for construction of large-scale semantic networks that will approximate symbolic knowledge stored in human brain, although creating such large-scale memories will require a large effort. Efforts to build concept descriptions from electronic dictionaries, ontologies, encyclopedias, results of collaborative projects and active searches in unstructured sources have been described in [6].

Third, episodic memory is required to store experiences from interactions with individual users, to understand the context of current interactions and interpret all events in view of this context. Various elements of semantic and episodic memories are kept in the working memory. All types of memory are intimately connected. Recognition of tokens is facilitated by the active part of semantic memory and made easier by the expectations resulting from episodic memory. Reading text leads to priming effects: expectation and anticipation of a few selected words, and inhibition of many others that do not come to the mind of the reader. Episodic memory is based on semantic relations of the concepts found in the text. Although several proposals for memory-based cognitive architectures have been formulated the role of different types of memory has not been stressed and no effort to create appropriate large-scale knowledge bases has been made. AGI requires such memories, and as a step towards such memory-based architecture an avatar that uses large semantic memory to play word games has been demonstrated [6].

What is the best practical way to implement these ideas? Template matching proved to be amazingly effective in simulation of dialogues and is still dominating in chatterbots [61], but it obviously does not lead to real understanding of the concepts that appear in the discussion. Neural template matching, or templates approximating the distribution of neuronal group activities in the brain during concept comprehension, is the simplest technique that goes beyond symbolic template matching, leading to sets of consistent concepts. Words are ambiguous and form concepts that have meanings modified by their contexts. In the brain a word $w = (w_f, w_s)$ has phonological component w_f (the memorized form of the word, string of phonemes or characters), and an extended semantic representation w_s (extended activations related to the use and category of the word, including immediate associations). The extended activation is not unique, only when the current context $Cont$ is specified (specific activations of other concepts are determined) the meaning of the word is established, resulting from spreading activation in the brain to form a global state $\Psi(w, Cont)$. This state changes with each new word received in sequence, with quasi-stationary states formed after each sentence is processed and understood. It is quite difficult to decompose the $\Psi(w, Cont)$ state into components, because the semantic representation w_s is strongly modified by the context. The state $\Psi(w, Cont)$ may be regarded as a quasi-stationary wave, with its core component centered on the phonological/visual brain activations w_f and with quite variable extended representation w_s . As a result the same word in a different sentence creates quite different states of activation, and the lexicographical meaning of the word may be only an approximation of an almost continuous process. To relate states $\Psi(w, Cont)$ to lexicographical meanings, one can cluster all such states for a given word in different contexts and define prototypes $\Psi(w_k, Cont)$ for different meanings w_k . These prototypes are neural templates that should replace symbolic templates. The form of the word w_f identifies several candidate templates with different meanings, and the one that fits to other templates, maximizing overall consistency of interpretations, is selected.

The symbolic approach to language is a poor substitute for neurolinguistic processes, and high-dimensional vector model of language, popular in statistical approach to natural language processing (NLP) [62], is a very crude approximation that does not reflect essential properties of the perception-action-naming activity of the brain [64][65]. The process of understanding words (spoken or read) starts from activation of word form representation (the phonological or grapheme representation) in the temporal lobe, quickly spreading neural activation to further brain areas, including the non-dominant (usually right) hemisphere, that does not contain representations of word forms, but learns to evaluate clusters of activations, forming constraints on the way words may be used, and forming general, higher-level concepts [60]. This continuous process may be approximated through a series of snapshots of patterns created by microcircuit activations $\phi(w, Cont)$ that can be treated as basis functions for the expansion of the state $\Psi(w, Cont) = \sum_i \alpha_i \phi_i(w, Cont)$, where the summation extends over all patterns that show significant activity resulting after presentation of the

word w . The high-dimensional vector model used in NLP measures only the co-occurrence of words $\mathbf{V}_{ij} = \langle \mathbf{V}(w_i), \mathbf{V}(w_j) \rangle$ in small window neighborhood, averaged over all contexts, while human knowledge includes also structural properties of concepts that are important, but do not appear explicitly in texts. The use of wave-like representation in terms of basis functions to describe neural states makes this formalism similar to that used in quantum mechanics, although no real quantum effects are implied here. Objects of discourse and actual episodes are memorized by hippocampus that links to the cortex and is able to recreate the original activations at the moment the episode has been experienced.

An outline of the road from single neurons, to brain modules, to societies of brains has been presented in [66]. Models of single neurons usually have little internal knowledge (thresholds) and relatively simple interactions modeled via weighted links (biophysical models of single neurons may of course be very complex). Assemblies of neurons at different levels form coalitions and may be regarded as specialized processors, passing structured information and attaining rather complex internal states. This may be approximated by interacting agents, with internal knowledge and means of communication, with coalitions of simple agents creating dynamic, higher-order units that may be viewed as “soft agents”, with new competencies arising at demand by variation on the main theme. Such meta-learning ideas for beating combinatorial complexity and solving pattern recognition and reasoning based on partial observations has been recently described [43], and preliminary implementation of general system architecture to support such approach has been presented [67].

The role of imagination, creativity, learning from partial observations and using this knowledge in an intuitive way, and the role of the right hemisphere in the linguistic processes, has only recently been discussed [32][33]. The AIM1 (Artificial Mind1) architecture based on these ideas is under development and will be presented in near future. This architecture will draw inspirations from some of the projects presented in this paper, but will be primarily aimed at ambitious applications requiring natural language processing. It is quite likely that this approach will lead to creation of conscious artifacts [37][68].

Acknowledgment: W.D. thanks Polish Committee for Scientific Research for research grant 2005-2007.

References

- [1] A. Newell, *Unified Theories of Cognition*: Harvard University Press, 1990.
- [2] A. Newell, H.A. Simon, GPS: A program that simulates human thought. In E. A. Feigenbaum & J. Feldman (Eds.), *Computers and Thought*. New York: McGraw-Hill, 1963.
- [3] K. Panton, C. Matuszek, D. Lenat, D. Schneider, M. Witbrock, N. Siegel, B. Shepard, Common Sense Reasoning – From Cyc to Intelligent Assistant. In: Y. Cai and J. Abascal (Eds.): *Ambient Intelligence in Everyday Life*, LNAI **3864** (2006) 1–31.
- [4] A. Turing, Computing Machinery and Intelligence, *Mind* **49** (1950) 433–460.
- [5] R. Carpenter, J. Freeman, Computing Machinery and the Individual: the Personal Turing Test, 2005, paper available at <http://www.jabberwacky.com/>.
- [6] J. Szymanski, T. Sarnatowicz, W. Duch, Towards Avatars with Artificial Minds: Role of Semantic Memory. *Journal of Ubiquitous Computing and Intelligence* **1** (2007, in print).
- [7] E.A. Feigenbaum, Some Challenges and Grand Challenges for Computational Intelligence. *J. of the ACM* **50**(1) (2003) 32–40.
- [8] N. Nilsson, Human-Level Artificial Intelligence? Be Serious! *The AI Magazine* **26**(4) (2005) 68–75.
- [9] D.E. Meyer, D.E. Kieras, A computational theory of executive cognitive processes and multiple-task performance: Part 1. Basic mechanisms. *Psychological Review* **104**(1), (1997) 3–65.
- [10] J.R. Anderson, C. Lebiere, The Newell test for a theory of cognition. *Behavioral and Brain Science* **26** (2003) 587–637.
- [11] D. Vernon, G. Metta, G. Sandini, A survey of artificial cognitive systems: Implications for the autonomous development of mental capabilities in computational agents. *IEEE Transactions on Evolutionary Computation* **11**(2) (2007) 151–180.
- [12] J. Hawkins, S. Blakeslee, *On intelligence: How a New Understanding of the Brain will Lead to the Creation of Truly Intelligent Machines*. Times Books 2004.
- [13] T. Hoya, *Artificial Mind System. Kernel Memory Approach*. Springer, 2005.
- [14] R. Hecht-Nielsen, *Confabulation Theory: The Mechanism of Thought*. Springer 2007.
- [15] J.F. Sowa, *Conceptual Structures*. Reading, Mass, Addison-Wesley, 1984.
- [16] M. Minsky, A Framework for Representing Knowledge. In: P.H. Winston, Ed. *The Psychology of Computer Vision*. New York: McGraw-Hill, 1975.
- [17] R.J. Kirby, *Adaptive Execution in Complex Dynamic Worlds*. Ph.D. Thesis, Yale University, 1989.
- [18] T.M. Mitchell, R., Keller, S. Kedar-Cabelli, Explanation-based generalization: A unifying view. *Machine Learning* **1** (1986) 47–80.
- [19] M.M. Veloso, J.G. Carbonell, Integrating analogy into a general problem-solving architecture. In M. Zemankova & Z. Ras (Eds.), *Intelligent Systems* (pp. 29–51). Chichester, England: Ellis Horwood, 1990.
- [20] N. Larvac, S. Dzeroski, *Inductive Logic Programming: Techniques and Applications*. New York: Ellis Horwood, 1994.
- [21] L.P. Kaelbling, M.L. Littman, A.W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research* **4** (1996) 237–285.
- [22] J.E. Laird, P.S. Rosenbloom, A. Newell, Soar: An architecture for general intelligence. *Artificial Intelligence* **33** (1987) 1–64.

- [23] J. Rosbe, R.S. Chong, D.E. Kieras, Modeling with Perceptual and Memory Constraints: An EPIC-Soar Model of a Simplified Enroute Air Traffic Control Task, SOAR Technology Inc. Report, Ann Arbor, Michigan, 2001.
- [24] P. Langley, An adaptive architecture for physical agents. In *Proc. of the 2005 IEEE/WIC/ACM Int. Conf. on Intelligent Agent Technology*. Compiegne, France: IEEE Computer Society Press, pp. 18-25, 2005.
- [25] P. Langley, D. Choi, Learning recursive control programs from problem solving. *J. of Machine Learning Res.* **7** (2006) 493-518.
- [26] P. Wang, Rigid flexibility. The Logic of Intelligence. Springer 2006
- [27] S.C. Shapiro, W.J. Rapaport, M. Kandefler, F.L. Johnson, A. Goldfain, Metacognition in SNePS, *AI Magazine* **28** (2007) 17-31.
- [28] J.L. McClelland, D.E. Rumelhart and the PDP Research Group. *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*. Cambridge, MA: MIT Press, 1986.
- [29] R.C. O'Reilly, Y. Munakata, *Computational Explorations in Cognitive Neuroscience: Understanding of the Mind by Simulating the Brain*. Cambridge, MA: MIT Press, 2000.
- [30] W. Duch, R. Adamczak, K. Grabczewski, A new methodology of extraction, optimization and application of crisp and fuzzy logical rules. *IEEE Transactions on Neural Networks*, **12** (2001) 277-306.
- [31] R.C. O'Reilly, T.S. Braver, J.D. Cohen A biologically-based computational model of working memory. In A. Miyake & P. Shah (Eds.), *Models of Working Memory*. Cambridge University Press, pp. 375-411, 1999.
- [32] W. Duch, M. Pilichowski, Experiments with computational creativity. *Neural Information Processing – Letters and Reviews* **11** (2007) 123-133.
- [33] W. Duch, Intuition, Insight, Imagination and Creativity. *IEEE Computational Intelligence Magazine* **2**(3) (2007) 40-52.
- [34] G.M. Edelman, Neural Darwinism: Selection and reentrant signaling in higher brain function. *Neuron* **10** (1993) 115-125.
- [35] B.J. Baars, *A Cognitive Theory of Consciousness*. New York: Cambridge University Press, 1988.
- [36] M.P. Shanahan, A cognitive architecture that combines internal simulation with a global workspace. *Consciousness and Cognition* **15** (2006) 157-176.
- [37] P. Haikonen, *Robot brains: circuits and systems for conscious machines*. Wiley, 2007.
- [38] J.A. Anderson, P. Allopenna, G.S. Guralnik, D. Sheinberg, J.A. Santini, Jr., D. Dimitriadi, B.B. Machta, and B.T. Merritt, Programming a Parallel Computer: The Ersatz Brain Project. In W. Duch, J. Mandziuk (Eds.), *Challenges to Computational Intelligence*. Springer: Berlin, pp. 61-88, 2007.
- [39] J. Weng and W. S. Hwang, From Neural Networks to the Brain: Autonomous Mental Development. *IEEE Computational Intelligence Magazine* **1**(3) (2006) 15-31.
- [40] R. Brooks, Elephants don't play chess. *Robotics and Autonomous Systems* **6** (1986) 3-15.
- [41] R. Brooks, L.A. Stein, Building Brains for Bodies. *Autonomous Robotics* **1** (1994) 7-25.
- [42] E. Korner, G. Matsumoto, Cortical architecture and self-referential control for brain-like computation. *IEEE Engineering in Medicine and Biology Magazine*, **21**(5) (2002) 121-133.
- [43] W. Duch, Towards comprehensive foundations of computational intelligence. In: W. Duch and J. Mandziuk, *Challenges for Computational Intelligence*. Springer Studies in Computational Intelligence, **63** (2007) 261-316.
- [44] TOSCA: A comprehensive brain-based cognitive architecture: *Biologically-Inspired Cognitive Architecture* (BICA) Phase 1 Architecture Report, DARPA-IPTO 2006.
- [45] R. Sun, F. Alexandre, *Connectionist symbolic integration*. Hillsdale, NJ: Erlbaum, 1997.
- [46] R. Sun, X. Zhang, Top-down versus bottom-up learning in cognitive skill acquisition. *Cognitive Systems Research* **5** (2004) 63-89.
- [47] R. Sun, E. Merrill, T. Peterson, From implicit skills to explicit knowledge: A bottom-up model of skill learning. *Cognitive Science*, **25**(2) (2001) 203-244.
- [48] S. Franklin, The LIDA architecture: Adding new modes of learning to an intelligent, autonomous, software agent. In *Proc. of the Int. Conf. on Integrated Design and Process Technology*. San Diego, CA: Society for Design and Process Science, 2006.
- [49] A. Nestor, B. Kokinov, Towards Active Vision in the DUAL Cognitive Architecture. *International Journal on Information Theories and Applications* **11** (2004) 9-15.
- [50] M. Minsky, *The Society of Mind*. Simon and Schuster, New York, 1986.
- [51] N.L. Cassimatis, Adaptive Algorithmic Hybrids for Human-Level Artificial Intelligence. *Advances in Artificial General Intelligence*. IOS Press. Eds. B. Goertzel and P. Wang, 2007.
- [52] M.A. Just, S. Varma, The organization of thinking: What functional brain imaging reveals about the neuroarchitecture of complex cognition. *Cognitive, Affective, and Behavioral Neuroscience* **7** (2007) 153-191.
- [53] L. Shastri, V. Ajjanagadde, From simple associations to systematic reasoning: A connectionist encoding of rules, variables, and dynamic bindings using temporal synchrony. *Behavioral & Brain Sciences* **16**(3) (1993) 417-494.
- [54] B. Goertzel, *From Complexity to Creativity*. New York, NY: Plenum Press, 1997.
- [55] B. Goertzel, *The Hidden Pattern*, BrownWalker Press, 2006.
- [56] K.A. Gluck, R.W. Pew (Eds.), *Modeling Human Behavior with Integrated Cognitive Architectures: Comparison, Evaluation*. Lawrence Erlbaum Associates 2005.
- [57] G.A. Kaminka, C.R. Burghart (eds.), Evaluating Architectures for Intelligence. Technical Report WS-07-04, AAAI Press, Menlo Park, CA, 2007.
- [58] H. Gardner, *Multiple intelligences: The theory in practice*. New York: Basic Books, 1993.
- [59] S. Lamb, *Pathways of the Brain: The Neurocognitive Basis of Language*. Amsterdam: J. Benjamins Publishing Co. 1999.
- [60] W. Duch, P. Matykiewicz, J. Pestian, Towards Understanding of Natural Language: Neurocognitive Inspirations. Lecture Notes in Computer Science **4669** (2007) 953-962.
- [61] R. Wallace, *The Elements of AIML Style*, ALICE A.I. Foundation, 2003.
- [62] C.D. Manning, H. Schütze, *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press, 1999.
- [63] F. Pulvermüller, *The Neuroscience of Language. On Brain Circuits of Words and Serial Order*. Cambridge, UK: Cambridge University Press, 2003.
- [64] S. Dehaene, L. Cohen, M. Sigman, F. Vinckier, The neural code for written words: a proposal. *Trends in Cognitive Science* **9**, 335-341, 2005.
- [65] M. Meeter, J.M.J. Murre, TraceLink: A model of consolidation and amnesia. *Cognitive Neuropsychology* **22** (5), 559-587, 2005.
- [66] W. Duch, J. Mandziuk, Quo Vadis Computational Intelligence? In: *Machine Intelligence. Quo Vadis?* Eds: P. Sinčák, J. Vaščák, K. Hirota. Advances in Fuzzy Systems - Applications and Theory - Vol. 21, World Scientific, 3-28, 2004.

- [67] K. Grabczewski, N. Jankowski, Versatile and Efficient Meta-Learning Architecture: Knowledge Representation and Management in Computational Intelligence. *IEEE Symposium Series on Computational Intelligence (SSCI 2007)*, Honolulu, HI, IEEE Press, pp. 51-58.
- [68] W. Duch, Brain-inspired conscious computing architecture. *Journal of Mind and Behavior* 26(1-2) (2005) 1-22.

LIDA and a Theory of Mind

David Friedlander^a and Stan Franklin^b

^a *Cognitive Computing Research Group, Houston, TX*

^b *Computer Science Department & Institute for Intelligent Systems,
The University of Memphis*

Abstract. Every agent aspiring to human level intelligence, every AGI agent, must be capable of a theory of mind. That is, it must be able to attribute mental states, including intentions, to other agents, and must use such attributions in its action selection process. The LIDA conceptual and computational model of cognition offers an explanation of how theory of mind is accomplished in humans and some other animals, and suggests how this explanation could be implemented computationally. Here we describe how the LIDA version of theory of mind is accomplished, and illustrate it with an example taken from an experiment with monkeys, chosen for simplicity.

Keywords. LIDA, Theory of Mind, Cognitive Architecture, Cognitive Science, Artificial Intelligence, Global Workspace Theory

Introduction

The theory of mind states that we ascribe minds to other individuals, and particularly attribute mental states to them even though, as individuals, we only have direct evidence of our own mental states [1]. We perceive the minds of others in the same way we perceive other aspects of our environment, by applying cognitive processes to sensory inputs. We use the results of such perception to select actions, and to learn from our perceptions of their effects. One can argue that a theory of mind process would be necessary for an artificial general intelligence (AGI) agent.

The central hypothesis of this paper is that the mind has the ability to build models of other cognitive agents in hypothetical environments and to reason about potential outcomes of possible actions as an aid to decision making. One possible objection is that, not only are our perceptions about other people's minds indirect, they are also sparse. However, the human mind is quite good at making sense of sparse data. There is a good analogy with visual perception. In vision, the fovea scans only a small part of the scene [2]. No detail is available for most of it and, for some of it, no data at all. Yet we perceive a continuous, fully filled in, detailed scene at all times. The brain supplies the missing perceptual qualities from a number of sources [3]: previous input from the recent past; world knowledge such as common sense or convention; abstract default information; similar information from the past. We propose that similar sources are used to make sense of sparse data while modeling other minds.

In this paper, we will use a cognitive model called LIDA [4], derived from Global Workspace Theory [5,6] and based on recent theories from psychology, neuroscience and cognitive science, to develop our hypothesis of the theory of mind and to show how it can explain various psychological phenomena. Section 1 contains a brief

description of the LIDA model, Section 2 shows how it can incorporate a theory of mind, Section 3 provides a specific example of how the model would work, and Section 4 contains conclusions and suggestions for future research.

1. LIDA

The LIDA model and its ensuing architecture are grounded in the LIDA cognitive cycle. As a matter of principle, every autonomous agent [7], be it human, animal, or artificial, must frequently sample (sense) its environment, process (make sense of) this input, and select an appropriate response (action). The agent's "life" can be viewed as consisting of a continual sequence of iterations of these cognitive cycles. Such cycles constitute the indivisible elements of attention, the least sensing and acting to which we can attend. A cognitive cycle can be thought of as a moment of cognition, a cognitive "moment." Higher-level cognitive processes are composed of these cognitive cycles as cognitive "atoms." Just as atoms are composed of protons, neutrons and electrons, and some of these are composed of quarks, glueons, etc., these cognitive "atoms" have a rich inner structure. We'll next concisely describe what the LIDA model hypothesizes as the rich inner structure of the LIDA cognitive cycle. More detailed descriptions are available elsewhere [8,9,10,11]. Figure 1 should help the reader follow the description. It starts in the upper left corner and proceeds clockwise.

During each cognitive cycle the LIDA agent first makes sense of its current situation as best as it can. It then decides what portion of this situation is most in need of attention. Broadcasting this portion, the current contents of consciousness, enables the agent to finally chose an appropriate action and execute it. Let's look at these three processes in a little more detail.

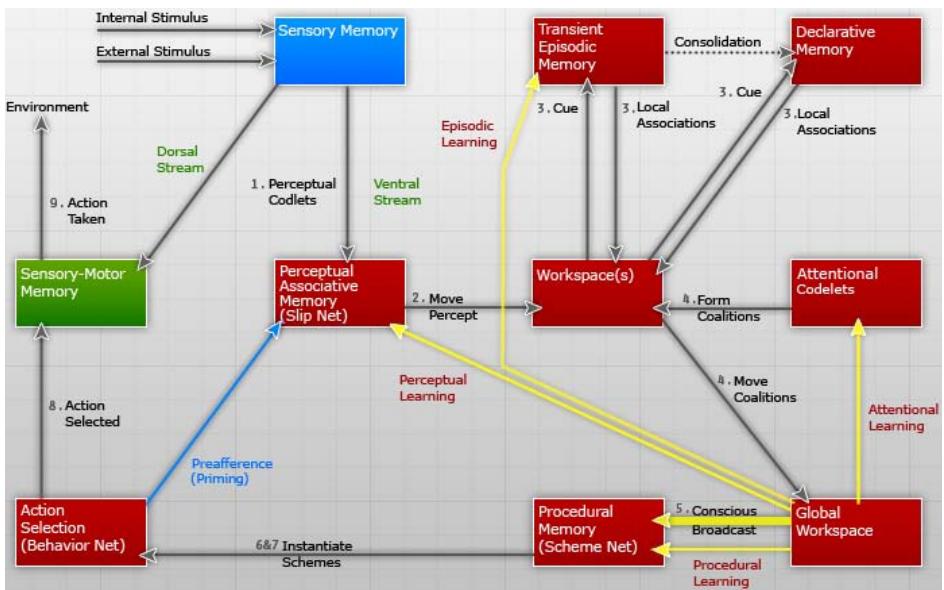


Figure 1. The LIDA Cognitive Cycle

The cycle begins with sensory stimuli from the agent's environment, both an external and an internal environment. Low-level feature detectors in sensory memory begin the process of making sense of the incoming stimuli. These low-level features are passed to perceptual memory where higher-level features, objects, categories, relations, actions, situations, etc. are recognized. These recognized entities, comprising the percept, are passed to the workspace, where a model of the agent's current situation is assembled. Workspace structures serve as cues to the two forms of episodic memory, yielding both short and long term remembered local associations. In addition to the current percept, the workspace contains recent percepts that haven't yet decayed away, and the agent's model of the then current situation previously assembled from them. The model of the agent's current situation is updated from the previous model using the remaining percepts and associations. This updating process will typically require looking back to perceptual memory and even to sensory memory, to enable the understanding of relations and situations. This assembled new model constitutes the agent's understanding of its current situation within its world. It has made sense of the incoming stimuli.

For an agent "living" in a complex, dynamically changing environment, this updated model may well be much too much for the agent to deal with all at once. It needs to decide what portion of the model should be attended to. Which are the most relevant, important, urgent or insistent structures within the model? Portions of the model compete for attention. These competing portions take the form of coalitions of structures from the model. One such coalition wins the competition. The agent has decided on what to attend.

But, the purpose of all this processing is to help the agent to decide what to do next [12]. To this end, the winning coalition passes to the global workspace, the namesake of Global Workspace Theory [5,6], from which it is broadcast globally. Though the contents of this conscious broadcast are available globally, the primary recipient is procedural memory, which stores templates of possible actions including their contexts and possible results [13]. It also stores an activation value for each such template that attempts to measure the likelihood of an action taken within its context producing the expected result. Templates whose contexts intersect sufficiently with the contents of the conscious broadcast instantiate copies of themselves with their variables specified to the current situation. These instantiations are passed to the action selection mechanism, which chooses a single action from these instantiations and those remaining from previous cycles. The chosen action then goes to sensory-motor memory, where it picks up the appropriate algorithm by which it is then executed. The action so taken affects the environment, and the cycle is complete.

The LIDA model hypothesizes that all human cognitive processing is via a continuing iteration of such cognitive cycles. These cycles occur asynchronously, with each cognitive cycle taking roughly 200 ms. The cycles cascade, that is, several cycles may have different processes running simultaneously in parallel. Such cascading must, however, respect the serial order of consciousness in order to maintain the stable, coherent image of the world with which consciousness endows us [14, 15]. This cascading, together with the asynchrony, allows a rate of cycling in humans of five to ten cycles per second. A cognitive "moment" is quite short! There is considerable empirical evidence from neuroscience studies suggestive of such cognitive cycling in humans [16, 17, 18, 19, 20 21]. None of this evidence is conclusive.

2. Theory of Mind in LIDA Agent

Procedural memory in LIDA is implemented as a scheme net [22] of templates called schemes. Information about how another cognitive agent does something is stored in the scheme net. Representations of other agents in perceptual memory and of their actions allow them to be recognized (activated) when that other agent is perceived. They remain in the workspace for a few 10's of seconds or a few hundred cognitive cycles. Structures may form as part of the updated internal model and win the competition for consciousness, which will cause them to be stored in episodic memory. This allows LIDA to reconstruct them after they decay from the workspace. This process creates a kind of virtual reality in the workspace, and allows LIDA to model other agents and predict what they are likely to do in a given situation. This ability to build internal structures predicting the actions of other agents is a high-level adaptation to the environment, which provides feedback on these predictions. In LIDA, schemes and perceptual categories used to make successful predictions have their activations increased. The opposite is true for components involved in unsuccessful predictions.

The ability to represent and model other cognitive agents is known as a theory of mind. It helps explain some aspects of human cognition such as one-shot learning by observing a teacher and the ability of people to read subtle social queues. There is a theory that the mirror neurons in the cortex are involved in this process [23, 24]. Other animals are also known to exhibit theory of mind [25].

There are two mechanisms for implementing the theory of mind in LIDA. Perceptual memory has concept nodes (representations) for both the self and for other agents. Thus, if an action were observed in a given context, similar percepts would be produced whether the cognitive agent itself or another agent took the action. The most significant difference would be that one percept would contain an instantiation of the "self" node, whereas the other would contain a representation for the concept of the other agent. The second mechanism is a set of codelets, small independently running programs, and schemes from procedural memory that contain slots that can bind to any agent allowing LIDA to perceive and reason about actions by other agents in the same way as it does about its own perceptions and actions.

These two mechanisms, combined with the learning procedures in LIDA, allow humans and, perhaps to some extent, other primates, to form complex representations of other agents. Simply put, a procedural learning mechanism converts a series of percepts containing another agent's concept node into a stream of schemes where the other agent's node is converted to a slot that can bind to any similar agent, including the self. This highlights one difference between learning in humans and in some lower animals, which must be trained by repeatedly rewarding the desired behavior until it is learned. This latter type of learning is similar to traditional statistical machine learning techniques, in contrast to human-like learning, represented in cognitive models that take into account research in psychology, neuroscience and cognitive science [22].

3. Example

Thinking about other peoples' cognitive processes involves using partial knowledge about what is actually true, and ideas about how the mind works, to create a narrative about what happened (or would happen), and what a given individual would do in a particular situation. Some aspects of other minds are, or appear to be, perceived

directly. Others require deliberation. An example of the former would be a “gut” feeling about whether a person is telling the truth based on their demeanor. An example of the latter would be reasoning about a person’s beliefs and motives.

A person’s demeanor is related to their facial expressions, body language, tone of voice, etc. Information about a person’s state of mind may be detected through the mirror neuron system, which reacts to subtle social signals whether expressed by the person themselves or observed by the person in others [26]. These signals accompany primal emotions such as anger, fear, or guilt and are difficult to mask [27]. Recognition of such signals has survival value at the individual level, though warning of threats, and at the group level, by increasing trust through the ability to detect false allegiances.

There is no evidence that the mirror system is used in “higher-level” tasks such as reasoning about another person’s beliefs. There is evidence that people possess a relatively common set of beliefs about how minds work [28]. For example, there is an over-emphasis on logic that people use to explain both their own actions and those of others. This can be referred to as “naïve psychology” in analogy to the “naïve physics” that people use in reasoning about the physical world. Both sets of beliefs contain heuristics that work well in most cases but are shown to be false in others. A famous example from naïve physics is that “heavier objects fall faster than lighter ones.” A similar example from naïve psychology is that “people act in their own self interest.”

In order to avoid the complexities of the theory of mind in humans, we will use as an example a primate experiment that showed evidence of it in monkeys.

3.1. Theory of Mind in *Macaca Mulatta* (*Rhesus Monkeys*)

A recent animal experiment [25] has shown evidence for the theory of mind in monkeys. In this section we describe the experiments and show how the LIDA model represents this process in a natural and realistic manner. By using experimental data from higher animals instead of humans, we can simplify the description while keeping the essential elements.

The experimenters approached a monkey with two containers, a “noisy” container with bells attached to the lid and body, and a “silent” container without the bells. A grape was placed in each container. The action of placing the grape caused noise for the first container but not the second. The experimenters then backed away from the containers but remained in view. In one set of experiments, the experimenters faced the containers and in another set, they faced the ground. Data were collected when the monkey took the grape from one of the containers. When the experimenters faced the ground, the monkey preferentially chose the silent container. When the experimenters faced the containers, the monkey chose randomly.

The researchers concluded that these data are evidence that the monkeys had a mental model of the human experimenters, who were considered potential rivals, i.e. competitors for the food. If the humans were not looking the containers, the monkey could get the food without alerting them by choosing the silent container. If the humans were looking at the containers, they would know of the monkey’s actions in either case, so it wouldn’t matter which container was chosen.

The cognitive states and actions of the monkey will now be illustrated using the LIDA model. Figures 2 to 4 show a simplified version of the hypothetical perceptual memory components needed to recognize the events in the experiments. As shown in the figures, there are four root concepts, actions, relationships, goals, and objects.

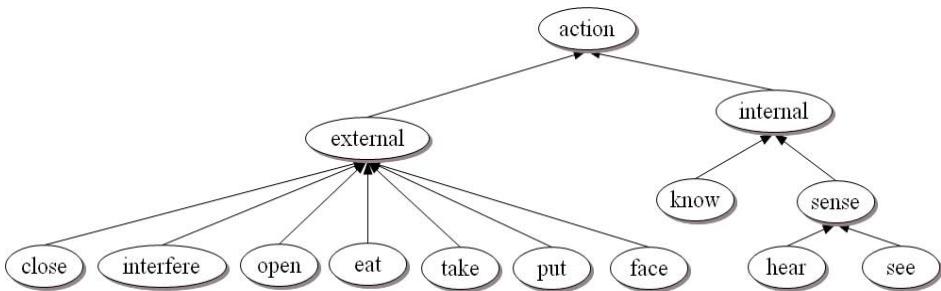


Figure 2. Perceptual Memory Needed for Experiment (with *a-kind-of* links)

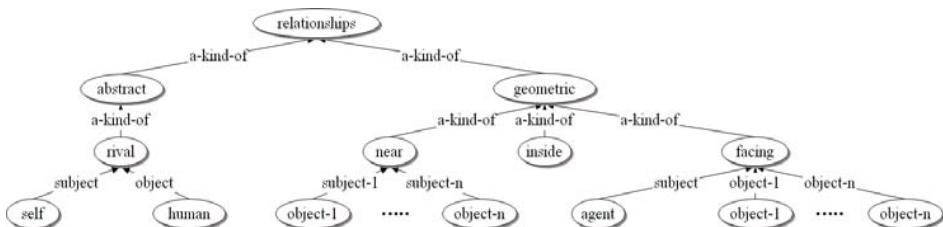


Figure 3. Perceptual Memory Relationships Needed for the Experiment

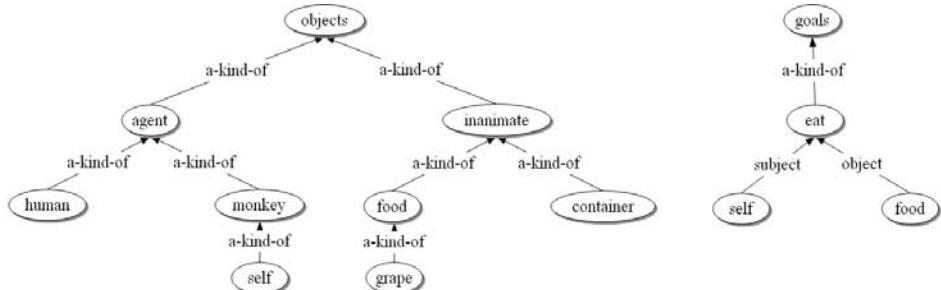


Figure 4. Perceptual Memory Objects and Goals Needed for the Experiment

Perceptual memory, and the other conceptual structures described in this paper, will be represented as semantic nets, that is, directed graphs with labels on both the nodes and links [29]. The graphs are simplified to trees by repeating certain nodes. For example, the “eat” node under “external action” is the same as the “eat” node under “goals.”

When the experimenter approaches the monkey with the containers and grapes, certain structures will be excited in perceptual memory and instantiated in the workspace as a percept. This is shown in figure 5, which simply contains a representation of the objects perceived in the scene. When the experimenter puts the grape in the noisy container, the perceptual structures in figure 6 are added.

One advantage of the representation in the figures is that it can be translated into natural language. For example, the perceptual structure on the upper left in Figure 6 could be translated to: The person put a grape in container1, causing it to make noise.

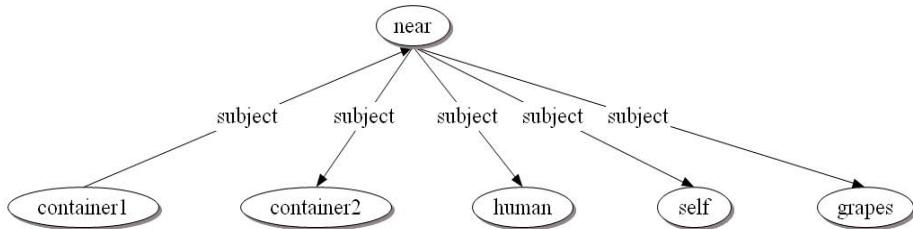


Figure 5. Initial Percept in Experiment

In order to avoid a proliferation of diagrams, only the natural language description will be used in some cases.

When the experimenter puts a grape in container 2, similar perceptual structures appear in the workspace except that there is no noise because container 2 doesn't have bells attached to it. These percepts translate to: The person put a grape in container2 and the person closed container2.

Structure Building Codelets act upon the elements in the workspace. These codelets are simple computational processes that scan the workspace for elements that match class variables. In this paper we will use class names beginning with a question mark. The codelets also contain probable results, also represented as graph structures. The results can combine the triggering elements and/or include a new structure.

The activation of the codelet depends on the likelihood of the result, given the trigger, which includes elements of the context implicit in the matched workspace structures. This likelihood based activation is constantly adjusted by feedback from the environment. The activation of the new structure depends on both the activation of the codelet and that of its triggering elements. Other structure building codelets can combine the new structures to build larger structures. This allows a reasonable number of codelets to build complex models. Attention Codelets are triggered by structures in the workspace. These codelets and the structures that triggered them form coalitions in the model. The coalitions compete for consciousness based their activation. The winner goes to consciousness and may eventually trigger actions by the agent.

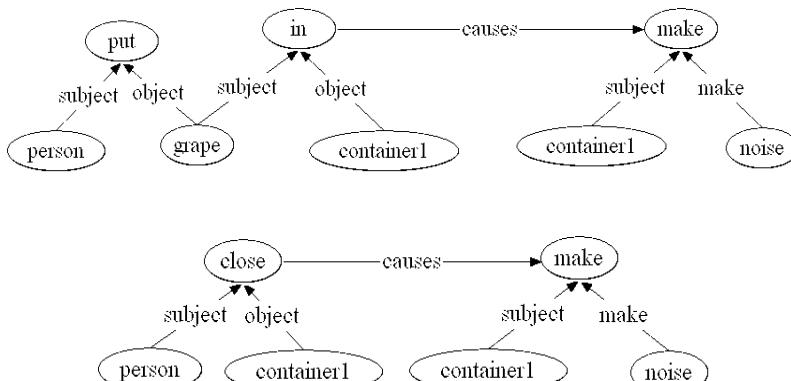


Figure 6. Percepts for Placing a Grape in the Noisy Container

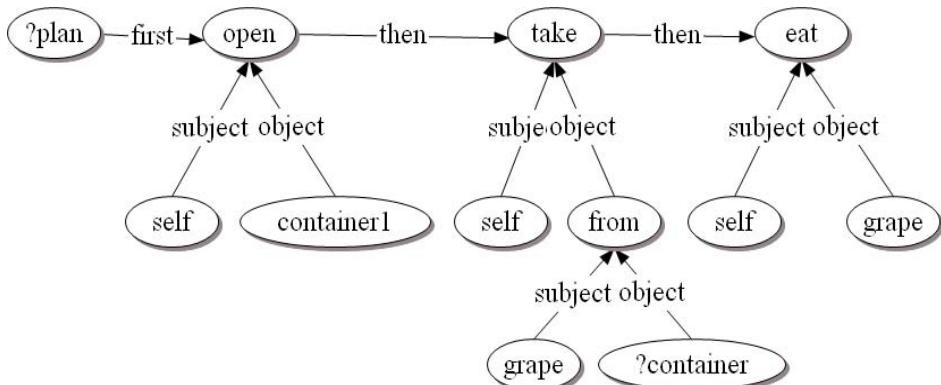


Figure 8. Alternative Plans: a) ?plan = plan1, ?container = container1, and b) ?plan = plan2, ?container = container2

The monkey sees the grapes, which are a kind of food. If the monkey is hungry or likes grapes, this perception may activate a goal structure in the workspace to eat the grape, as shown in Figure 7. Goal structures from perceptual memory are only potential goals. If this structure makes it to “consciousness,” it will be broadcast to all components of the model, including Procedural Memory, where it could activate schemes with results that may achieve the goal.

Instances of the activated schemes will be incorporated in the behavior net [30, 31]. This will excite other components of the net, where the goals can be converted into behaviors to achieve them through the activation of streams of behavior codelets. In this case, the higher-level behavior is likely to result in volitional decision making to construct a plan to obtain and eat the grape. Its actions are internal and include activation of codelets to search, write to, and build structures in the workspace.

The monkey’s mind goes through a planning process to construct a causally linked structure from perceptual structures in the workspace to the goal structure [6, 32, 33]. There are choices on how to achieve the goal, whether to take the grape from container 1 or container 2. The resulting plans are shown in Figure 8.

According to ideomotor theory [34, 5], a cognitive agent will consider one plan of action at a time. In the LIDA implementation of this theory [33], once the consideration begins, a timer is set. If no objection to the plan comes to mind before the timer ends, the plan will be performed. If there is an objection, the timer is reset and an alternative plan is considered.

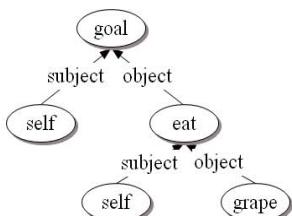


Figure 7. The Monkey’s Goal is to Eat a Grape

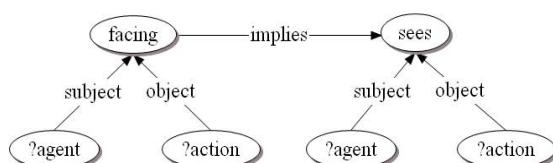


Figure 9. If an agent is Facing an Action, the Agent is Likely to See the Action

In this experiment, any objections to the monkey's plans would likely concern possible interference by the human, which the monkey considers a rival. The monkey reasons about what the human would know about potential actions the monkey could take. This is the essence of the Theory of Mind. In the experiment, it can be accomplished with the following structures (Figures 9 to 12).

If the monkey considers plan two first, no objection is created and it will go to consciousness (unless some coalition with higher activation wins the competition). It will then be broadcast to procedural memory and activate schemes that are sent to the behavior net. The resulting behavior stream will instantiate a series of external behavior codelets enabling the monkey to physically open the silent container and take the grape.

If the monkey considers plan one first, however, an objection will be raised. This is due to a structure related to the concept of a rival (Figure 12): "?agent1 is a rival of ?agent2 causes ?agent1 to interfere with the plans of ?agent2 if they are known to ?agent1." At this point, the monkey abandons plan one and considers plan two. Since there are no objections, the plan will be carried out. This explains the experimental results for the case where the human faces the ground.

In the case when the experimenter is facing the scene, the monkey's plan will be known whether or not it chooses the noisy box. This is a cognitive dilemma in the sense that the monkey could consider the first plan, find it blocked, go the second plan, find that it is also blocked, then go back to the first plan, resulting in an oscillation. There are three mechanisms for breaking the deadlock [33]. First, the plans for eating the grape compete with other potential activities represented in the workspace. The winning coalition makes it to consciousness and is acted upon. In the experiment, for example, there were cases when another monkey drove the subject from the scene and both plans were abandoned.

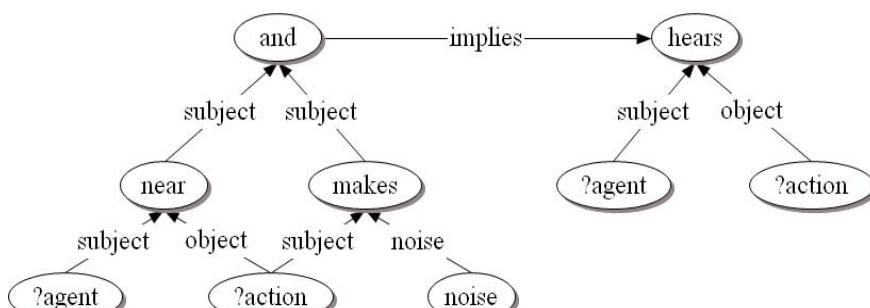


Figure 10. If an Agent is near an Action, and the Action makes a Noise, the Agent will Hear the Action

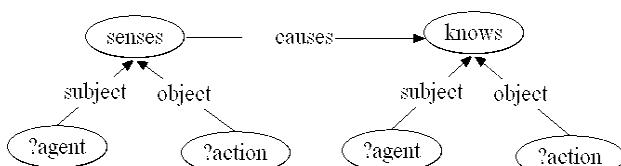


Figure 11. If an Agent Senses an Action, the Agent Knows the Action (has occurred)

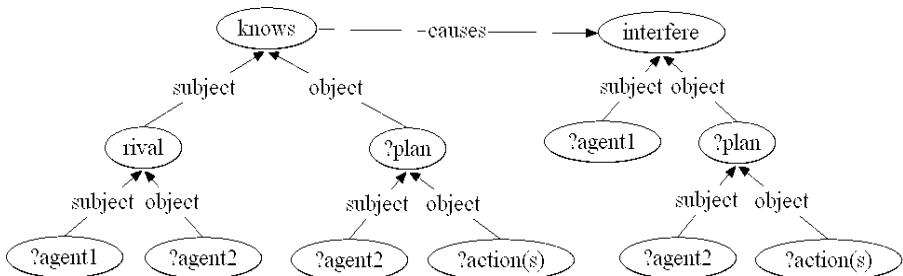


Figure 12. If Agent1 is a Rival of Agent2, and Agent1 knows of Planned Actions by Agent2, Agent1 will Interfere with Agent2's Actions

Second, every time the same plan is reconsidered, the length of time on the timer is shortened so that it could end before the objection has time to form. Finally, if the other mechanisms do not force a decision, a metacognitive process will be triggered to accept whichever plan is under consideration. This multiplicity of tie-breaking mechanisms illustrates the importance of decisiveness for survival in higher animals and results in a random choice. It explains the case where the experimenter is facing the scene.

4. Discussion, Conclusions and Future Work

The LIDA model can explain the results of the primate experiment in a psychologically realistic way. It offers an explanation of the theory of mind in monkeys in the sense that the actions depending on where the experimenters were facing were derived from a representation of the experimenter's mental state and the monkey's belief in their potential to interfere with the monkey's plans.

LIDA also accounts for the results that were not analyzed in the experiment, the times when the monkey abandoned the scene without trying to retrieve either grape. In LIDA, this is a result of losing the competition for consciousness. LIDA's cognitive cycle is based on human cognition and is consistent with the older *Sense-Plan-Act* cycle, but contains many more cognitive functions. LIDA's perceptions of the ongoing activities of itself and other agents create percepts in the workspace that trigger codelets for building additional structures, taking actions, and other cognitive functions that result in the model's interpretation of the current situation.

The basic representations in LIDA for implementing the theory of mind are found in both perceptual and procedural memories. A percept involving a given action and context can contain either the "self" node or the node for another agent, depending on who performed it. Schemes in procedural memory can contain slots that can be bound to any agent, the self or another. This is more general than what is known about the mirror neuron system, which responds strongly only to basic actions such as eating or grasping objects with the hands [35].

While a detailed explanation of all the types of learning and adaptation in LIDA is beyond the scope of this paper, LIDA's ability for self-organization results from: a large number of simple behaviors and primitive features that can be combined in arbitrary ways; feedback from the environment; decay of learned concepts and procedures, including the ability to forget; and both competitive and cooperative learning, i.e., competition between coalitions of cognitive structures.

Uncertainty plays a role in LIDA's reasoning through the base activation of its *behavior codelets*, which depend on the model's estimated probability of the codelet's success if triggered. LIDA observes the results of its behaviors and updates the base activation of the responsible codelets dynamically.

It avoids combinatorial explosions by combining reasoning via association [36] with reasoning via deduction. One can create an analogy between LIDA's workspace structures and codelets and a logic-based architecture's assertions and functions. However, LIDA's codelets only operate on the structures that are active in the workspace during any given cycle. This includes recent perceptions, their closest matches in other types of memory, and structures recently created by other codelets. The results with the highest estimate of success, i.e. *activation*, will then be selected. No attempt is made to find all possible solutions or reason until either a solution is found it is shown that none exists. If reasoning takes too long, time-keeping mechanisms such as the ones described above will force termination. The disadvantage is that incorrect conclusions are possible and potential solutions to problems can be overlooked, similar to the way in which human cognition works.

One would expect higher level cognition to be more sophisticated in humans than in other primates and perhaps lacking in some lower mammals and other animals. The primate experiment was selected for this paper because of the simplicity of the situation involving evidence for the theory of mind. Its implementation in LIDA provides a mechanism to explain "one-shot" learning by observing a teacher.

In this paper, the model is being tested only qualitatively, showing that it explains the behaviors exhibited by the monkeys. Further research will involve enhancing the existing LIDA implementation so that the experiment can be simulated and quantitatively confirm the predicted results, or not.

References

- [1] Premack, D. G. and G. Woodruff. 1978. Does the chimpanzee have a theory of mind? *Behavioral and Brain Sciences* 1: 515-526.
- [2] Osterberg, G. 1935. Topography of the layer of rods and cones in the human retina. *Acta Ophthalm. suppl.* 6, no. 1-103.
- [3] Tse, P. U. (2003). If vision is 'veridical hallucination', what keeps it veridical? Commentary (p. 426-427) on Gestalt isomorphism and the primacy of subjective conscious experience: a Gestalt Bubble model, by Steven Lehar. *Behavioral and Brain Sciences*, 26(4):375-408.
- [4] Franklin, Stan. 2007. A foundational architecture for artificial general intelligence. In *Advances in artificial general intelligence: Concepts, architectures and algorithms*, proceedings of the AGI workshop 2006, ed. Ben Goertzel and Pei Wang:36-54. Amsterdam: IOS Press.
- [5] Baars, Bernard J. 1988. *A cognitive theory of consciousness*. Cambridge: Cambridge University Press.
- [6] Baars, Bernard. 1997. In the theatre of consciousness. Global workspace theory, a rigorous scientific theory of consciousness. *Journal of Consciousness Studies* 4: 292–309.
- [7] Franklin, S., and A. C. Graesser. 1997. Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. In *Intelligent Agents III*. Berlin: Springer Verlag.
- [8] Baars, B. J., and S. Franklin. 2003. How conscious experience and working memory interact. *Trends in Cognitive Science* 7:166-172.
- [9] Franklin, S., B. J. Baars, U. Ramamurthy, and M. Ventura. 2005. The Role of Consciousness in Memory. *Brains, Minds and Media* 1:1-38, pdf.
- [10] Franklin, S. and F. G. Patterson, Jr. (2006). The Lida Architecture: Adding New Modes of Learning to an Intelligent, Autonomous, Software Agent. *Integrated Design and Process Technology*, IDPT-2006, San Diego, CA, Society for Design and Process Science.
- [11] Ramamurthy, U., Baars, B., D'Mello, S. K., & Franklin, S. (2006). LIDA: A Working Model of Cognition. *Proceedings of the 7th International Conference on Cognitive Modeling*. Eds: Danilo Fum, Fabio Del Missier and Andrea Stocco; pp 244-249. Edizioni Goliardiche, Trieste, Italy.

- [12] Franklin, Stan. 1995. Artificial minds. Cambridge MA: MIT Press.
- [13] D'Mello, Sidney K, U Ramamurthy, D'Mello, A Negatu, and S Franklin. 2006. A procedural learning mechanism for novel skill acquisition. In Proceeding of adaptation in artificial and biological systems, aisb'06, ed. Tim Kovacs and James A R Marshall, 1:184–185. Bristol, England: Society for the Study of Artificial Intelligence and the Simulation of Behaviour.
- [14] Merker, Bjorn. 2005. The liabilities of mobility: A selection pressure for the transition to consciousness in animal evolution. Consciousness and Cognition 14: 89–114.
- [15] Franklin, Stan. 2005. Evolutionary pressures and a stable world for animals and robots: A commentary on merker. Consciousness and Cognition 14: 115–118.
- [16] Lehmann D, Strik WK, Henggeler B, Koenig T, and Koukkou M. 1998. Brain electric microstates and momentary conscious mind states as building blocks of spontaneous thinking: I. Visual imagery and abstract thoughts. Int. J Psychophysiology 29, no. 1: 1-11.
- [17] Massimini, M., F. Ferrarelli, R. Huber, S. K. Esser, H. Singh, and G. Tononi. 2005. Breakdown of Cortical Effective Connectivity During Sleep. Science 309:2228-2232.
- [18] Sigman, M., and S. Dehaene. 2006. Dynamics of the Central Bottleneck: Dual-Task and Task Uncertainty. PLoS Biol. 4.
- [19] Uchida, N., A. Kepecs, and Z. F. Mainen. 2006. Seeing at a glance, smelling in a whiff: rapid forms of perceptual decision making. Nature Reviews Neuroscience 7:485-491.
- [20] Willis, J., and A. Todorov. 2006. First Impressions: Making Up Your Mind After a 100-Ms Exposure to a Face. Psychological Science 17:592-599.
- [21] Melloni, L., Molina, C., Pens, P., Torres, D., Singer, W., and Rodriguez, E. 2007. Synchronization of Neural Activity across Cortical Areas Correlates with Conscious Perception. J. Neuroscience, 27(11): 2858-2865.
- [22] D'Mello, Sidney K, S Franklin, U Ramamurthy, and B J Baars. 2006. A cognitive science based machine learning architecture. In AAAI 2006 Spring Symposium Series Sponsor: American Association for Artificial Intelligence. Stanford University, Palo Alto, California, USA.
- [23] Frey, Scott H and Valerie E. Gerry. 2006. Modulation of neural activity during observational learning of actions and their sequential orders. Journal of Neuroscience 26: 13194-13201.
- [24] Buckner, Randy L. and Daniel C. Carroll. 2007. Self-projection and the brain. Trends in Cognitive Sciences 11, no. 2: 49-57.
- [25] Santos, Laurie R, Aaron G Nissen, and Jonathan A Ferrugia. 2006. Rhesus monkeys, macaca mulatta, know what others can and cannot hear. Animal Behaviour 71: 1175–1181.
- [26] Arbib, M.A. and G. Rizzolatti. 1997. Neural expectations: A possible evolutionary path from manual skills to language. Communication and Cognition 29: 393-423.
- [27] Ekman, P. 1993. Facial expression of emotion. American Psychologist 48: 384-392.
- [28] Saxe, Rebecca. 2005. Against simulation: the argument from error. TRENDS in Cognitive Sciences Vol. 9 No. 4, pp 174-179.
- [29] Franklin, Stan. 2005. Perceptual memory and learning: Recognizing, categorizing, and relating. In Symposium on Developmental Robotics: American Association for Artificial Intelligence (AAAI). Stanford University, Palo Alto CA, USA.
- [30] Maes, P. 1989. How to do the right thing. Connection Science 1:291-323.
- [31] Negatu, Aregahegn and Stan Franklin. 2002. An action selection mechanism for 'conscious' software agents. Cognitive Science Quarterly 2, no. special issue on "Desires, goals, intentions, and values: Computational architectures." Guest editors Maria Miceli and Cristiano Castelfranchi.: 363–386.
- [32] Hofstadter, D. R. & the Fluid Analogies Research Group (1995). Fluid concepts and creative analogies. New York: Basic Books.
- [33] Franklin, Stan. 2000. Deliberation and voluntary action in 'conscious' software agents. Neural Network World 10: 505–521.
- [34] James, W. 1890. The Principles of Psychology. Cambridge, MA: Harvard University Press.
- [35] Gallese, V., Fadiga, L., Fogassi, L., Rizzolatti, G.. Action recognition in the premotor cortex. Brain (1996), 119, 593-609.
- [36] Franklin, S. 2005. Cognitive robots: Perceptual associative memory and learning. In Proceedings of the 14th annual international workshop on robot and human interactive communication (Ro-Man 2005):427-433.

How Might Probabilistic Reasoning Emerge from the Brain?

Ben GOERTZEL, Cassio PENNACHIN

Novamente LLC

Abstract: A series of hypotheses is proposed, connecting neural structures and dynamics with the formal structures and processes of probabilistic logic. First, a hypothetical connection is proposed between Hebbian learning in the brain and the operation of probabilistic term logic deduction. It is argued that neural assemblies could serve the role of logical terms; synapse-bundles joining neural assemblies could serve the role of first-order term-logic statements; and in this framework, Hebbian learning at the synaptic level would be expected to have the implicit consequence of probabilistic deduction at the logical statement level. A conceptual problem arises with this idea, pertaining to the brain's lack of a mechanism of "inference trails" as used to avoid circular reasoning in AI term logic inference systems; but it is explained how this problem may be circumvented if one posits an appropriate inference control mechanism. Finally, a brief discussion is given regarding the potential extension of this approach to handle more complex logical expressions involving variables – via the hypothesis of special neural structures mapping neural weights into neural inputs, hence implementing "higher order functions."

1. Introduction

Neuroscience cannot yet tell us how the brain generates abstract cognition [1]. However, the current body of experimental and theoretical knowledge does contain some tantalizing hints; and among these is a striking formal parallel between Hebbian learning as observed to occur in the brain [2,3] and certain inference rules and formulae existing in the theory of probabilistic logical inference [4-6]. In this paper we explore this parallel in moderate detail, with a view toward exploring what it may teach us about both complex neurodynamics and probabilistic inference. The result is a series of specific hypotheses pertaining to the manner in which probabilistic logical inference, at multiple levels of abstraction and sophistication, might emerge from complex neurodynamics. As an implicit side-effect, these ideas clarify the potential parallels between brain structures/dynamics and contemporary AGI architectures that incorporate uncertain term logic, such as the NARS system [7] and the Novamente Cognition Engine (NCE) [8] with its Probabilistic Logic Networks (PLN) component.

We believe this is an important direction to be investigating, given the numerous recent successes of probabilistic methods in various domains of AI research, e.g. [9-14]. An increasing segment of the AI community is convinced that probabilistic methods are an essential aspect of intelligence; and if this is so, it behooves us to try to understand the potential neural underpinnings of probabilistic inference. Of course, the human mind is not always accurate in making probabilistic estimations [15], but this

does not imply that we aren't in many cases carrying out such estimations in some sense. An exploration of the manners in which probabilistic reasoning system may display inference pathologies similar to human beings is provided in [4] and would take us too far afield here. To be sure, though, we are not claiming the human brain to be a highly accurate probabilistic-calculation engine. Rather we are claiming that probabilistic inference may be a reasonable, approximate high-level model of certain structures and dynamics that emerge from brain activity.

First we articulate certain provisional assumptions regarding neural knowledge representation. These assumptions are not radical and essentially constitute a specific version of the classical Hebbian proposal that knowledge is neurally stored in cell assemblies [2], incorporating Susan Greenfield's emendation that "core" cell assemblies are opportunistically expanded into larger transient assemblies when they come into attentional focus [16,17]. The Hebb/Greenfield approach suggests that logical terms may be neurally implemented as cell assemblies; and we build on this by suggesting that first-order term logic statements (representing conditional probabilities) may be neurally implemented as bundles of synapses between assemblies. This suggestion is harmonious with recent work on Bayesian inference as a property of neuronal populations [18,19]. All this leads up to the quite natural hypothesis that Hebbian learning on the neuronal level gives rise to dynamics on the cell-assembly level that are roughly equivalent to first-order probabilistic term logic.

This appealing parallel runs into an apparent snag when one considers the "trail" mechanism utilized in modern term-logic-based AI systems to avoid pathologies of circular inference [4,5,7]. However, we present here arguments and simulation results indicating that this may not be a significant issue for probabilistic term logic systems with appropriate control mechanisms, and hence may not be an obstacle to the conceptual linkage between neurobiology and logic conjectured here.

In the final section we make some hypotheses regarding the extension of this perspective to handle more complex term logic expressions involving variables and quantifiers. The essential idea proposed is that, if one adopts a logical formalism involving combinators rather than explicit variables, then the problem of mapping complex logic expressions into neural structures and dynamics boils down to the problem of neurally representing what functional programmers call "higher-order functions" [20]. We suggest one possible neural architecture via which higher-order functions could be implemented, involving a novel "router" substructure that encodes the connection-weights of one neural subnetwork as inputs to another neural subnetwork.

The overall meta-point we intend to make with these hypotheses is that there is not necessarily any significant "conceptual gap" between brain structures/dynamics and probabilistic logic, or uncertain logic more generally. The hypothesis that the brain's dynamics give rise to emergent-level uncertain logic seems plausible and consistent with current knowledge in both AI and neuroscience. Due to limitations in the current experimental toolset, we don't yet know enough to pinpoint exactly how this emergence happens, but we can make plausible hypotheses which can guide ongoing research, both AI and biological.

2. Provisional Assumptions Regarding Neural Knowledge Representation and Neurodynamics

Before addressing reasoning, we must first deal briefly with two issues regarding the brain: knowledge representation and learning. No one really knows how the brain represents complex knowledge, nor how it learns, but there are some fairly widely-respected high-level hypotheses, and for the purpose of the present paper, we will adopt the ones that seem to us most likely. Due to space considerations we do not review basic neurobiology and cognitive neuroscience here but assume the reader possesses the requisite textbook knowledge [1].

2.1. Knowledge Representation via Cell Assemblies

The hypothesis we'll adopt regarding knowledge representation, put simply, is that a distinct mental concept is represented in the brain as either:

- a set of “cell assemblies”, where each assembly is a network of neurons that are interlinked in such a way as to fire in a (perhaps nonlinearly) synchronized manner
- a distinct temporal activation pattern, which may occur in any one (or more) of a particular set of cell assemblies

The general definition of a “mental concept” is a bit subtle (see Goertzel, 2006) but a simple example (which will serve adequately for the present discussion) is a category of perceptual stimuli to which the organism systematically reacts in different ways. Also, although we will focus mainly on declarative knowledge here, we note that the same basic representational ideas can be applied to procedural and episodic knowledge: these may be hypothesized to correspond to temporal activation patterns as characterized above.

In the biology literature, perhaps the best-articulated modern theories championing the cell assembly view are those of Gunther Palm [21,22] and Susan Greenfield [16, 17]. Palm focuses on the dynamics of the formation and interaction assemblies of cortical columns. Greenfield argues that each concept has a core cell assembly, and that when the concept rises to the focus of attention, it recruits a number of other neurons beyond its core characteristic assembly into a “transient ensemble.”¹

It's worth noting that there may be multiple redundant assemblies representing the same concept – and potentially recruiting similar transient assemblies when highly activated. The importance of repeated, slightly varied copies of the same subnetwork has been emphasized by Edelman [23] among other neural theorists.

¹ The larger an ensemble is, she suggests, the more vivid it is as a conscious experience; an hypothesis that accords well with the hypothesis made in [24] that a more informationally intense pattern corresponds to a more intensely conscious quale -- but we don't need to digress extensively onto matters of consciousness for the present purposes.

2.2. Hebbian Learning

Regarding learning, our main assumption is that synapses in the brain are adapted via some variant of Hebbian learning. The original Hebbian learning rule, proposed by Donald Hebb in his 1949 book [2], was roughly

1. The weight of the synapse $x \rightarrow y$ increases if x and y fire at roughly the same time
2. The weight of the synapse $x \rightarrow y$ decreases if x fires at a certain time but y does not

Over the years since Hebb's original proposal, many neurobiologists have sought evidence that the brain actually uses such a method. What they have found, so far, is a lot of evidence for the following learning rule [25,26]

1. The weight of the synapse $x \rightarrow y$ increases if x fires shortly before y does
3. The weight of the synapse $x \rightarrow y$ decreases if x fires shortly after y does

The new thing here, not foreseen by Donald Hebb, is the “postsynaptic depression” involved in rule component 2.

Now, the simple rule stated above does not sum up all the research recently done on Hebbian-type learning mechanisms in the brain. The real biological story underlying these approximate rules is quite complex, involving many particulars to do with various neurotransmitters. Ill-understood details aside, however, there is an increasing body of evidence that not only does this sort of learning occur in the brain, but it leads to distributed experience-based neural modification: that is, one instance synaptic modification causes another instance of synaptic modification, which causes another, and so forth² [27].

2.3. Virtual Synapses and Hebbian Learning Between Assemblies

Hebbian learning is conventionally formulated in terms of individual neurons, but, it can be extended naturally to assemblies via defining “virtual synapses” between assemblies.

Since assemblies are sets of neurons, one can view a synapse as linking two assemblies if it links two neurons, each of which is in one of the assemblies. One can then view two assemblies as being linked by a bundle of synapses. We can define the weight of the synaptic bundle from assembly A1 to assembly A2 as the number w so that (*the change in the mean activation of A2 that occurs at time t+epsilon*) is on average closest to $w^*(\text{the amount of energy flowing through the bundle from A1 to A2 at time } t)$. So when A1 sends an amount x of energy along the synaptic bundle pointing from A1 to A2, then A2's mean activation is on average incremented/decremented by an amount $w*x$.

In a similar way, one can define the weight of a bundle of synapses between a certain static or temporal activation-pattern P1 in assembly A1, and another static or temporal activation-pattern P2 in assembly A2. Namely, this may be defined as the

² This has been observed in “model systems” consisting of neurons extracted from a brain and hooked together in a laboratory setting and monitored; measurement of such dynamics *in vivo* is obviously more difficult.

number w so that (*the amount of energy flowing through the bundle from A1 to A2 at time t*) $*w$ best approximates (*the probability that P2 is present in A2 at time t+epsilon*), when averaged over all times t during which P1 is present in A1.

It is not hard to see that Hebbian learning on real synapses between neurons implies Hebbian learning on these virtual synapses between cell assemblies and activation-patterns.

3. Probabilistic Term Logic as Emergent from Neural Structures and Dynamics

Our next step is to connect these neurological entities we have hypothesized (virtual synapses between assemblies and activation-patterns) to entities existing within mathematical term logic: inheritance and similarity relationships. This will allow us to connect Hebbian learning on virtual synapses to term logic inference on these relationships.

We assume the reader has a basic familiarity with uncertain term logic, as described e.g. in [5] and [11]. For space reasons, here we will only mention the uncertain term logic concepts that we use directly, and will make little effort to give adequate conceptual background.

In this section we discuss only “first order term logic,” which does not involve explicit variables or quantifiers, and may be described as the logic of inheritance relationships. We’ll turn to higher-order term logic a little later. Roughly speaking, the semantics of the term logic relationship “A inherits from B” or $A \rightarrow B$, is that when B is present, A is also present. The truth value of the relationship measures the percentage of the times that B is present, that A is also present. “A is similar to B” or $A \sim B$, is a symmetrical version, whose truth value measures the percentage of the times that either one is present, that both are present.

A great deal of subtlety emerges when one examines the semantics of inheritance in detail [4,11]. Many subtypes of the basic inheritance relation arise, dealing with such distinctions as intension vs. extension and group vs. individual. Also, similarity relations may be treated as symmetrized inheritance relations. Here we will ignore these various aspects and assume a simplistic approach to inheritance, considering $A \rightarrow B$ to be basically interpretable as $P(B|A)$. Other aspects of inheritance may be treated according to variations of the ideas given here, but this is not explored here due to space considerations. There are two separate theories of uncertain term logic currently available: NARS, which is nonprobabilistic; and PLN, which is probabilistic. The main ideas presented here are valid only for PLN as they depend on probabilistic semantics. However, to the extent that NARS approximates PLN they are also of course applicable to NARS.

Figure 1 shows the basic inference rules of first-order term logic [4-6,11]. Each of these comes with a truth value formula, which determines the truth value of the conclusion link based on the truth values of the premise links. In PLN, these formulas are grounded in probability theory; e.g. the deduction formula is based on a heuristic probabilistic independence assumption. The NARS deduction formula closely approximates the PLN deduction formula; but the same is not true for induction and abduction.

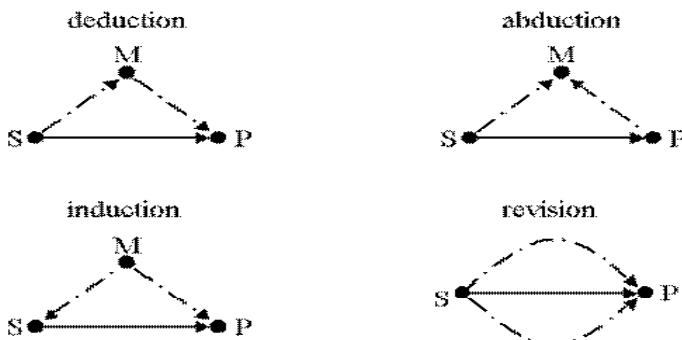


Figure 1. The four primary inference rules in first-order term logic

How can this be tied in with the brain? Suppose we have two assemblies A1 and A2, and these are activated in the brain when the organism is presented with stimuli in category C1 and C2 respectively (to take the simplest case of concepts, i.e. perceptual categories). Then, we may say that there is a *neural inheritance* $A1 \rightarrow A2$, whose probabilistic strength³ is the number w so that *(probability of A1's mean activation being greater than T at time t)*w* best approximates *(probability of A2's mean activation being greater than T at time t+epsilon)* for an appropriate choice of epsilon and T. This weight w, intuitively, represents the conditional probability $P(A2 \text{ is active} | A1 \text{ is active})$.

In a similar way, if we have two assembly activation patterns P1 and P2, which are defined as specific types of activity patterns occurring in A1 and A2 respectively, and which correspond to categories C1 and C2 respectively, we can define a neural inheritance $P1 \rightarrow P2$, whose probabilistic truth value is the number w so that *(P1 is present in A1 at time t)*w* best approximates *(P2 is present in A2 at time t+epsilon)*, on average over various times t, and assuming a threshold T used to determine when a pattern is present in an assembly.

It is immediate that, if there is a virtual synapse between A1 and A2 or P1 and P2, there will also be a neural inheritance there. Furthermore there will be a monotone increasing algebraic relationship between the weight of the virtual synapse and the probability attached to the neural inheritance. Inhibitory virtual synapses will correspond to very low link probabilities; excitatory virtual synapses will correspond to high link probabilities.

However, we can have neural inheritance without any virtual synapse. This is a key point, as it lets us distinguish neural inheritance relationships that are *explicit* (that correspond to virtual synapses) from those that are *implicit* (that do not). And this leads us directly to probabilistic reasoning, which is about transforming implicit inheritance relationships into explicit ones. The fundamental inference rules of term logic, as described above, create new inheritance links from old ones. The conclusions are implicit in the premises but until the inference is done, they may not be explicitly

³ As uncertain term logic systems often use multiple-component truth values, the term “strength” is commonly used to denote the primary component, which in this case is a probability value.

contained in the knowledge base of the system doing the reasoning. Probabilistic reasoning in the brain, we suggest, is all about translating implicit neural inheritance relationships into explicit ones.

In the PLN approach to term logic (but not the NARS approach), the four forms of inference shown in Figure 1 can actually be reduced to three forms: revision (which in its simplest form is weighted-averaging), deduction, and inversion (which reverses the direction of a link, and in probabilistic term logic is essentially Bayes rule). Let us elaborate what these mean in terms of cell assemblies.

Suppose that A1 and A2 are two assemblies, and there is a neural inheritance between A1 and A2. Then, there will also be a neural inheritance between A2 and A1, with a truth value given by Bayes rule. And according to Hebbian learning if there is a virtual synapse $A1 \rightarrow A2$, there is likely to grow a virtual synapse $A2 \rightarrow A1$. And according to the approximate correlation between virtual synapse weight and neural inheritance probability, this new virtual synapse from $A2 \rightarrow A1$ will have a weight corresponding to a probability approximating the one corresponding to the weight of $A1 \rightarrow A2$.

Similarly, suppose that A1, A2 and A3 are three assemblies. Then, if we have virtual synapses between $A1 \rightarrow A2$ and $A2 \rightarrow A3$, Hebbian learning suggests that a virtual synapse will grow between $A1 \rightarrow A3$. And what will the probability of the neural inheritance corresponding to this virtual synapse be? On average, it will be the probability one obtains by assuming the probabilities associated with $A1 \rightarrow A2$ and $A2 \rightarrow A3$ are independent. But, this means that on average, the probability associated with $A1 \rightarrow A3$ will accord with the value produced by the PLN deduction formula, which embodies precisely this independence assumption. Here, we have an additional source of error beyond what exists in the Bayes rule case; but, in the mean, the desired correspondence does hold.

So, according to the above arguments – which admittedly have been intuitive rather than mathematically rigorous – it would seem that we can build term logic inference between concepts out of Hebbian learning between neurons, if we assume cell assembly based knowledge representation, via introducing the conceptual machinery of virtual synapses and neural inheritance.

3.1. Avoiding Issues with Circular Inference

When one develops the ideas from the previous section, connecting uncertain term logic inference with neurodynamics, in more detail, only one possible snag arises. Existing computational frameworks for uncertain term logic inference utilize special mechanisms for controlling circular inference, and these mechanisms have no plausible neurological analogues. In this section we explore this issue and argue that it's not necessarily a big deal. In essence, our argument is that these biologically unnatural circularity-avoidance mechanisms are unnecessary in a probabilistic term logic system whose operations are guided by appropriate adaptive attention-allocation mechanisms. It's only when operating probabilistic term logic inference in isolation, in a manner that's unnatural for a resource-constrained intelligent system, that these circular-inference issues become severe.

We note however that this conclusion seems to be specific to probabilistic term logic, and doesn't seem to hold for NARS term logic, in which the circular inference problem may be more severe, and may in fact require a trail mechanism more strongly. We have not investigated this issue carefully.

To understand the circular inference problem, look at the triangles in Figure 1. It's easy to see that by performing deduction, induction and abduction in sequence, we can go around and around an inference triangle forever, combining the links in different orders, inferring each link in the triangle from the two others in different orders over and over again. What often happens when you do this in a computer program performing uncertain term logic inference, however, is that after long enough the inference errors compound, and the truth values descend into nonsense. The solution taken in the NARS and PLN uncertain term logic inference engines is something called *inference trails*. Basically, each inheritance link maintains a trail, which is a list of the nodes and links used as premises in inferences determining its truth value. And a rule is put in place that the link L should not be used to adjust the truth value of the link M if M is in L's trail.

Trails work fine for computer programs implementing uncertain term logic, though managing them properly does involve various complexities. But, from the point of view of the brain, trails seem quite unacceptable. It would seem implausible to hypothesize that the brain somehow stores a trail along with each virtual synapse. The brain must have some other method of avoiding circular inferences leading to truth value noisification.

In the context of the Novamente Cognition Engine (NCE) [8], an integrative AI system into which we have integrated the PLN probabilistic term logic engine, we have run a number of experiments with trail-free probabilistic inference. The first of these involved doing inferences on millions of nodes and links (with nodes representing words and links derived via word co-occurrence probabilities across a text corpus). What we found was that, in practice, the severity of the circular-inference problem depended on the inference control strategy. When one implemented a strategy in which the amount of attention devoted to inference about a link L was proportional to an estimate of the amount of information recently gained by doing inference about L, then one did not run into particularly bad problems with circular inference. On the other hand, if one operated with a small number of nodes and links and repeatedly ran the same inferences over and over again on them, one did sometimes run into problems with truth value degeneration, in which the term logic formulas would cause link strengths to spuriously converge to 1 or 0.

To better understand the nature of these phenomena, we ran computer simulations of small networks of NCE nodes interlinked with NCE links indicating inheritance relationships, according to the following idea:

1. Each node is assumed to denote a certain perceptual category
2. For simplicity, we assume an environment in which the probability distribution of co-occurrences between items in the different categories is stationary over the time period of the inference under study
3. We assume the collection of nodes and links has its probabilistic strengths updated periodically, according to some "inference" process
4. We assume that the results of the inference process in Step 3 and the results of incorporating new data from the environment (Step 2) are merged together ongoingly via a weighted-averaging belief-revision process

In our simulations Step 3 was carried out via executions of PLN deduction and inversion inference rules. The results of these simulations were encouraging: most of the time, the strengths of the nodes and links, after a while, settled into a "fixed point"

configuration not too distant from the actual probabilistic relationships implicit in the initial data. The final configuration was rarely equivalent to the initial configuration, but, it was usually close.

For instance one experiment involved 1000 random “inference triangles” involving 3 links, where the nodes were defined to correspond to random subsets of a fixed finite set (so that inheritance probabilities were defined simply in terms of set intersection). Given the specific definition of the random subsets, the mean strength of each of the three inheritance relationships across all the experiments was about .3. The Euclidean distance between the 3-vector of the final (fixed point) link strengths and the 3-vector of the initial link strengths was roughly .075. So the deviation from the true probabilities caused by iterated inference was not very large. Qualitatively similar results were obtained with larger networks.

The key to these experiments is the revision in Step 4: It is assumed that, as iterated inference proceeds, information about the true probabilities is continually merged into the results of inference. If not for this, Step 3 on its own, repeatedly iterated, would lead to noise amplification and increasingly meaningless results. But in a realistic inference context, one would never simply repeat Step 3 on its own. Rather, one would carry out inference on a node or link only when there was new information about that node or link (directly leading to a strength update), or when some new information about other nodes/links indirectly led to inference about that node-link. With enough new information coming in, an inference system has no time to carry out repeated, useless cycles of inference on the same nodes/links – there are always more interesting things to assign resources to. And the ongoing mixing-in of new information about the true strengths with the results of iterated inference prevents the pathologies of circular inference, without the need for a trail mechanism.

What we see from these various experiments is that if one uses an inference control mechanism that avoids the repeated conduction of inference steps in the absence of infusion of new data, issues with circular inference are not severe, and trails are not necessary to achieve reasonable node and link strengths via iterated inference. Circular inference can occur without great harm, so long as one only does it when relevant new data is coming in, or when there is evidence that it is generating information. This is not to say that trail mechanisms are useless in computational systems – they provide an interesting and sometimes important additional layer of protection against circular inference pathologies. But in an inference system that is integrated with an appropriate control mechanism they are not required. The errors induced by circular inference, in practice, may be smaller than many other errors involved in realistic inference. For instance, in the mapping between the brain and uncertain term logic proposed above, we have relied upon a fairly imprecise proportionality between virtual synapse weight and neural inheritance. We are not attempting to argue that the brain implements precise probabilistic inference, but only an imprecise analogue. Circular inference pathologies are probably not the greatest source of imprecision.

4. How Might More Complex Logical Statements and Inferences Be Expressed in the Brain?

The material of the previous sections comprises a speculative but conceptually coherent connection between brain structures and dynamics on the one hand, and

probabilistic logic structures and dynamics on the other. However, everything we have discussed so far deals only with first-order term logic, i.e. the logic of inheritance relationships between terms. Extension to handle similarity relationships, intensional inheritance and so forth is straightforward – but what about more complex term logic constructs, such as would conventionally be expressed using variables and quantifiers. In this section we seek to address this shortcoming, via proposing a hypothesis as to how probabilistic term logic in its full generality might be grounded in neural operations. This material is even more speculative than the above ideas, yet *something* of this nature is critically necessary for completing the conceptual picture.

The handling of quantifiers, in itself, is not the hard part. In [6] it is shown that, in a term logic framework, if one can handle probabilistic variable-bearing expressions and functions, then one can handle quantifiers attached to the variables therein. So the essence of the problem is how to handle variables and functions. And we suggest that, when one investigates the issue in detail, a relatively simple hypothesis emerges clearly as essentially the only plausible explanation, if one adopts the neural assembly theory as a working foundational assumption.

In the existing body of mathematical logic and theoretical computer science, there are two main approaches to handling higher-order expressions: variable-based, or combinator-based [20]. It seems highly implausible, to us, that the human brain is implementing some sort of intricate variable-management scheme on the neural-assembly level. Lambda-calculus and other formal schemes for manipulating variables, appear to us to require a sort of complexity and precision that self-organizing neural networks are ill-suited to produce via their complex dynamics. Of course it is possible to engineer neural nets that do lambda calculus (as neural nets are Turing-complete, as are biologically realistic models of cortical-column-assembly-based computation [28]), but this sort of neural-net structure seems unlikely to emerge via evolution, and unlikely to get created via known epigenetic processes.

But what about combinators? Here, it seems to us, things are a bit more promising. Combinators are higher-order functions; functions that map functions into functions; functions that map {functions that map functions into functions} into {functions that map functions into functions}, and so forth. There are specific sets of combinators that are known to give rise to universal computational capability; indeed, there are *many* such specific sets, and one approach to the implementation of functional programming languages is to craft an appropriate set of combinators that combines universality with tractability (the latter meaning, basically, that the combinators have relatively simple definitions; and that pragmatically useful logic expressions tend to have compact representations in terms of the given set of combinators).

We lack the neurodynamic knowledge to say, at this point, that any particular set of combinators seems likely to map into brain function. However, we may still explore the fundamental neural functionalities that would be necessary to give rise to a combinatory-logic-style foundation for abstract neural computation. Essentially, what is needed is the capability to supply one neural assembly as an *input* to another. Note that what we are talking about here is quite different from the standard notion of chaining together neural assemblies, so that the output of assembly A becomes the input of assembly B. Rather, what we are talking about is that *assembly A itself* – as a mapping from inputs to outputs – is fed as an input to assembly B. In this case we may call B a higher-order neural assembly.

Of course, there are numerous possible mechanisms via which higher-order neural assemblies could be implemented in the brain. Here we will discuss just one.

Consider a neural assembly A1 with certain input neurons, certain output neurons, and certain internal “hidden layer” neurons. Then, suppose there exists a “router” neural assembly X, which is at the receiving end of connections from many neurons in A1, including input, output and hidden neurons. Suppose X is similarly connected to many other neural assemblies: A2, A3, ... and so forth; and suppose X contains a “control switch” input that tells it which of these assemblies to pay attention (so, for instance, if the control input is set to 3, then X receives information about A3). When X is paying attention to a certain assembly, it routes the information it gets from that assembly to its outputs. (Going further, we may even posit a complex control switch, that accepts more involved commands; say, a command that directs the router to a set of k of its input assemblies, and also points it to small neural assembly implementing a combination function that tells it how to combine these k assemblies to produce a composite.)

Finally, suppose the input neurons of assembly B are connected to the router assembly X. Then, depending on how the router switch is set, B may be said to receive one of the assemblies Ak as input. And, next, suppose B’s output is directed to the control switch of the router. Then, in effect, B is mapping assemblies to assemblies, in the manner of a higher-order function. And of course, B itself is “just another neural assembly,” so that B itself may be routed by the router, allowing for assemblies that map {assemblies mapping assemblies} into assemblies, and so forth.

Where might this kind of “router” assembly exist in the brain? We don’t know, at the moment. Quite possibly, the brain may implement higher-order functions by some completely different mechanism. The point we want to make however, is that there are concrete possibilities via which the brain could implement higher-order logic according to combinatory-logic type mechanisms. Combinators might be neurally represented as neural assemblies interacting with a router assembly, as hypothesized above, and in this way the Hebbian logic mechanisms proposed in the previous sections could be manifested more abstractly, allowing the full-scope of logical reasoning to occur among neural assemblies, with uncertainty management mediated by Hebbian-type synaptic modification.

Conclusions

While our current level of knowledge about the brain does not support a rigorously grounded understanding of the emergence of abstract cognition from neural function, it is possible to draw interesting and highly evocative connections between the conceptual and formal models used to understand the brain, and the formal structures used in probabilistic logic to model and implement advanced aspects of cognition. From an AGI perspective, it is clear that, at this time, these connections do not provide sufficient guidance that it is possible to model an AGI architecture closely on the brain. However, we believe it is possible to create meaningful conceptual and formal mappings between AGI systems and neurobiological structures and dynamics, even when at first glance the AGI systems in question may involve completely non-biological-looking things such as logical inference engines. The mapping given here between uncertain term logic and neurobiology allows one to construct mappings between uncertain term logic based AI systems like NARS [7] and Novamente [8] and the human brain – allowing for instance a deepening of the parallels outlined in [29]

between the Novamente system and the human brain. This seems a fruitful ground for ongoing investigation.

References

- [1] Gazzaniga, Michael. *Cognitive Neuroscience*. Norton, 2002
- [2] Hebb, Donald. *The Organization of Behavior*. Wiley, 1949
- [3] Paulsen, O.; Sejnowski, T. J. (2000). "Natural patterns of activity and long-term synaptic plasticity". *Current opinion in neurobiology* 10 (2): 172-179
- [4] Ben Goertzel, Matthew Iklé, Izabela Goertzel, Ari Heljakka, Probabilistic Logic Networks, Berlin; New York, Springer Verlag, 2008, to appear
- [5] Matthew Iklé, Ben Goertzel, Izabela Goertzel, "Indefinite Probabilities for General Intelligence," *Advances in Artificial General Intelligence*, IOS Press, 2007.
- [6] Matthew Iklé, Ben Goertzel, "Quantifier Logic for General Intelligence," *Proceedings of AGI-08*, IOS Press, 2008.
- [7] Wang, Pei. *Rigid Flexibility: The Logic of Intelligence*. Springer-Verlag, 2006
- [8] Goertzel, Ben, Moshe Looks, Cassio Pennachin, "Novamente: An Integrative Architecture for Artificial General Intelligence," *Proceedings of AAAI Symposium on Achieving Human-Level Intelligence through Integrated Systems and Research*, Washington DC, August 2004.
- [9] Christopher D. Manning, Hinrich Schuetze, Foundations of Statistical Natural Language Processing, Cambridge, MA, MIT Press, 1999.
- [10] Sebastian Thrun, Wolfram Burgard and Dieter Fox, Probabilistic Robotics, Cambridge, MA, MIT Press, 2005.
- [11] John N. Holmes, Wendy J. Holmes, W.J. Holmes, Speech Synthesis and Recognition, Reading, UK, Taylor & Francis Ltd, 2002.
- [12] Judea Pearl, Probabilistic Reasoning in Intelligent Systems. San Mateo, Morgan Kaufmann, 1988.
- [13] Joseph Y. Halpern, Reasoning About Uncertainty, Cambridge, MA, MIT Press, 2003.
- [14] progic07: The Third Workshop on Combining Probability and Logic
- [15] Kahneman, Daniel, Paul Slovic and Amos Tversky. *Judgment Under Uncertainty*. Cambridge University Press, 1982
- [16] Subhjoyit Chakraborty1, 3, Anders Sandberg2 and Susan A. Greenfield/. Differential dynamics of transient neuronal assemblies in visual compared to auditory cortex, *Experimental Brain Research*, 1432-1106, 2007
- [17] Greenfield SA, Collins TF. A neuroscientific approach to consciousness. *Prog Brain Res*. 2005;150:11-23.
- [18] Amari, Shun-ichi, Si Wu. Neural Implementation of Bayesian Inference in Population Codes, *NIPS 2001*
- [19] Ma WJ, Beck JM, Latham PE, Pouget A Bayesian inference with probabilistic population codes. *Nat Neurosci* 2006 Nov 9(11):1432-8
- [20] Field, Anthony and Peter Harrison. *Functional Programming*. Addison-Wesley, 1097
- [21] Markert H, Knoblauch A, Palm G. Modelling of syntactical processing in the cortex. *Biosystems*. 2007 May-Jun;89(1-3):300-15. Epub 2006
- [22] Palm, Gunther Neural Assemblies. An Alternative Approach to Artificial Intelligence. Springer, Berlin, Heidelberg, New York, 1982
- [23] Edelman, Gerald. *Neural Darwinism*. Basic Books, 1988
- [24] Goertzel, Ben. *The Hidden Pattern*. BrownWalker, 2006
- [25] Roberts P.D., Bell C.C. ; Spike-timing dependent synaptic plasticity in biological systems. *Biological Cybernetics*, 87, 392-403 , 2002
- [26] Lisman J, Spruston N.; Postsynaptic depolarization requirements for LTP and LTD: a critique of spike timing-dependent plasticity. *Nature Neuroscience*' 8, 839-41, 2005
- [27] Bi, G-q, Poo, M-m . Synaptic modifications by correlated activity: Hebb's postulate revisited. *Ann Rev Neurosci* ; 24:139-166, 2001
- [28] Simen, P.; Polk, T.; Lewis, R.; Freedman, E. Universal computation by networks of model cortical columns. *Neural Networks*, 2003. *Proceedings of the International Joint Conference on Volume 1, Issue 1, 20-24 July 2003 Page(s): 230 - 235 vol.1*
- [29] Goertzel, Ben and Cassio Pennachin. Novamente: An Integrative Approach to Artificial General Intelligence. In *Artificial General Intelligence*, Springer-Verlag, 2006

An Integrative Methodology for Teaching Embodied Non-Linguistic Agents, Applied to Virtual Animals in Second Life

Ben GOERTZEL, Cassio PENNACHIN, Nil GEISSWEILLER, Moshe LOOKS,
Andre SENNA, Welter SILVA, Ari HELJAKKA, Carlos LOPES

Novamente LLC, Washington DC

Abstract. A teaching methodology called Imitative-Reinforcement-Corrective (IRC) learning is described, and proposed as a general approach for teaching embodied non-linguistic AGI systems. IRC may be used with a variety of different learning algorithms, but it is particularly easily described in EC lingo. In these terms, it is a framework for automatically learning a procedure that generates a desired type of behavior, in which: a set of exemplars of the target behavior-type are utilized for fitness estimation; reinforcement signals from a human teacher are used for fitness evaluation; and the execution of candidate procedures may be modified by the teacher via corrections delivered in real-time. An example application of IRC to teach behaviors to AI-controlled artificial animals embodied in the Second Life virtual world is described in detail, including a review of the overall virtual-animal-control software architecture and how the integrative teaching/learning methodology fits into it. In this example application architecture, the learning algorithm may be toggled between hillclimbing and probabilistic evolutionary learning. Envisioned future applications are also discussed, including an application to embodied language learning applicable to agents in Second Life and other virtual worlds.

Keywords. Reinforcement learning, imitative learning, corrective learning, evolutionary programming, hill-climbing, MOSES, intelligent virtual agents

1. Introduction

Supposing one intelligent agent (the “teacher”) has knowledge of how to carry out a certain behavior, and wants to transfer this knowledge to another intelligent agent (the “student”). But, suppose the student agent lacks the power of language (which might be, for example, because language is the thing being taught!). How may the knowledge be transferred? At least three methodologies are possible:

- **Imitative learning:** The teacher acts out the behavior, showing the student by example
- **Reinforcement learning:** The student tries to do the behavior himself, and the teacher gives him feedback on how well he did

- **Corrective learning:** As the student attempts the behavior, the teacher actively corrects (i.e. changes) the student's actions, guiding him toward correct performance

Obviously, these three forms of instruction are not exclusive. What we describe here, and call IRC learning, is a pragmatic methodology for instructing AGI systems that combines these three forms of instruction. We believe this combination is a potent one, and is certainly implicit in the way human beings typically teach young children and animals.

We present IRC learning here primarily in the context of virtually embodied AGI systems – i.e., AGI systems that control virtual agents living in virtual worlds. There is an obvious extension to physical robots living in the real world and capable of flexible interaction with humans. In principle, IRC learning is applicable more broadly as well, and could be explored in various non-embodied context such as (for instance) automated theorem-proving. In general, the term “IRC learning” may be used to describe any teacher/student interaction that involves a combination of reinforcement, imitation and correction. While we have focused in our practical work so far on the use of IRC to teach simple “animal-like” behaviors, the application that interests us more in the medium term is language instruction, and we will enlarge upon this a bit in the Conclusion.

In collaboration with The Electric Sheep Company, our software firm Novamente LLC is currently in the midst of creating a large-scale commercial implementation of IRC learning, as a methodology for teaching virtual animals in Second Life and other online virtual worlds. The virtual animals we are now experimenting with are nonlinguistic animals that can carry out spontaneous behaviors while seeking to achieve their own goals, and can also specifically be trained by human beings to carry out novel tricks and other behaviors (which were not programmed into them, but rather must be learned by the AI on the fly based on interaction with an avatar controlled by a human teacher). This current experimental work will be used as the basis of a commercial product to be launched sometime in 2008.

In Section 2 we will give a brief overview of our virtual-animal software architecture, and explain how the IRC methodology fits in, utilizing either hillclimbing or evolutionary learning, allied with simple inference, as the underlying learning engine (aspect 3 above). This software is work-in-progress and we don't yet have anywhere near a full understanding of what the strengths and limitations of the IRC methodology will be in this context, but it has already proved capable of learning some simple behaviors and we are confident it will prove considerably extensible. After describing this Second Life virtual animal application in detail, we then briefly review our plans for future R&D, which include a subsequent application to embodied language learning in Second Life and other virtual worlds.

We note that this is a compacted version of this paper, and a longer version is available on the AGI-2008 conference website.

1.1. IRC Learning in the Context of the Quest for Powerful AGI

One may decompose the overall task of creating a powerful AGI system into four aspects (which of course are not entirely distinct, but still are usefully distinguished):

1. **Cognitive architecture** (the overall design of an AGI system: what parts does it have, how do they connect to each other)
2. **Knowledge representation** (how does the system internally store declarative, procedural and episodic knowledge; and how does it create its own representation for knowledge of these sorts in new domains it encounters)
3. **Learning** (how does it learn new knowledge of the types mentioned above; and how does it learn how to learn, and so on)
4. **Teaching methodology** (how is it coupled with other systems so as to enable it to gain new knowledge about itself, the world and others)

This article focuses on the fourth of these aspects, presenting some ideas about AGI teaching methodology that we believe to have quite general significance, although they were developed in the context of a specific AGI system (the Novamente Cognition Engine) that is founded on specific commitments regarding the other three aspects. For the first author's views on the other three aspects of AGI, in the context of the NCE and more generally, the reader is directed to [1-3] and other prior publications. The focus on the fourth aspect in this paper should not be intended as a slight to the other three aspects: this is a brief paper and focuses narrowly on one part of the problem, but we mustn't forget that it's only one part of the problem. We need to build our AGI systems well, but we also need to teach them well, and it's important to understand fully and deeply exactly what that means. The research presented here has been conducted under the working hypothesis that, via constructing appropriately-architected AGI systems and then instructing them appropriately in virtual worlds, it may be possible to move from the current relatively primitive state of technology to an advanced level of AGI much more rapidly than the bulk of AI researchers believe.



Figure 1. Screenshot from Second Life, illustrating various behaviors of a Novamente-AI-controlled virtual dog. The dog chasing a cat illustrates spontaneous behavior driven by the dog's internal goals; the figure on the upper right illustrates single-avatar teaching (of soccer skills); the bottom figure illustrates two-avatar teaching (of frisbee skills).

1.2. The Power of Virtual Worlds for Accelerating Progress toward Powerful AGI

From an AI theory perspective, virtual worlds may be viewed as one possible way of providing AI systems with *embodiment*. The issue of the necessity for embodiment in AI is an old one, with great AI minds falling on both sides of the debate. The classic GOFAI systems (see [4] for a high-level review) are embodied only in a very limited sense; whereas [5] and others have argued for real-world robotic embodiment as the golden path to AGI. Our own view is somewhere in the middle: as outlined in [3] we suspect embodiment is very useful though probably not strictly necessary for AGI, and we also suspect that at the present time, it is probably more generally worthwhile for AI researchers to spend their time working with virtual embodiments in digital simulation worlds, rather than physical robots.

The notion of virtually embodied AI is nowhere near a new one, and can be traced back at least to Winograd's (1972) classic SHRDLU system. However, technology has advanced a long way since SHRDLU's day, and the power of virtual embodiment to assist AI is far greater in these days of Second Life, Word of Warcraft, HiPiHi, Creatures, Club Penguin and the like. The range of possibilities is both obvious and astounding; but to concretely understand the potential power of virtual embodiment for AGI, in this essay we'll focus on the virtual-animal product mentioned above.

2. Creating Virtual Animals for Second Life

Next we discuss some of the practical steps we are currently taking, aimed at gradually realizing the above-described vision. We briefly describe the software architecture we have developed for controlling virtual animals in Second Life – which for the sake of this discussion we will call the Virtual Animal Brain (VAB). This architecture contains some nontrivial AI within it, related to action selection and the overall relationship between goals, procedures and contexts. However, we describe it here primarily to provide context for the discussion in the following two sections, which deal with the combined use of imitative and reinforcement learning in the context of learning embodied behaviors via evolutionary program learning, hillclimbing, and associative memory.

The capabilities of our virtual animals, in their current form, include: spontaneous exploration of the environment; automated enactment of a set of simple predefined behaviors; efficient learning of another set of predefined behaviors; flexible trainability: i.e., (less efficient) learning of flexible behaviors invented by pet-owners on the fly; communication with the animals, for training of new behaviors and a few additional purposes, occurs in a special subset of English here called ACL (Animal Command Language); individuality: each animal has its own distinct personality; spontaneous learning of new behaviors, without need for explicit training

Our main focus here will be on the “flexible trainability” aspect, but we will also touch on other aspects as appropriate. And, though we won't stress this point, the same ideas discussed here in the context of teacher-focused learning, may also be used in a slightly modified form to enable spontaneous learning based on embodied experience gathered during self-directed world-exploration.

Beyond the above, some capabilities intended to be added in relatively-near-future VAB versions include: recognition of novel categories of objects, and integration of object recognition into learning; generalization based on prior learning, so as to be able

to transfer old tricks to new contexts; and use of computational linguistics (integrated into the Novamente Cognition Engine, which as described below underlies the VAB) to achieve a more flexible conversational facility. These will also be briefly discussed below. Note also that the VAB architecture described here is not particular to Second Life, but has been guided somewhat by the particular limitations of Second Life.

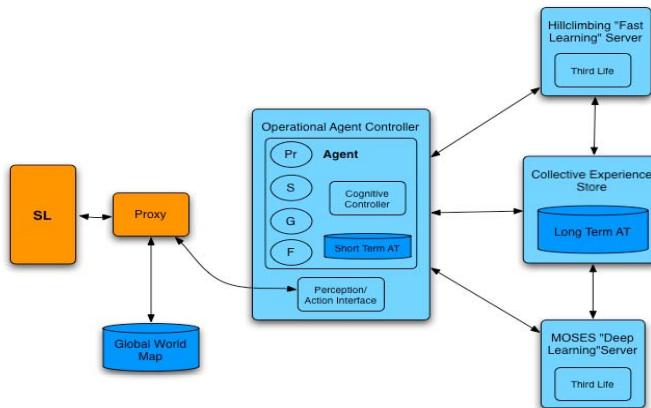


Figure 2. Initial Virtual Animal Brain software architecture

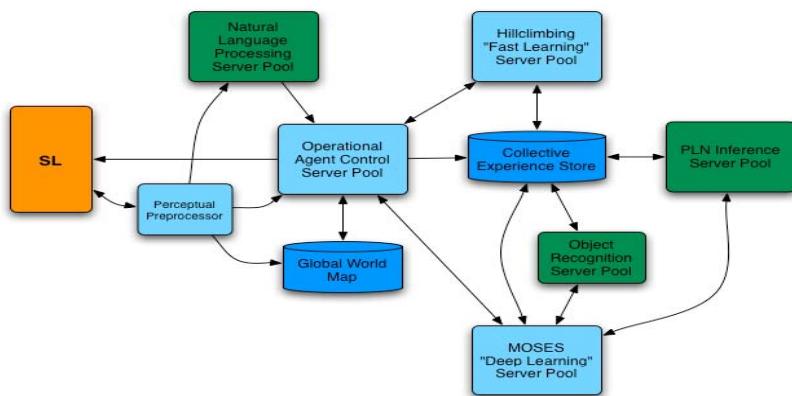


Figure 3. Next-Phase Virtual Animal Brain software architecture

(tentatively intended for implementation in 2008)

2.1. Software Architecture for Virtual Animal Control

Figure 2 shows the initial version of the VAB architecture, which has currently been implemented. Figure 3 shows the next version of the architecture, which has been designed in detail and if all goes well will be implemented during 2008.

All components of VAB but the Perceptual Pre-processor and the Global World Map will be specialized instances of the Novamente Cognition Engine, an existing C++ software system described in [1-3] which contains numerous functional components and has been used within two commercial applications (the Biomind ArrayGenius product for gene expression data analysis [7]; and the INLINK product for interactive natural language knowledge entry, founded on the RelEx semantic analysis engine [8]) and also within an R&D system that controls a humanoid agent learning simple behaviors in a 3D simulation world [9].

The learning servers will be described in the following section, as learning is the main focus of this paper. In the remainder of this section we will describe the other aspects of the architecture, which form the context in which the learning occurs.

The Global World Map is in its initial incarnation basically a 2D navigation mesh for Second Life, where each point in space is labeled with information about the agents that reside there. Some markup is included in the map to indicate 3-dimensional features of the environment. Future versions will involve extension into a full 3D navigation mesh.

First, the perceptual pre-processor is essentially a proxy that stands between SL and the rest of the architecture, translating the output of SL into an XML format that the VAB can understand. The current VAB system receives perceptions at a fairly high level – for instance, it observes a list of nearby objects, with metadata attached to them, and information about the spatial coordinates they occupy. Regarding avatars, it receives information regarding the animations the avatars are running at a given time (a more useful list for some avatars than others, as in some cases avatars may run animations for which our AI agent doesn't know the semantics). Examples of the perceptions emanating from the Perceptual Pre-processor are things like (using a notation in which \$ precedes variables):

- I am at world-coordinates \$W
- Object with metadata \$M is at world-coordinates \$W
- Part of object with metadata \$M is at world-coordinates \$W
- Avatar with metadata \$M is at world-coordinates \$W
- Avatar with metadata \$M is carrying out animation \$A
- Statements in Petaverse, from the pet owner

There is also proxy code that translates the actions and action-sequences generated by the VAB into instructions SL can understand (such as “launch thus-and-thus animation”). Due to the particularities of Second Life¹, the current VAB system carries out actions via executing pre-programmed high-level procedures, such as “move forward one step”, “bend over forward” and so forth. Example action commands are:

¹ The current Second Life API does not provide direct access to the skeletons underlying the characters executed in the world.

- Move (\$d , \$s) :\$d is a distance, \$s is a speed
- Turn (\$a, \$S) : \$a is an angle , \$S is a speed
- Jump (\$d, \$h, \$s) : \$h is a maximum height, at the center of the jump
- Say (\$T), \$T is text : for agents with linguistic capability, which is not enabled in the current version
- pick up(\$O) : \$O is an object

Next, in the agent control software component, each virtual animal is associated with its own “animal brain” software object, including among other things:

- An in-built (weighted) set of goals, and a set of “personality parameters” that guide various aspects of its behavior
- A package of basic information: Location, owner, current physical form, etc.
- A declarative memory, which contains e.g. associations between names of behaviors, and schemata (procedures) carrying out those behaviors
- A “procedural memory” containing procedures that may be useful to enact
- An “episodic memory” containing specific episodes the animal has been involved in, especially those that have led to reward from a teacher, or to achievement of one of the animal’s goals
- An attention allocation process, determining which schemata the pet carries out at each point in time. This process will vary its behavior based on the pet’s “personality” as specified by user-supplied parameters
- An “active procedural pool” containing procedures that are currently in the midst of being acted out

The declarative memory stores a reasonably long record of what all the pets have seen and done. Of course, when memory gets full, sufficiently old memories are saved to disk (or deleted, but it would be nice to have samples on disk for later mining)

One of the aspects of the system under current active development is the Collective Experience Store, which contains a process that scans the experience base and extracts interesting “unsupervised learning problems.” E.g. “Fred likes to feed pets. Let’s figure out what causes him to feed one pet but not another, and place that knowledge in the minds of those pets whose experiences are most useful in doing this figuring out.”

Next, another extremely critical part of the architecture: action selection. As Stan Franklin [10] has pointed out, ultimately intelligence has got to come down to action selection. An intelligent agent has got to decide what actions to take at what points in time – this is the way it goes about achieving its goals. In the VAB, action selection is controlled for each animal within the object allocated to that animal inside the agent control component.

As noted above, each animal has a set of goals, which are provided in advance. These are fairly basic things such as: don’t be too hungry or too full, don’t be too thirsty or too hydrated, seek social interaction, seek novelty, seek praise. Each of these goals has a certain numerical weight attached to it, and the relative weighting of the different goals in an individual animal constitutes an important part of that animal’s “personality.”

Each animal’s brain also contains a declarative knowledge base, containing information the animal has learned during its lifetime – who is its owner, if it has an owner; where it has found various items at various points of time in the past; who has been friendly versus evil to it; and so forth. The declarative knowledge base is an

AtomTable, to use NCE lingo; it contains weighted, typed nodes and links in the manner typically used within the Novamente Cognition Engine.

And each animal's brain also contains a procedural knowledge base, consisting of the set of behavior-generating procedures the animal has found useful in various contexts in the past. The declarative knowledge base contains relationships, in NCE node-and-link format, expressing the content "*Procedure P, in context C has led to the achievement of goal G.*" and associated with appropriate weightings including probabilistic truth values. For example, in the mind of an animal serving as a pet, there might be a link

```
Implication <[.8,.95],.95
AND
    Inheritance current_smell food
    Evaluation current_location my_home
    Execution goto my_food_bowl
    Evaluation am_hungry
    maintain_appropriate_fullness
```

expressing the fact that if the animal is hungry, and smells food, and is at home, then going to its food bowl may be a way of achieving its goal of maintaining appropriate fullness (not being too full or too hungry). In this example the actual procedure involved is a very simple one, simply the "goto" procedure that heads toward an object. (The $<[.8,.95],.95>$ is an uncertain truth value, representing in the NCE's "indefinite probability" format, see [11]). In general, the calculation of such truth values is not a difficult matter; the hard part is identifying what are the right contexts to use in conjunction with each procedure and each goal. In most cases, though, it turns out that the right contexts for virtual animals are relatively simple ones; this will obviously be more of a problem when one turns to applying a similar approach to virtual humanoid agents or other agents with more complex lives.

How does action selection work, then? Each procedure in the animal's mind is assigned an importance level, at each point in time, based on the degree to which it's estimated to imply the currently important goals given the currently relevant context. And note that the context here includes the other procedures that are executing at that point in time. Some procedures may take a while to execute – so the animal's brain must also maintain a list of the procedures that are currently in the middle of running. Then, a procedure is selected for execution, with the probability of its selection being proportional to its urgency.

Eventually, to make highly intelligent virtual agents, it will be necessary to do advanced probabilistic logical inference (as carried out e.g. by the NCE's PLN inference engine) on the fly in the course of action selection, in order to calculate importances based on the current context in a sufficiently flexible way. Another aspect of action selection not yet taken into account in our virtual animals is subgoaling. These issues lead to the necessity for a more complex system combining attention allocation and goals, as contained in the NCE design (and described in moderate detail in [1]).

Finally, Figure 3 shows some additional components intended for addition to the VAB, but not yet in place in the operational software. First, object recognition – by which we refer to the capability of a pet to look at an object in the SL world and determine what kind of object it is (a vehicle, a car, a truck, a hat, a shirt, a vest, etc.). This may be achieved via deployment of the MOSES algorithm as a supervised categorization engine. For the first version of the VAB, object recognition occurs only

insofar as objects are labeled with appropriate metadata indicating their type. Next, we would like to incorporate the full power of the NCE's Probabilistic Logic Networks reasoning engine [11] into the VAB, so as to enable more effective generalization within the Collective Experience Store. The current VAB utilizes some PLN inference rules within a simple control scheme, but this is not nearly as powerful as enabling general PLN backward and forward chaining inference on the animals' collective memory store. Ultimately, in fact, we would like to incorporate PLN into the real-time agent controller component of the system as well, so as to enable more intelligent and contextually appropriate action selection; but this is a bigger job than incorporating PLN into the CES, because it requires real-time control of advanced PLN inference. Finally, even though in general real non-human animals don't understand very much natural language, nevertheless it will be of considerable value to integrate a more robust NLP capability into the VUB in a later version. This is because the existing version of the Animal Command Language, while quite powerful in its capabilities, is still very brittle compared to natural languages. Of course, this also constitutes a step toward creating more linguistically ambitious animals as mentioned above, such as parrots that talk; and also toward creating humanoid avatars that communicate in English with a level of grounded understanding, rather than in the manner of chat-bots.

3. IRC Learning in the Virtual Animal Brain

Perhaps the best way to introduce the essential nature of the IRC teaching protocol is to give a brief snippet from a script that was created to guide the actual training of our Second Life virtual animals. This snippet involves only I and R; the C will be discussed afterwards.

This snippet demonstrates a teaching methodology that involves two avatars: Bob who is being the teacher, and Jill who is being an "imitation animal," showing the animal what to do by example.

1. *Bob wants to teach the dog Fido a trick. He calls his friend Jill over.
"Jill, can you help me teach Fido a trick?"*
2. *Jill comes over. "How much will you pay me for it?*
3. *Bob gives her a kiss.*
4. *"All right," says Jill, "what do you want to teach him?"*
5. *"Let's start with fetching stuff," replies Bob.*
6. *So Bob and Jill start teaching Fido to fetch using the Pet language....*
7. *Bob says: "Fido, I'm going to teach you to play fetch with Jill."*
8. *Fido sits attentively, looking at Bob.*
9. *Bob says: "OK, I'm playing fetch now."*
10. *Bob picks up a stick from the ground and throws it. Jill runs to get the stick and brings it back to Bob.*
11. *Bob says: "I'm done fetching.*
12. *Bob says, "You try it."*
13. *Bob throws a stick. Fido runs to the stick, gets it, and brings it back.*
14. *Bob says "Good dog!"*
15. *Fido looks happy.*
16. *Bob says: "Ok, we're done with that game of fetch.*
17. *Bob says, "Now, let's try playing fetch again."*

18. *This time, Bob throws a stick in a different direction, where there's already a stick lying on the ground (call the other stick Stick 2).*
19. *Fido runs and retrieves Stick 2. As soon as he picks it up, Bob says "No." But Fido keeps on running and brings the stick back to Bob.*
20. *Bob says "No, that was wrong. That was the wrong stick. Stop trying!"*
21. *Jill says, "Furry little moron!"*
22. *Bob says to Jill, "Have some patience, will you? Let's try again."*
23. *Fido is slowly wandering around, sniffing the ground.*
24. *Bob says "Fido, stay." Fido returns near Bob and sits.*
25. *Bob throws Stick 2. Fido starts to get up and Bob repeats "Fido, stay."*
26. *Bob goes and picks up Stick 1, and walks back to his original position.*
27. *Bob says "Fido, I'm playing fetch with Jill again."*
28. *Bob throws the first stick in the direction of stick 2.*
29. *Jill goes and gets stick 1 and brings it back to Bob.*
30. *Bob says "I'm done playing fetch with Jill."*
31. *Bob says "Try playing fetch with me now." He throws stick 1 in another direction, where stick 3 and stick 4 are lying on the ground, along with some other junk.*
32. *Fido runs and gets stick 1 and brings it back.*
33. *Bob and Jill both jump up and down smiling and say "Good dog! Good dog, Fido!! Good dog!!"*
34. *Fido smiles and jumps up and licks Jill on the face.*
35. *Bob says, "Fido, we're done practicing fetch."*

The text directed by Bob to Fido is in a limited dialect of English we call the ACL or Animal Command Language (which takes several different forms with varying levels of linguistic sophistication, but this point can be bypassed here, as the focus of this paper is not computational linguistics). Line 7 initiates a formal training session, and Line 33 terminates this session. The training session is broken into “exemplar” intervals during which exemplars are being given, and “trial” intervals during which the animal is trying to imitate the exemplars, following which it receives reinforcement on its success or otherwise. For instance line 9 initiates the presentation of an exemplar interval, and line 11 indicates the termination of this interval. Line 12 indicates the beginning of a trial interval, and line 16 indicates the termination of this interval.

The above example of combined imitative/reinforcement learning involves two teachers, but, this is of course not the only way things can be done. Jill could be eliminated from the above teaching example. The result of this would be that, in figuring out how to imitate the exemplars, Fido would have to figure out which of Bob’s actions were “teacher” actions and which were “simulated student” actions. This is not a particularly hard problem, but it’s harder than the case where Jill carries out all the simulated-student actions. So in the case of teaching fetch with only one teacher avatar, on average, more reinforcement trials will be required.

Another interesting twist on the imitative/reinforcement teaching methodology described above is the use of explicit correctional instructions from the teacher to the animal. This is not shown in the above example but represents an important addition to the methodology shown there. One good example of the use of corrections would be the problem of teaching a dog to sit and wait until the teacher says “Get Up,” using only a single teacher. Obviously, using two teachers, this is a

much easier problem. Using only one teacher, it's still easy, but involves a little more subtlety, and becomes much more tractable when corrections are allowed.

One way that human dog owners teach their dogs this sort of behavior is as follows:

- Tell the dog "sit"
- tell the dog "stay"
- Whenever the dog tries to get up, tell him "no" or "sit", and then he sits down again
- eventually, tell the dog to "get up"

The real dog understands, in its own way, that the "no" and "sit" commands said after the "stay" command are meta-commands rather than part of the "stay" behavior.

In our virtual-pet case, the easy way to do this is to give the Animal Command Language an explicit META flag. In this case, the teaching would look like

- tell the dog "I'm teaching you to stay"
- Tell the dog "META: sit"
- Whenever the dog tries to get up, tell him "META: no" or "META:sit", and then he sits down again
- eventually, tell the dog to "get up"
- tell the dog "I'm done teaching you to stay"--

Even without the META tag, this behavior is learnable via our learning algorithms within a modest number of reinforcement trials. But this well illustrates the give-and-take relationship between the sophistication of the teaching methodology and the number of reinforcement trials required. In many cases, the best way to reduce the number of reinforcement trials required to learn a behavior is not to increase the sophistication of the learning algorithm, but rather to increase the information provided during the instruction process. No matter how advanced the learning algorithm, if the teaching methodology only gives a small amount of information, it's going to take a bunch of reinforcement trials to go through the search space and find one of the right procedures satisfying the teacher's desires. One of the differences between the real-world learning that an animal or human child (or adult) experiences, and the learning "experienced" by standard machine-learning algorithms, is the richness and diversity of information that the real world teaching environment provides, beyond simple reinforcement signals. Virtual worlds provide a natural venue in which to experiment with providing this sort of richer feedback to AI learning systems.

4. The Cognitive Infrastructure Supporting IRC Learning in the Virtual Animal Brain

We now turn to the two pools of "learning servers" described in the above architecture diagram. These architectural components exist to carry out supervised or unsupervised learning, in order to learn new procedures for governing agent behavior, which may then be placed in the Agent Control Server Pool and associated there with the proper animal or set of animals. They constitute a specific cognitive infrastructure

implementing the IRC learning methodology in the virtual-animal context, with some extensibility beyond this context as well.

In the VAB, we have chosen to deploy two different learning algorithms, with different strengths and weaknesses. We have implemented a variety of hillclimbing, which is a fast learning algorithm but may fail on harder problems (in the sense of requiring an unacceptably large number of reinforcement trials). And we are currently in the midst of integrating MOSES, a sophisticated probabilistic evolutionary learning algorithm [12], as an alternative. Compared to hillclimbing, MOSES is much smarter but slower, and may take a few minutes to solve a problem. The two algorithms (as implemented for the VAB) share the same knowledge representation (a certain kind of C++ “program tree” used for representing procedures) and some other software components (e.g. normalization rules for placing procedures in an appropriate hierarchical normal form, as described in [12]).

The big challenge involved in designing the VAB system, AI-wise, is that these learning algorithms, used in a straightforward way with feedback from a human-controlled avatar as the fitness function, would need an excessive number of reinforcement trials to learn relatively simple behaviors. This would bore the human beings involved with teaching the animals. This is not a flaw of the particular learning algorithms being proposed, but is a generic problem that would exist with any AI algorithms. To choose an appropriate behavior out of the space of all possible behaviors satisfying reasonable constraints, requires more bits of information that is contained in a handful of reinforcement trials.

Most “animal training” games (e.g. Nintendogs may be considered as a reference case) work around this “hard problem” by not allowing teaching of novel behaviors. Instead, a behavior list is made up front by the game designers. The animals have preprogrammed procedures for carrying out the behaviors on the list. As training proceeds they make fewer errors, till after enough training they converge “miraculously” on the pre-programmed plan. This approach only works, however, if all the behaviors the animals will ever learn have been planned and scripted in advance.

The first key to making learning of non-pre-programmed behaviors work, without an excessive number of reinforcement trials, is in “fitness estimation” -- code that guesses the fitness of a candidate procedure at fulfilling the teacher’s definition of a certain behavior, without actually having to try out the procedure and see how it works. This is where the I part of IRC learning comes in. At an early stage in designing the VAB application, we realized it would be best if the animals were instructed via a methodology where the same behaviors are defined by the teacher both by demonstration *and* by reinforcement signals. The ACL language described above is designed to encourage this. Learning based on reinforcement signals only can also be handled, but learning will be slower.

In evolutionary programming lingo, we see that: procedures play the role of genotypes; demonstrated exemplars, and behaviors generated via procedures, play the role of phenotypes; and reinforcement signals from pet owner play the role of fitness.

One method of imitation-based fitness estimation used in the VAB involves an internal simulation world called Third Life (TL). TL can be visualized using a simple testing UI, but in the normal course of operations it doesn’t require a user interface; it is an internal simulation world, which allows the VAB to experiment and see what a certain procedure would be likely to do if enacted in the SL virtual world. Of course, the accuracy of this kind of simulation depends on the nature of the procedure. For procedures that solely involve moving around and interacting with inanimate objects, it

can be very effective. For procedures involving interaction with human-controlled avatars, other animals, or other complex objects, it may be unreliable – and making it even moderately reliable would require significant work that has not yet been done, in terms of endowing TL with realistic simulations of other agents and their internal motivational structures and so forth. Ultimately, for TL to work well would require an agent with a sophisticated Theory of Mind in the developmental-psychology sense (see [13] for a treatment of Piagetan developmental psychology in an AGI context). But short of this, TL has nonetheless proved useful for estimating the fitness of simple behavioral procedures.

When a procedure is enacted in TL, this produces an object called a “behavior description” (BD), which is represented in the NCE’s generic Atomspace (weighted labeled hypergraph) knowledge representation format. The BD generated by the procedure is then compared with the BD’s corresponding to the “exemplar” behaviors that the teacher has generated, and that the student is trying to emulate. Similarities are calculated, which is a fairly subtle matter that involves some heuristic inferences. An estimate of the likelihood that the procedure, if executed in SL, will generate a behavior adequately similar to the exemplar behaviors.

Furthermore, this process of estimation may be extended to make use of the animal’s long-term memory as collected in the CES component. Suppose a procedure P is being evaluated in the context of exemplar-set E. Then: 1) The experience base is mined for pairs (P' , E') that are similar to (P, E); 2) The fitness of these pairs (P', E') is gathered from the experience base; 3) An estimate of the fitness of (P, E) is then formed.

Of course, if a behavior description corresponding to P has been generated via TL, this may also be used in the similarity matching against long-term memory. The tricky part here, of course, is the similarity measurement itself, which can be handled via simple heuristics, but if taken sufficiently seriously becomes a complex problem of uncertain inference.

One thing to note here is that, although learning is done by each animal individually, this learning is subtly guided by collective knowledge within the fitness estimation process. Internally, we have a “borg mind” with multiple animal bodies, and an architecture designed to ensure the maintenance of unique personalities on the part of the individual animals in spite of the collective knowledge and learning underneath.

At time of writing, we have just begun to experiment with the learning system as described above, and are using it to learn simple behaviors such as playing fetch, basic soccer skills, doing specific dances as demonstrated by the teacher, and so forth. We still lack solid feel for the limitations of the methodology as currently implemented.

Note also that, going forwards, there is a possibility to use NCE’s PLN inference component to allow generalization of learned behaviors. For instance, with inference deployed appropriately, a pet that had learned how to play tag would afterwards have a relatively easy time learning to play “freeze tag.” A pet that had learned how to hunt for Easter eggs would have a relatively easy time learning to play hide-and-seek. Now, even the initial VAB will have some level of generalization ability in place, due to the use of the Collective Experience Store for fitness estimation. However, explicit use of inference will allow much more rapid and far-reaching inference capabilities.

Finally, how may corrections be utilized in the learning process we have described? Obviously, the corrected behavior description gets added into the knowledge base as an additional exemplar. And, the fact of the correction acts as a

partial reinforcement (up until the time of the correction, what the animal was doing was correct). But beyond this, what's necessary is to propagate the correction backward from the BD level to the procedure level. For instance, if the animal is supposed to be staying in one place, and it starts to get up but is corrected by the teacher (who says "sit" or physically pushes the animal back down), then the part of the behavior-generating procedure that directly generated the "sit" command needs to be "punished." How difficult this is to do, depends on how complex the procedure is. It may be as simple as providing a negative reinforcement to a specific "program tree node" within the procedure, thus disincentivizing future procedures generated by the procedure learning algorithm from containing this node. Or it may be more complex, requiring the solution of an inference problem of the form "Find a procedure P' that is as similar as possible to procedure P, but that does not generate the corrected behavior, but rather generates the behavior that the teacher wanted instead." This sort of "working backwards from the behavior description to the procedure" is never going to be perfect except in extremely simple cases, but it is an important part of learning. We have not yet experimented with this extensively in our virtual animals, but plan to do so as the project proceeds.

There is also an interesting variant of correction in which the agent's own memory serves implicitly as the teacher. That is, if a procedure generates a behavior that seems wrong based on the history of successful behavior descriptions for similar exemplars, then the system may suppress that particular behavior or replace it with another one that seems more appropriate – inference serving the role of a correcting teacher.

Finally, we note that a similar approach can be utilized for purely unsupervised learning, without any teacher involved. In that case, the animal's intrinsic goal system acts implicitly as a teacher. Experimentation with this sort of learning is on our list of virtual-animal R&D goals for 2008.

For instance, suppose the animal wants to learn how to better get itself fed. In this case, exemplars are provided by instances in the animal's history when it has successfully gotten itself fed. Reinforcement is provided by, when it is executing a certain procedure, whether or not it actually gets itself fed or not. Correction as such doesn't apply, but implicit correction may be used via deploying history-based inference. If a procedure generates a behavior that seems wrong based on the history of successful behavior descriptions for the goal of getting fed, then the system may suppress that particular behavior.

The only real added complexity here lies in identifying the exemplars. In surveying its own history, the animal must look at each previous instance in which it got fed (or sample thereof), and for each one recollect the series of N actions that it carried out prior to getting fed. It then must figure out how to set N – i.e. which of the actions prior to getting fed were part of the behavior that led up to getting fed, and which were just other things the animal happened to be doing a while before getting fed. To the extent that this exemplar mining problem can be solved adequately, innate-goal-directed spontaneous learning becomes closely analogous to teacher-driven learning as we've described it. Experience is an effective teacher.

Conclusion and Next Steps

We have described some recent work involving the use of the IRC teaching/learning methodology to instruct virtual animals in Second Life. This

constitutes a significant step beyond what is commonly done in virtual worlds and games regarding virtual-animal instruction; and a significant conceptual step beyond the pure reinforcement learning methodology that is commonly studied in the AI field. We have conducted some simple experiments using the IRC methodology in our Virtual Animal Brain already, but we still have a lot to learn about the best ways to pragmatically combine reinforcement, imitative and corrective learning in a virtual-world context. Along these lines, we have formulated a detailed roadmap for further research and development in the domain of virtual animal instruction. This includes a number of items mentioned above: object recognition, extension of the integrative methodology described above to spontaneous learning, and further integration of PLN inference to allow more sophisticated history-based fitness estimation and context-based action selection.

In the lengthier, online version of this paper, we take a bit of time to review the connections between this virtual-animal work and the larger AGI project of which it forms a component, and describe some of our medium-to-long-term plans for using IRC to enable a transition beyond nonlinguistic virtual animals, utilizing ideas drawn from Irene Pepperberg's [15] work teaching parrots English, and Michael Tomasello's [16] work on socially grounded language understanding.

References

- [1] Goertzel, Ben (2007). Virtual Easter Egg Hunting: A Thought-Experiment in Embodied Social Learning, Cognitive Process Integration, and the Dynamic Emergence of the Self. In Advances in artificial general intelligence, Ed. by Ben Goertzel and Pei Wang:36-54. Amsterdam: IOS Press.
- [2] Goertzel, Ben (2006). Patterns, Hypergraphs and General Intelligence. Proceedings of International Joint Conference on Neural Networks, IJCNN 2006, Vancouver CA
- [3] Goertzel, Ben (2006). The Hidden Pattern. BrownWalker Press
- [4] Crevier, Daniel (1993), AI: The Tumultuous Search for Artificial Intelligence, New York, NY: Basic Books
- [5] Brooks, Rodney (1999). Cambrian Intelligence. MIT Press.
- [6] Winograd, Terry (1972) . Understanding Natural Language. San Diego: Academic.
- [7] Goertzel, Ben, Cassio Pennachin, Lucio Coelho, Leonardo Shikida, Murilo Queiroz (2007). Biomind ArrayGenius and GeneGenius: Web Services Offering Microarray and SNP Data Analysis via Novel Machine Learning Methods In Proceedings of IAAI 2007, Vancouver CA, July 2007
- [8] Goertzel, Ben, Hugo Pinto, Ari Heljakka, Michael Ross, Izabela Goertzel, Cassio Pennachin. Using Dependency Parsing and Probabilistic Inference to Extract Gene/Protein Interactions Implicit in the Combination of Multiple Biomedical Research Abstracts, Proceedings of BioNLP-2006 Workshop at ACL-2006, New York
- [9] Heljakka, Ari, Ben Goertzel, Welter Silva, Izabela Goertzel and Cassio Pennachin (2006). Reinforcement Learning of Simple Behaviors in a Simulation World Using Probabilistic Logic, in Advances in Artificial General Intelligence, IOS Press.
- [10] Franklin, Stan (1995). Artificial Minds. MIT Press.
- [11] Ikle', Matt, Ben Goertzel, Izabela Goertzel and Ari Heljakka (2007). Indefinite Probabilities for General Intelligence, in Advances in Artificial General Intelligence, IOS Press.
- [12] Looks, Moshe (2006). Competent Program Evolution. PhD Thesis, Department of Computer Science, Washington University, St. Louis
- [13] Goertzel, Ben and Stephan Bugaj (2006). Stages of Cognitive Development in Uncertain-Logic-Based AI Systems. In Advances in artificial general intelligence, Ed. by Ben Goertzel and Pei Wang:36-54. Amsterdam: IOS Press.
- [14] Manning, Christopher and Heinrich Schütze (1999). Foundations of Statistical Natural Language Processing. MIT Press.
- [15] Pepperberg, Irene (2000). The Alex Studies. Harvard University Press.
- [16] Tomasello, Michael (2003). Constructing A Language. Harvard University Press.

VARIAC: an Autogenous Cognitive Architecture

J Storrs Hall

Storrmont: Laporte, PA 18626, USA

Abstract. Learning theory and programs to date are inductively bounded: they can be described as “wind-up toys” which can only learn the kinds of things that their designers envisioned. We conjecture [1] that general intelligence involves an unbounded learning ability. VARIAC is an experimental cognitive architecture designed to learn by modifying and extending itself, including its ability to learn, so that it can learn to become a better learner.

Keywords. autogeny, learning, cognitive architectures, automatic programming

Rationale

0.1. Automatic Programming

The approach to AI described herein is informed by the point of view that learning is programming: learning a category is designing an algorithm to distinguish members from non-members; learning a skill is writing a program to perform it; learning general knowledge about the world is building a model that can predict the consequences of situations or actions.

Thus any general learning program is a program that writes programs. It must invent both algorithms and representations. For human programmers, as programs become more complex it is advantageous to develop new notations to handle abstractions efficiently. This will hold true for AI systems as well. An unbounded learning system will be one which invents new programming languages.

Search, as exemplified by Solomonoff induction, is in theory capable of producing arbitrarily complex programs; but in practice the limitation of finite computational resources represents a formidable obstacle to the approach.

Automatic programming, a robust field from the beginnings of AI until the early 1980s, experienced a significant decline thereafter. The Stanford PSI project [2], which constructed LISP programs per a natural language dialogue with the user, appears to have been the high point of classical automatic programming. Lou Steinberg, a PSI principal, has conjectured that automatic programming is “AI complete” in the sense that it requires general knowledge and competence to understand what program the user wants without requiring him to specify it in the same detail he would have used in a programming language.¹

¹Louis Steinberg, personal communication with the author, April 2006

For a situated, experiential robot, however, automatic programming can obtain traction from another source: the system can attempt to write programs that model (and predict) phenomena, and ones that drive the robot in imitation of observed actions. If the robot already has programs that perform actions or model phenomena that are similar to the new ones, the problem is reduced to modifying an existing program – a considerably simpler task, and a technique universally used by both human programmers and evolution.

0.2. Self-improvement and “wind-up toys”

An AI capable of unbounded self-improvement must be able to invent new representations in which to think. At the base, any robot or organism has the representation of the world given by the raw signals from its sensors. It imposes an ontology on the world by interpreting these as more abstract concepts, in representations that make use of implicit notions of 3-dimensional space or rigid objects, for example.

In most robotic and AI systems to date, such an ontology is prepared for the system, in its entirety, by its human programmers. A few notable exceptions, such as AM and Eurisko, to the contrary notwithstanding, concept formation has been one of the most poorly developed of AI subfields. We believe a large part of this is due to overly simplistic notions of concept (such as predicates in FOPC). A more complete notion, including the requirement to implement the abilities to recognize, predict, and plan with the phenomenon involved, reveals the full scope of the problem and the challenge.

A robot which had induced (or been given) notions of 3D space and objects from sensor data might proceed to formulate kinematics, then dynamics with the concepts of force and acceleration, and ultimately laws of universal invariants such as the conservation of energy. Although fancifully ambitious, this progression demonstrates that an unbounded learning process must form concepts not only from patterns of sense data but of previously learned concepts.

1. Background

1.1. Computational Learning Theory

It is common in computational learning theory to represent situations or actions to be categorized by numeric vectors in some high-dimensional space. In the simplest case, the category can then be described by a hyperplane separating instances from non-instances. In the general case, however, the description of a category can be as complex as the description of an arbitrary subset of the set of possible descriptions of cases.

Consider for example a category of numeric vectors which includes just those having a prime number of 1 bits in their floating-point representations. It is difficult to imagine a geometric definition of this category, but one based on a program is straightforward. We conjecture that many real-world categories have structures as complex, if not as capricious, as this example: fractal by virtue of recursive description. (In CLT, the use of kernel functions, for example, represents a step in this direction.) In other words, we conjecture that many of the categories an AGI must learn will be best described by programs.

1.2. AI and Automatic Programming

Historically, AI has been the driving force behind many if not most of the advances in programming languages. LISP introduced abstract syntax trees (ASTs) as a basic datatype, and automatic storage management. It was also arguably the first functional programming language. PLANNER, PROLOG, and various theorem provers introduced automatic inference, now widely used as the mechanism behind the type systems of most modern languages. SNOBOL (designed for natural language processing) introduced pattern-matching. The “object-oriented” semantics of many modern languages can be traced back to “frame-based” AI systems of the 1970s.

LISP and PROLOG are prime languages for writing programs about programs for two main reasons. First is that they do have ASTs as primitive datatypes, together with a collection of operations on them of which needed functionality can be easily composed. Second is that their semantics are already reasonably abstract, so that neither programs written in them, nor programs written by those programs, need be concerned with details such as register usage, linkage conventions, or storage allocation.

Properly designed and implemented, high-level programming abstractions can reduce the size of programs by an order of magnitude as compared with the same programs in lower-level languages. But the effect is compounded in automatic programming programs (APPs): not only is the APP simplified compared to a low-level program doing the same task, but the task itself is simpler, since the object program is also simplified. For higher-order APPs, e.g. ones that write APPs themselves, the effect is exponential in order. We will refer to this phenomenon as the *recursive simplification* of the automatic programming problem.

1.3. Automatic Design

In the 1990s, the author and colleagues at Rutgers developed an architecture for the automatic design of microprocessors. It was based on recursive search within a hierarchy of abstraction levels, guided by a utility-based evaluation function. The utility function used different simulators at each abstraction level (ranging from RTL at the highest level to SPICE at the lowest) to evaluate the expected performance of a candidate design. This was augmented at any given level with an improved estimate recursively backed up from lower levels in the search tree by taking standard statistical measures of the population there. The evaluation function further estimated expected cost of search, and expected utility gain from further searching, by similar statistical means.

This system was able to produce pipelined, single cycle-per-instruction RISC microprocessor designs from high-level descriptions of the desired instruction set. It is the intent of the VARIAC project to adapt this architecture to produce programs from high-level specifications in essentially the same way.

1.4. Procedural Embedding of Knowledge

Classic AI programs such as SHRDLU embedded much of their knowledge in programs. SHRDLU, for example [3], had a language called PROGRAMMAR for expressing grammatical knowledge, and used MICRO-PLANNER for the semantics of its blocks-world. Follow-on knowledge-representation languages such as KRL-0 [4] attempted to retain

this stance, but typically lost Turing completeness in the face of the pressure towards declarative semantics: the more closely the representation resembled logic or natural language text, the more likely the semantics was to resemble syntactic inference mechanisms. This trend continued throughout the “expert systems” era.

The necessities of robotics, among other considerations, have forced a return to tightly-coded semantics in languages that are more directly adapted to real-time control than is the raw predicate calculus. These programs often make use of trained recognizers for predetermined concepts, but they do not originate their own concepts. No significant automatic programming is done, either by the robotic control programs or in the process of writing them. What is more, such low-level programs typically give up the generality of expressiveness that was the intent of declarative representations.

The main objections to procedural embedding are that procedural programs are opaque to inference, resistant to composition, and brittle under modification. Thus it seems desirable to design a programming language that is transparent, composable, and robust, while retaining the ability to specify semantics simply and directly.

2. Key Concepts in VARIAC

The design of VARIAC² is shaped by two main pressures. The first is the state of the art in programming language and compiling technology, which defines just how expressive a language we can implement with a usable degree of efficiency. The present section describes the language from this engineering point of view. The other pressure is the goal of building a system that can escape the hold of the so-called bootstrap fallacy and be capable of open-ended self-improvement. The following sections describe how such a system can be built given the capabilities described in this one.

There are many techniques, from AI and other fields, that are well understood and can be put to good use in the implementation of a programming language. Somewhere between the difficulty of a conventional optimizing compiler and that of a full-fledged automatic programming system with natural language input, the problem of compiling an extremely abstract yet formally specified language is a good match for much of the current inventory of symbolic AI techniques. Most of the following language features are implemented in one or more existing programming languages; no language has all of them. Each is an abstraction, in the sense that it specifies a set of concerns the programmer need not worry about (as garbage collection relieves the concern for storage allocation).

2.1. Higher-order Functional and Relational

Programming language theory addresses the procedural/declarative dichotomy by way of functional programming. A *pure* functional language describes a function as a declarative combination of primitive functions; no sequential or procedural semantics are implicit. In a major step in the direction of autogeny, higher-order functional languages compute functions directly as the values of expressions. This is a major source of abstraction and expressive power; see *e.g.* [5].

²VARIAC is an acronym for Vectors, Abstraction, and Recursion Integrated for Autogenous Cognition. Note also that in electrical engineering, a variac is an auto-transformer.

The sequence- and side-effects-free nature of functional languages encourages a “value semantics,” essentially a mathematician’s view of the data instead of an assembly language programmer’s. This is a substantial abstraction (although challenging to implement efficiently). Similarly, functional languages almost universally have automatic storage allocation. Research in functional languages has made substantial progress in the efficient compilation of programs using these abstractions.

Conceptually, a function is a table wherein the result to be returned is stored at an index specified by the argument(s). In a language with a level of abstraction above the concerns of data-storage in computer memory and time-sequencing of instructions, there is no need to make a distinction between a function represented as callable code, an array stored in coordinate memory, or a database relation whose access is based on whatever indexing scheme is appropriate. Once the view of data as a table is adopted, it is no longer necessary to assign a preferred direction from arguments to values; relational languages such as Prolog and Kanren take this view. In other words, the same predicate/relation can be used for, e.g., addition and subtraction: `plus(2,2,X)` or `plus(2,Y,4)`.

Human memory is associative; recall is based on parallel pattern-matching. There is no clear distinction between memory and program in the human mind. VARIAC’s view of program and data is similar: any object can be function or array. For example, the function `sin` takes a number and returns the trigonometric sine, while `Sin` is a one-dimensional array with an unbounded numerical index. The distinction (based on capitalization) is syntactic only; both refer to the same object. One can assign an array value to `Foo` and then use `foo` as a function without further definition. (Indeed, the expression `Sin^2 + Cos^2` evaluates to an object very similar to the scalar 1, the technical difference being its more limited domain when it in turn is used as a function!)

2.2. Scientific / Numerical

Any representations and algorithms must be couched in terms of primitive objects and operations. We assume that any possible representations and algorithms can ultimately be couched in terms of bit strings and Turing machines; universality in this sense is well-understood and need not concern us further. For reasons of practicality, however, VARIAC provides more developed representations, of which bit strings and state machines are a special case. These address two major areas of concern: descriptions of the physical world, and descriptions of language and algorithm constructs.

It is only reasonable, when describing the physical world, to avail ourselves of the centuries of work that have been done in the physical sciences to develop formal representations thereof. The language of physical science is numbers, vectors, matrices, and equations, simple and differential. Furthermore, this language is also used to describe statistics and probability, which are fast becoming an integral part of the AI practitioner’s toolkit.

Symbolic methods for manipulating programs and logical formulae, necessary for optimization of programs, are essentially of the same kind as those for manipulating equations and mathematical formulae. It is therefore surprising that there remains a gap between numerical languages (e.g. Fortran) and symbolic ones (Prolog) in practice. VARIAC attempts to close the gap and be equally facile at symbolic and numerical manipulation.

2.3. Discrete and Continuous

Arrays may be discrete or continuous in space and signals may be discrete or continuous in time. Pictures, sonar depth fields, maps, and so forth are among the many things that benefit from being treated as continuous, at a level of abstraction above the details of discretization.

For example, the occupancy field which is the basis of Hans Moravec's breakthrough Bayesian sensor fusion algorithm for mobile robots [6] is just such a surface, as is the sensor model which must be added to it. Suppose W_0 is an initial estimate of the occupancy field, Bearing and Pos status signals for the robot, Dist the distance reading from the sensor, and Model the learned sensor model, a 3D array indexed by the reading and 2 spatial dimensions. Then

$W_0 += \text{Bearing} \text{ rotate} \text{ Pos translate} [\text{Dist};;] \text{ Model}$

is the entire code for the world model. Its value, as well as those of its inputs, are reactive, i.e. signals that are functions of time. There are no loops or other control structure. The accumulate ($+=$) function adds successive values of a discrete signal or integrates a continuous one.

2.4. Frame-arrays and Tensor Fields

Minsky's concept of a "frame" was widely adopted by the AI community in the 1970s and 1980s; but Minsky himself has opined that the more important aspect of his idea, the notion of the frame-array, was largely ignored. The frame itself was an organizing principle for the data representing a situation as it might be observed and interpreted by a robot. The frame-array was a collection of such frames, indexed or generated in such a way that the results of typical actions or events were automatically predicted by the presentation of an appropriate new frame.

Consider a robot head with two degrees of freedom, such that its pose can be specified by a 2-vector. In a fixed environment, a camera in the head will return a specific pixel array for any given pose. A two-dimensional array, indexed by pose, of two-dimensional arrays of pixels is thus a very simplistic form of a frame-array for this robot. In physics and engineering, such a space mapping each point to a structured value is called a "tensor field." It is a device of enormously flexible representational power. Unlike physics, and more like Minsky, we allow any structured value, not only numeric arrays, to be "mapped into" each point of a field.

2.5. Reactive: the Garbage Collection of Time

Typically functional languages have addressed the declarative/procedural gap by being "impure," that is by mixing sequential semantics with the functional as does LISP, or by the use of "IO monads" as in HASKELL [7], an explicitly-computed trace of the interactions the program involves.

Another option, and the approach taken with VARIAC, is that the "program" is a declarative description of an equation, circuit, or machine that has a well-defined dynamic behavior in time. The equations of physics, as well as standard signal systems in control theory, have this semantics, explicitly referring to time derivatives and integrals of quantities. Simulation languages such as LABVIEW also take this approach, and it has much in common with stream- and dataflow-based languages.

Simply put, the semantics of a reactive³ language are that instead of a flow of control which sequentially activates the statements of the program, all the program elements are active all the time, as if they were components of a circuit. Such a program can be compiled (not without some difficulty!) into one which both takes advantage of such parallelism as is available, and uses scheduling and interrupts to emulate the “always-on” effect. Reactivity is thus to sequencing and coordination as garbage collection is to storage allocation: it provides a model of unbounded processor time just as CONS provides a model of unbounded storage. In both cases the resulting program is simplified from one which confronts the details directly.

The result of combining relational and reactive semantics is that a VARIAC program is the equivalent of a set of simultaneous equations, a circuit, or a constraint network. Given this view, it is straightforward to create structures such as semantic networks with parallel spreading activation behavior (e.g. [9]). Another paradigm that maps neatly into this framework is Minsky’s “Society of Mind” agencies (hereinafter “Minsky SOM”), with nodes activating each other in hierarchical cascades. These can be enhanced by virtue of the fact that VARIAC allows the passing of arbitrary values, including functions, from node to node.

2.6. *Reflective: First-class Types*

Modern programming language theory relies heavily on the theory of types. Following the Curry-Howard Isomorphism, any program using algebraic constructive types can be mapped onto a proof that the result has in fact the desired type. Besides being a boon to compiler writers (and thus to automatic programming), types are a major step in the direction of a language to describe data.

Another language to describe data is the string patterns in SNOBOL and its successors. Optimized specializations of these as language grammars are found in parser generators such as YACC.

Finally, symbolic mathematics programs such as MACSYMA have been developed, again originally an offshoot of AI (as in SAINT), which implement a considerable ability to describe patterns of numbers and other mathematical concepts.

VARIAC’s types are essentially a unification of these forms of specifying values and patterns. More importantly, they are an integral part of the language. Typically, existing programming languages either have static typing, meaning that the compiler reasons about types before emitting object code in which the types are implicit in the representations, or dynamic, meaning that types are explicitly represented in datastructures. Only in a few unusual languages (e.g. BERTRAND [10] and Q⁴) is type used as an integral part of the specification of the computation.

Types in VARIAC are values which can be combined and manipulated as easily as numbers or symbolic expressions. This is a key capability for general knowledge representation as well as for self-describing and self-extending programs. The type description ability is based on David McAllester’s ONTIC Language [11] which was in use as the description language for some automatic theorem provers in the 1990s but never expanded into a full programming language. It has much in common with both the patterns

³Not a particularly descriptive name, but the one that has come to be accepted in the programming language theory community. See e.g. [8].

⁴<http://www.musikwissenschaft.uni-mainz.de/~ag/q/qdoc.pdf>

and grammars mentioned above and with classic AI knowledge-representation languages such as KRL-0 and MDS [12].

Since programs and objects are unified in VARIAC (as in any higher-order functional language), types represent abstractions of programs and form a key ingredient of our automatic programming methodology. Rather than merely searching a syntactic space of programs, as is done in genetic programming for example, we can narrow the semantic space with abstractions of increasing specificity. (For example, when attempting the formulation of the inverse kinematics of a robot arm, an abstraction might include domain and range specifications, and the fact that the desired function is an inverse of a known forward kinematic function.)

3. From Programs to Cognition

A growing program will need to be both scientist and engineer in its own world – to construct explanatory and predictive models, and also to construct competent control programs as it learns skills. It must parse the world, as well as language it hears, into coherent structures of useful pieces. It must clothe frameworks of words with the fabric of imitated actions, cut and stitched to fit.

3.1. *Sigmas*

The simplest way to learn and predict is to copy a trace of the experiential stream into a recorded trajectory in some appropriate representation. In the simplest forms, this can be a series of points in some n -dimensional space. Conceptually straightforward, although computationally expensive, nearest-neighbor methods can then map any given situation to previously experienced ones (or to averages of clusters thereof) and predict the evolution of the present state by a simple geometric quadrature. Learning is simple – it consists of adding new experiential tracks to the memory.

In practice this is far too abstract a view. If the experiential stream were simply recorded without compression, it would imply a human long-term memory at least a billion times the size of current best estimates, and processing power orders of magnitude beyond that. In practice the “firehose of experience” is compressed into a few bits per second of memorable information. More than compressed, however, it is re-represented into the form of useful abstractions. Re-representation is well understood and is used extensively in most AI and robotics architectures, simple examples being the formation of maps and 3D models from rangefinder and camera datastreams.

In order for this CBR-like learning method to work, it is only necessary for the re-representation to map into a space with a useful metric susceptible to quadrature, and for the space to be abstract and compressed enough for the operations to be computationally tractable.⁵ As far as we know, this is not possible for general experience as a whole, but it is possible for many specific instances and categories. We claim that it is reasonable to identify a “concept” with a particular abstraction space (with a metric that supports

⁵Note that estimates of brain processing power give 1 or 2 orders of magnitude more processing power per bit of memory than is typical with von Neumann computer architectures, an argument that simple, brute-force hardware associative memory may be a more appropriate model than ingenious hashing and indexing schemes on serial processors.

CBR-like learning) and the functions that perform re-representation into that space from whatever other representations are available.

We refer to such a space, functions, the associative memory for trajectories, and the quadrature mechanism as a “sigma,” for situation-goal-memory-action. Such a sigma, in a reactive implementation, can be used as either a controller or prediction machine (in a manner similar to that of Minsky [13]) by re-routings of its addressing and output connections.

There is a straightforward translation from any reactive functional program with well-defined types to a network of appropriately connected sigmas (with appropriately initialized traces). Note that in this translation, not only are the units that capture the experiential memory of the system (if any) represented as functions implemented as interpolating associative memories, but so too are the functions that map between the different representation spaces, and thus define the concepts.

It is possible, if not trivial, to produce figures of merit for any such re-representation structure at the highest levels, which transforms the problem of experiential learning and concept formation to an abstract programming problem in a form amenable to our rational utility-based program construction system.

3.2. Active Production Networks

Networks of reactive functions map directly onto circuit models at any level from digital gate logic to the connected modules of digital signal processing, so it is essentially trivial to implement any of the many standard sensory and control architectures in the literature in this form. Somewhat less obvious is how to implement the complex interpretation capability that is conventionally done with search-based algorithms, such as natural-language parsing.

A method that shows promise in this regard is the Active Production Network developed by Mark Jones of (the then) Bell Telephone Laboratories in the 1980s [14]. It is a parallel message-passing semantic network model, with a structure that reflects the grammar it accepts. The network accepts words sequentially, and its state in terms of active messages and accumulated values at the nodes reflects its interpretation of each initial substring. This formulation of a grammar matches very well with spreading-activation semantic and agent networks. (Note that at a level of abstraction including both the pure functional and the reactive aspects, APNs and recursive descent parsers unify.⁶)

Experimental APNs have shown not only the basic ability to parse, but remarkable robustness in the face of misformed or noisy input. In addition, they interface very naturally to semantic models expressed as reactive functional programs, in such a way that semantic constraints are automatically incorporated into the parsing process. It is suggestive to note that some theories of evolutionary neuroscience [15] place language and fine manipulation ability together. Minsky SOM agencies for complex motor control look remarkably like APNs; we conjecture that a unified formulation can be used for act interpretation as well, and for language understanding and generation.⁷

⁶i.e. the essential distinctions are differing treatments of concerns that the abstraction has already automated.

⁷Unpublished experiments by Jones using APNs for text generation were promising although preliminary. Private communication with the author, 1988.

3.3. An Economy of Mind

It is becoming common for computational neuroscientists to speak of the dopamine reward-prediction error signal mechanism as the “currency” of the brain. It clearly performs a central role in the motivation and decision-making function. Less obviously, however, it turns out to play a crucial role in learning and concept formation as well.

This agrees well with the experience of the author and his colleagues at Rutgers in the field of agoric and utility-based design, if learning is equated with designing a complex structure such as a program. In such a design process, it is necessary to evaluate components and sub-assemblies in order to choose between design alternatives, and ultimately to produce a valuation surface for an abstract space of partial designs. In the brain, the dopamine system “backs up” the reward signal to perform a kind of credit assignment. In agoric systems going back to Holland’s “bucket brigade,” a market model performs the same function.

Utility-based design performs a best-first search in a tree-structured space generated by an abstraction hierarchy over the object structure. As such it is easily controlled and is appropriate for use in the base VARIAC language implementation. A more complete, parallel, and open-ended market model, such as CHARLES SMITH [16], seems more appropriate to an architecture of general intelligence, but comes with many more open questions.

4. Towards an Autogenous Architecture

The field of AGI, as distinct from more general application-oriented AI, has as yet little general agreement on the specifics of the architecture of a general cognitive agent. One methodological element that does seem much more prevalent in AGI than in AI at large, however, is the notion of building a “child machine,” as proposed by Turing, and educating it to adult competence rather than building a mature system *in toto* [17].

Given this position, our experimental program for AGI development consists of a blocks-world situation reminiscent of SHRDLU. It will employ a physical robot (currently under construction) with arm(s) and binocular vision. It is immobile, allowing for relatively powerful processing hardware; the current setup has ten general-purpose and approximately 300 vector processing units. This is expected to increase over the course of the project.

The software architecture of the robot controller is also similar to that of SHRDLU, except that the world simulator is replaced by a robot controller and vision interpretation stack. The most innovative element of the SHRDLU architecture, the crosstalk between the parser and the world model which was used for disambiguation [18], is retained and extended. The grammar, model, controller, and higher-level vision are all implemented as reactive function networks capable of self-extension or modification; the form and extent of self-modification possible or desirable is the key object of the research and as yet a very open question.

4.1. Learning

We conjecture that most learning is done by imitation and/or analogy. Early experiments will thus be of the form *I touch the block; you touch the block*. Key early issues are the

ability to parse the sensory stream in space into objects and in time into actions. Then comes the ability to match objects and actions with words, and finally the ability to put those together into novel structures.

We further conjecture that learning by verbal instruction is similar to learning by imitation. If the language of thought is formulated properly and integrated with semantic cognition, parsing sentences should result in structures not unlike the ones from parsing the experiential stream of watching an exemplar activity. Indeed, if the language is learned by experience, these will be the only structures available to assign as meanings to new words and constructs!

The emphasis in the 80s on applications displaced blocks-world experiments from the forefront of AI, but we feel that this was a mistake. A deskbound robot could, after it became proficient with blocks, proceed to learn with more complex toys, say Lego sets. It could learn to assemble or repair complex machinery. It could play chess, Monopoly, or marbles; read, write, and draw pictures. It could be a laboratory chemist or a hibachi chef. If we assume – and this is the scientific hypothesis to be determined by experiment – that a desktop world is a sufficient semantic domain to be a basis for understanding general language, then a desktop robot could grow to be a completely open-ended, general intelligence.

4.2. Implementation

The kernel of VARIAC is a unification-augmented term-graph rewriting engine with reflexive types. That is, each expression has a type, which can itself be any expression. (This continues recursively until grounded in a primitive type.) This generates a language, if used raw, of an expressive power similar to PROLOG (because it has unification and backtracking), but completely declarative. For example, a symbolic algebra package can be written in about a page of code.

The status of VARIAC as of even date represents about a year of mostly exploratory and definitional work. The current interpreter is third in a series of implementations adding successively more of the listed abstractions to the semantic base. In a longer retrospective, VARIAC is a development of the machine definition language Caslor [19] and the methods of program construction derive from the AI-in-design work of [16].

5. Summary

No matter what formalisms and datastructures it manipulates, an AI must, in effect, write programs. Understanding is in essence the ability to simulate and thus to predict. Learning is the ability to understand things one did not before, and thus involves implementing new simulations – creating new representations and new algorithms. *Every AI system implemented to date* has been a “wind-up toy,” built with unextendable basic representations designed entirely by human programmers.⁸

We can cut the problem of AGI in half by designing a programming language at such a high level that it significantly reduces not only the work we must do to implement the system initially, but the work the system must do to reprogram, and thus to improve, itself. The first half of the problem, implementing the language thus defined, avails us

⁸With the possible exception of some AGI systems under development!

of many of the narrow-AI tools that have been developed in the past half-century: Expert systems, planning and design algorithms, theorem provers, utility-based search, and many other standard AI techniques are applicable to the task of compiling a very-high-level but formally-specified programming language.

More important than the specific functions and datastructures of the language are its abstractions, the concerns it lets the programmer (including the automatic programming system) ignore. All of the abstractions we have chosen have been implemented in some existing programming language, with the exception of the reflexive type system. The integration of these abstractions into a single coherent language is certainly non-trivial but appears to be a straightforward development project.

References

- [1] J. Storrs Hall, *Self-Improving AI: an Analysis*, AI@50, Dartmouth College, June, 2006
- [2] Cordell C. Green, Richard J. Waldinger, David R. Barstow, Robert A. Elschlager, Douglas B. Lenat, Brian P. McCune, David E. Shaw, and Louis I. Steinberg, *Progress report on program-understanding systems.*, Stanford University, Stanford, CA, 1974
- [3] Terry Winograd, *Procedures as a Representation for Data in a Computer Program for Understanding Natural Language*. MIT AI Technical Report 235, February 1971
- [4] Bobrow, D. G., and T. Winograd. (1977). An overview of KRL-0, a knowledge representation language. *Cognitive Science* 1(1):3-46.
- [5] J. Hughes, *Why Functional Programming Matters*, Computer Journal 32:2 pp98-107, 1989.
- [6] Hans P. Moravec. *Sensor fusion in certainty grids for mobile robots*. *AI Magazine*, 9(2):61–74, 1988.
- [7] Philip Wadler. *Comprehending Monads*. Proceedings of the 1990 ACM Conference on LISP and Functional Programming, Nice, 1990.
- [8] Zhanyong Wan, *Functional Reactive Programming for Real-Time Reactive Systems*, Ph.D. Dissertation, Yale University, 2002.
- [9] Ian Horswill. "Cerebus: A Higher-Order Behavior-Based System." *AI Magazine*, 2001.
- [10] Wm Leier. *Constraint Programming Languages: Their Specification and Generation*. Addison-Wesley Publishing Co., Reading, Massachusetts, 1988.
- [11] David McAllester. *Ontic proof verification system*. <ftp://ftp.ai.mit.edu/pub/ontic/ontic.tar.Z>.
- [12] Chitoor V. Srinivasan, *Model Space of the Meta Description System*, Report SOSAP-TR-19, Department of Computer Science, Rutgers University. 1976. MDS was explicitly self-referential (hence the “Meta”).
- [13] Minsky, Marvin. *The Emotion Machine*. New York, Simon and Schuster, 2006, pp 159-160.
- [14] Mark A. Jones and Alan S. Driscoll, *Movement In Active Production Networks*, Proc Assoc. Comp. Ling. 1985, pp. 161-166; citeseer.ist.psu.edu/580365.html
- [15] Ornella Castelli and Carlo Peretto, *The Phylogenesis of Language: The Grammar of Gestures and the Manipulation of Words*, Human Evolution 21(1), March 2006
- [16] J Storrs Hall, Louis Steinberg, and Brian D. Davison (1998) *Combining agoric and genetic methods in stochastic design*, Nanotechnology 9(3) (September 1998) 274-284; Steinberg, Hall, and Davison, (1998): *Highest Utility First Search Across Multiple Levels of Stochastic Design*, pp. 477-484. Proceedings of the Fifteenth National Conference on AI, Madison, 1998.
- [17] David Vernon, Giorgio Metta, and Giulio Sandini, A Survey of Artificial Cognitive Systems: Implications for the Autonomous Development of Mental Capabilities in Computational Agents, IEEE Transactions on Evolutionary Computation, Special Issue on Autonomous Mental Development, 11(2), 2007
- [18] Terry Winograd, *Understanding Natural Language*, New York: Academic Press, 1972, p. 5.
- [19] J. Storrs Hall, (1999): *Towards a Hardware Description Language for Molecular Machinery*, Seventh Foresight Conference on Nanotechnology, Santa Clara, CA

Probabilistic Quantifier Logic for General Intelligence: An Indefinite Probabilities Approach

Matthew IKLÉ^a, Ben GOERTZEL^b

^a*Adams State College, Alamosa, Colorado
and Novamente LLC*
^b*Novamente LLC*

Abstract: Indefinite probabilities are a novel technique for quantifying uncertainty, which were created as part of the PLN (Probabilistic Logic Networks) logical inference engine, which is a key component of the Novamente Cognition Engine (NCE), an integrative AGI system. Previous papers have discussed the use of indefinite probabilities in the context of a variety of logical inference rules, but have omitted discussion of quantification. Here this gap is filled, and a mathematical procedure is provided allowing the propagation of indefinite probabilities through universal and existential quantifiers, and also through a variety of fuzzy quantifiers corresponding to natural language quantifiers (such as “few”, “many”, “a lot”, “hardly any”, etc.). Proper probabilistic handling of various quantifier transformation rules is also discussed. Together with the ideas in prior publications, and a forthcoming sequel paper on indefinite probabilities for intensional inference, these results allow probabilistic logic based on indefinite probabilities to be utilized for the full scope of inferences involved in intelligent reasoning. To illustrate the ideas and algorithms involved, we give two concrete examples: Halpern’s “crooked lottery” predicate, and a commonsense syllogism that uses fuzzy quantifiers together with the standard PLN term logic deduction rule.

1. Introduction

Probability theory plays an increasingly large role in AI research, spanning the divide between AGI and narrow AI research, and infiltrating every subdomain of the AI field. Probabilistic methods hold a prominent role in linguistics ([1]), robotics ([2]) speech processing ([3], [4]) and data analysis and modeling ([5]), to name only a handful of the many areas that could be cited. Our own venture into the domain of Artificial General Intelligence, the Novamente Cognition Engine (NCE) ([6]), utilizes probability theory as a common methodology and language binding together diverse cognitive components.

Probability theory in itself, however, is a very general tool, and may be applied in a variety of different ways, in conjunction with a diversity of other formalisms. One of the key foci of recent research regarding the utilization of probability theory for AI is the unification of probability and logic ([7], [8]). Of course, the simple unification of probability and logic is trivial, as logic is a general tool that can be used to reason about anything; but unifying the two formalisms in a way that is helpful in terms of

constructing an AI system is another story. One approach to carrying out this unification is the PLN (Probabilistic Logic Networks) framework ([9]), which is a key component of the Novamente Cognition Engine (NCE; [6]), an integrative AGI system developed by the authors and their colleagues during the period since 2001. PLN has been specifically constructed to unify probability and logic in a manner that

- supports the full scope of inferences required within an intelligent system, including e.g. first and higher order logic, intensional and extensional reasoning, and so forth
- lends itself naturally to methods of inference control that are computationally tractable, and able to make use of the inputs provided by non-logical cognitive mechanisms (such as dynamic attention allocation and evolutionary learning, both of which play roles in the NCE)\

PLN as a general framework supports multiple measures of uncertainty, but the primary measure utilized currently is “indefinite probabilities,” a novel measure formed by hybridizing Walley’s imprecise probabilities with the Bayesian notion of credible intervals ([9], [10]). In order to utilize indefinite probabilities with PLN, each of PLN’s logical inference rules must be associated with an “indefinite truth value formula or procedure,” which tells you, given the indefinite truth values associated with the premises of the inference rule, what is the indefinite truth value associated with the conclusion of the rule. Prior papers have given indefinite truth value formulas for a number of PLN inference rules, but have not dealt with quantifiers. In this paper we remedy this deficit, and discuss the propagation of indefinite probabilities through universal and existential quantifiers as well as fuzzy quantifiers.

The overall goals of our approach to quantifiers, as presented here, are as follows:

- 1) logical and conceptual consistency
- 2) agreement with standard quantifier logic for the crisp case (for all expressions to which standard quantifier logic assigns truth values)
- 3) gives intuitively reasonable answers in practical cases
- 4) compatibility with probability theory in general and PLN in particular
- 5) handles fuzzy quantifiers as well as standard universal and existential quantifiers

The application to fuzzy quantifiers is particularly conceptually satisfying, in that it places these slippery notions, with their close connection to ambiguous natural language, on a firm probabilistic foundation, in a way that (we argue) is more conceptually coherent and pragmatically useful than prior attempts in similar directions. Zadeh refers to fuzzy quantifiers characterizing absolute cardinality (such as several, few, many etc.), as fuzzy quantifiers of the first kind; those characterizing relative cardinality (such as most, almost all, many) as fuzzy quantifiers of the second kind; and he also introduces a third-kind of fuzzy quantifiers which are ratios of quantifiers of the second kind ([11], [12]). Here we explicitly demonstrate how to treat analogs of fuzzy quantifiers of the first kind within the indefinite probabilities framework. Straightforward generalizations of the ideas contained here, however, can also generate analogs of fuzzy quantifiers of the second and third kinds.

There is no objective standard via which to compare different approaches to the combination of uncertain inference and quantifier inference. However, our own feeling is that the approach articulated here is more satisfactory from a pragmatic AI perspective than prior attempts. As we are committed to a probabilistic foundation for AI, we are not fully satisfied with nonprobabilistic fuzzy approaches; and the classical probabilistic approaches appear to us to have conceptual problems. For instance,

Halpern's approach [7] involves a counterintuitive mixture of two different kinds of probabilities (subjective and frequentist). Our approach does not involve these sorts of interpretive subtleties, avoiding them due to the introduction of third-order probabilities. On the other hand, the introduction of quantifiers into Markov Logic Networks as done in [13], while interesting and surely useful for certain applications, does not comprise a suitably general framework for uncertain quantifier logic: it deals with evaluation of expressions involving quantifiers across databases, but not with abstract quantifier manipulations. The approach presented here is comprehensive, conceptually coherent, probabilistically grounded and computationally tractable, and for these reasons we suggest it to be an adequate formulation of uncertain quantifier logic for AGI purposes.

2. Review of Indefinite Probabilities

As described in ([9], [10]), indefinite probabilities were motivated in part by Walley's imprecise probabilities ([14]). A truth-value for an indefinite probability takes the form of a quadruple $([L, U], b, k)$. The meaning of such a truth-value, attached to a statement S is, roughly: There is a probability b that, after k more observations, the truth value assigned to the statement S will lie in the interval $[L, U]$. Just as with Walley's imprecise probabilities, we interpret an interval $[L, U]$ by assuming some particular family of distributions (usually Beta) whose means lie in $[L, U]$.

The inclusion of the credibility level b , not present in Walley's interval truth values, allows our intervals to remain narrower than those produced by Walley's and other interval probabilities in the literature. As we showed in ([10]), our approach reduces to Walley's imprecise probabilities by setting $b = 1$. In that prior paper we also discuss, in detail, the philosophical and conceptual underpinnings of indefinite probabilities and so will not repeat those here.

2.1. Inference with Indefinite Probabilities

We now review the basic methodology by which indefinite probability formulas may be derived corresponding to inference rules. More detail on this methodology is given in ([9], [10]).

We assume that the inference rules in question already come with truth value formulas that apply in the case the probability value attached to each premise is exactly known (an ideal limiting case that will essentially never occur in reality). An example inference rule that has been treated this way is the term logic deduction rule

```
Inh A B
Inh B C
|-
Inh A C
```

where Inh is shorthand notation for ExtensionalInheritance, a specific Novamente/PLN link type as defined in ([9]), which refers to a probabilistic subset relationship.

The probabilistic truth value formula for the deduction rule is given by

$$S_{AC} = S_{AB} S_{BC} + (1-S_{AB}) (S_C - S_B S_{BC}) / (1- S_B).$$

To execute inference formulas (corresponding to inference rules such as the deduction rule given just above) using indefinite probabilities, we make heuristic distributional assumptions. We assume a “second-order” distribution that has $[L, U]$ as a $(100b)\%$ credible interval. We then assume “first-order” distributions whose means are drawn from the second-order distribution. These distributions are to be viewed as heuristic approximations intended to estimate unknown probability values existing in hypothetical future situations. While the accuracy and utility of the indefinite probability method may depend on the appropriateness of the distributional assumptions, we have found that the beta and bimodal families seem adequate for most cases arising in real-world inference problems.

Given a logical inference rule, the indefinite probability based inference process proceeds in three basic steps.

1. Given intervals, $[L_i, U_i]$, of mean premise probabilities, we first find a distribution from the “second-order distribution family” supported on $[L_1, U_1] \supset [L_i, U_i]$, so that these means have $[L_i, U_i]$ as $(100 \cdot b_i)\%$ credible intervals.
2. For each premise, we use Monte-Carlo methods to generate samples for each of the “first-order” distributions with means given by samples of the “second-order” distributions. We then apply the inference rules to the set of premises for each sample point, and calculate the mean of each of these distributions.
3. Find a $(100 \cdot b_i)\%$ credible interval, $[L_f, U_f]$, for this distribution of means; e.g. by assuming a symmetric credible interval about the mean.

As an example, note the results obtained by doing indefinite probability calculations for the following deduction:

Women are beautiful.
 Beautiful things bring happiness.
 |-
 Women bring happiness

We summarize the truth value premises and conclusion in the following table:

Premises	Truth Value
Women	$\langle [0.45, 0.55], 0.9, 10 \rangle$
Women are beautiful	$\langle [0.8, 0.95], 0.9, 10 \rangle$
Beautiful things	$\langle [0.4, 0.8], 0.9, 10 \rangle$
Beautiful things bring happiness	$\langle [0.8, 0.95], 0.9, 10 \rangle$
Happiness	$\langle [0.4, 0.9], 0.9, 10 \rangle$
Conclusion	Truth Value
Women bring happiness	$\langle [0.76939, 0.87028], 0.9, 10 \rangle$

3. Quantifiers in Indefinite Probabilities

The above approach works perfectly well for many inference rules, but is inadequate to handle universal, existential or fuzzy quantifiers. The best way we have found to handle quantifiers within the indefinite probabilities framework is to introduce another level of complexity and utilize third-order probabilities. To understand this, we first consider the problem of “direct evaluation” of the indefinite truth values of universally and existentially quantified expressions.

We solve this problem via taking a semantic approach that is considerably conceptually different from the one standardly taken in formal logic. Normally, in logic, expressions with unbound variables are not assigned truth values; truth value assignment comes only with quantification. In our approach, however, we assign truth values to expressions with unbound variables, yet without in doing so binding the variables. This is unusual but not contradictory in any way: an expression with unbound variables, as a mathematical entity, may certainly be mapped into a truth value without introducing any mathematical or conceptual inconsistency. This allows one to define the truth value of a quantified expression as a mathematical transform of the truth value of the corresponding expression with unbound variables, a notion that is key to our approach.

This unusual semantic approach adds a minor twist to the notion that our approach to uncertain inference on quantified expressions reduces to standard crisp inference on quantified expressions as a special case. The twist is that our approach reduces to the standard crisp approach in terms of truth value assignation for all expressions for which the standard crisp approach assigns a truth value. However, our approach also assigns truth values to some expressions (formulas with unbound variables) to which the standard crisp approach assigns no truth value.

Following up on this semantic approach, we will now explain how, if we have an indefinite probability for an expression $F(t)$ with unbound variable t , summarizing an envelope E of probability distributions corresponding to $F(t)$, we may derive from this an indefinite probability for the expression “ $\text{ForAll } x, F(x)$ ”? (Having carried out the transform in this direction, it will then be straightforwardly possible to carry out a corresponding transform in reverse.) The approach we take here is to consider the envelope E to be part of a higher-level envelope $E1$, which is an envelope of envelopes. The question is then: given that we have observed E , what is the chance (according to $E1$) that the true envelope describing the world actually is almost entirely supported within $[1-e, 1]$, where the latter interval is interpreted to constitute “essentially 1” (i.e., e is the margin of error accepted in assessing ForAll-ness), and the phrase “almost entirely supported” is defined in terms of a threshold parameter?

Similarly, in the case of existential quantification, we want to know the indefinite probability corresponding to “ $\text{ThereExists } x, F(x)$.” The question is then: given that we have observed E , what is the chance (according to $E1$) that the true envelope describing the world actually is *not* entirely supported within $[0, e]$, where the latter interval is interpreted to constitute “essentially zero” (i.e., e is the margin of error accepted in assessing ThereExists-ness)?

The point conceptually is that quantified statements require you to go one level higher than ordinary statements. So if ordinary statements get second-order probabilities, quantified statements must get third-order probabilities. And, the same line of reasoning that holds for “crisp” universal and existential quantifiers, turns out to hold for fuzzy quantifiers as well. In fact, in the approach presented here, crisp

quantifiers are innately considered as an extreme case of fuzzy quantifiers, so that handling fuzzy quantifiers doesn't really require anything extra, just some parameter-tuning.

The following subsections elaborate the above points more rigorously.

3.1. Direct Evaluation of Universally Quantified Expressions

We first consider the case of the direct evaluation of universally quantified statements, an inference rule for which the idea is as follows: Given an indefinite truth value for $F(t)$, we want to get an indefinite TV for $G = \text{ForAll } x, F(x)$.

The roles of the three levels of distributions are roughly as follows. The first- and second-order levels play the role, with some modifications, of standard indefinite probabilities. The third-order distribution then plays the role of “perturbing” the second-order distribution. The idea is that the second-order distribution represents the mean for the statement $F(x)$. The third-order distribution then gives various values for x , and the first-order distribution gives the sub-distributions for each of the second-order distributions.

The process proceeds as follows:

Step 1: Calculate [lf1,uf1] Interval for the third-order distribution.

This step proceeds as usual for indefinite probabilities: see ([9], [10]). Given L , U , k , and b , set $s = 0.5$. We want to find a value for the variable **diff** so that the probability density function defined by

$$f(x) = \frac{(x - L1)^{ks} (U1 - x)^{k(1-s)}}{\int_{L1}^{U1} (x - L1)^{ks} (U1 - x)^{k(1-s)} dx}$$

where $L1=L-\text{diff}$ and $U1=U+\text{diff}$ is such that

$$\frac{\int_{L1}^L (x - L1)^{ks} (U1 - x)^{k(1-s)} dx}{\int_{L1}^{U1} (x - L1)^{ks} (U1 - x)^{k(1-s)} dx} = \frac{1-b}{2}$$

and

$$\frac{\int_{U1}^{U} (x - L1)^{ks} (U1 - x)^{k(1-s)} dx}{\int_{L1}^{U1} (x - L1)^{ks} (U1 - x)^{k(1-s)} dx} = \frac{1-b}{2}.$$

Once one of these last two integrals is satisfied, they both should be. Alternatively, one can find **diff** for which

$$\frac{\int\limits_{L}^{U} (x - L)^{ks} (U - x)^{k(1-s)} dx}{\int\limits_{L1}^{U1} (x - L1)^{ks} (U1 - x)^{k(1-s)} dx} = b.$$

Step 2: Generate vectors of means for perturbed F(x) values.

Generate vectors of mean values for each premise.

Step 2.1: Generate values from desired “third-order” distribution family.

At present we are using only beta distributions.
Generate a vector of length $n1$ of random values chosen from a standard beta distribution.

Step 2.2: Scale random means to interval [lf1,uf1]

Scale the vector from step 2.1 to [lf1,uf1] using a linear transformation.

Step 3: Generate symmetric intervals [if2[i], if2[j]] for each of the means found in step 2.

These intervals are now the desired [L1, U1] intervals for the third-order distributions.

Step 4: Generate the second-order distributions.

For each mean for the third-order distributions, generate a sub-distribution. These sub-distributions represent the second-order distributions.

Step 5: Generate first-order distributions with means chosen from the second-order distributions.**Step 6: Determine the percentage of elements in each first-order distribution that lie within the interval [1-e, 1].**

Recall that we are using the interval [1-e, 1] as a “proxy” for the probability 1. The goal here is to determine the fraction of the first-order distributions that are “almost entirely contained” in the interval [1-e, 1]. By almost entirely contained, we mean that the fraction contained is at least proxy_confidence_level (PCL).

Step 7: Find Conclusion ([L,U],b) Interval

For each of the third-order means, we calculate the average of all of the second-order distributions that are almost entirely contained in [1-e, 1], giving a list of $n1$ elements, **probs**, of probabilities. We finally find the elements of **probs** corresponding to quantiles

$n1 * (1-b)/2$.round for L, and
 $n1 * (0.5+b)/2$.round for U.

3.1.1. The ThereExists Rule

We obtain the ThereExists rule through the equivalence

ThereExists x, F(x) $\Leftrightarrow \sim$ [ForAll x, \sim F(x)].

4. Propagating Indefinite Probabilities through Quantifier-Based Inference Rules

As well as “directly evaluating” quantifiers in the manner of the prior section, it is also necessary within a logical reasoning system to carry out various quantifier manipulations. We now discuss a variety of transformation rules that work on quantifiers, drawn from standard predicate logic.

First, we have already seen that what is called “the rule of existential generalization” holds in the indefinite probabilities framework (this is just a reformulation of what we have called “direct evaluation” of existentially quantified expressions, above):

$$\begin{aligned} 1) \quad & F(c) <[L,U], b, k> \\ & | \\ & \exists \$x, F(\$x) <[L,U], b, k> \end{aligned}$$

where c may be any expression not involving $\$x$.

Next, consider universal specification:

$$\begin{aligned} 2) \quad & \forall \$x, F(\$x) <[L,U], b, k> \\ & | \\ & F(c) <[L,U], b, k> \end{aligned}$$

where c is any expression not involving $\$x$.

To see that universal specification also holds with indefinite probabilities, given the truth value above for $\text{ForAll } \$x, F(\$x)$, we can obtain an indefinite truth value for $F(t)$. We then use the mean of $F(t)$ over all values t , as an heuristic approximation to $F(c)$ for a given value c .

We have already also seen, at least implicitly, that all the standard quantifier exchange formulas hold for indefinite probabilities:

$$\begin{aligned} 3) \quad & \neg(\exists x)Fx \Leftrightarrow (\forall x)\sim Fx \\ & (\exists x)\sim Fx \Leftrightarrow \neg(\forall x)Fx \\ & \neg(\exists x)\sim Fx \Leftrightarrow (\forall x)Fx \\ & (\exists x)\sim Fx \Leftrightarrow \neg(\forall x)\sim Fx \end{aligned}$$

For our last transformation rule, we consider the operation of removing constants from within existential quantifiers. In predicate logic we have that:

$$4) \quad \forall X: G \text{ AND } F(x) = G \text{ AND } \forall X: F(x)$$

Unlike the case for crisp predicate logic, however, this rule is not, in general, true using indefinite probabilities. For example, consider the following set of premises with parameter settings $e=0.5$ and $PCL=0.7$: truth value for $G = <[0.45, 0.46], 0.9, 10>$ and truth value for $F(x) = <[0.71, 0.72], 0.9, 10>$. Then the result for $\forall X: G \text{ AND } F(x)$ becomes $<[0.0, 0.04913], 0.9, 10>$, while that for $G \text{ AND } \forall X: F(x)$ is $<[0.23046, 0.28926], 0.9, 10>$. On the other hand, we note that a different set of premises can yield

similar results from the two approaches. Assuming the same parameter values for e and PCL, and truth values for both $F(x)$ and G of $\langle [0.99, 1.0], 0.9, 10 \rangle$ gives a result of $\langle [0.98331, 0.99626], 0.9, 10 \rangle$ using $\forall X: G \text{ AND } F(x)$, and a similar result of $\langle [0.98344, 0.99620], 0.9, 10 \rangle$ using $G \text{ AND } \forall X: F(x)$.

For insight into what is happening here, we view $H(F)(t) = G \text{ AND } F(t)$ as a distortion of the distribution of F . In addition, if $J(F) = \forall x J(x)$, then $J(F)$ is a nonlinear distortion of F , so that even though $H(F)$ is a linear distortion, it need not commute with J . An obvious and interesting question is then: Under what combination of premise values and parameter settings do the operators H and J ‘almost’ commute? Due to space considerations, we defer a thorough study of that question to a future paper. It does appear, however, that premise values near 1 lead to better commutativity than do values farther from 1.

5. Fuzzy Quantifiers

Analyzing the indefinite probabilities approach to the quantifiers *ForAll* and *ThereExists*, it should be readily apparent that indefinite probabilities provide a natural method for “fuzzy” quantifiers such as *AlmostAll* and *AFew*.

In our discussion of the *ForAll* rule above, for example, the interval [PCL, 1] represents the fraction of bottom-level distributions completely contained in the interval [1-e, 1]. Recall that the interval [1-e, 1] represents a proxy for probability 1.

In analogy with the interval [PCL, 1] representing the *ForAll* rule, we can introduce the parameters `lower_proxy_confidence` (LPC) and `upper_proxy_confidence` (UPC) so that the interval [LPC, UPC] represents an *AlmostAll* rule or *AFew* rule. More explicitly, by setting $[LPC, UPC] = [0.9, 0.99]$, the interval could now naturally represent *AlmostAll*.

Similarly, the same interval could represent *AFew* by setting LPC to a value such as 0.05 and UPC to, say 0.1.

Through simple adjustments of these two proxy confidence parameters, we can thus introduce a sliding scale for all sorts of fuzzy quantifiers. Moreover, each of these fuzzy quantifiers is now firmly grounded in probability theory through the indefinite probabilities formalism.

6. Examples

To further elucidate the above formalism, we now consider two examples. For our first example, we consider an example drawn from [15], which is there called the “crooked lottery” and extensively discussed:

$$[\neg \exists x : \text{Winner}(x) \rightarrow \text{false}] \& [\exists y : \forall x : (\text{Winner}(x) \parallel \text{Winner}(y)) \rightarrow \text{Winner}(y)]$$

The first clause is intended to represent the idea that everyone has a nonzero chance to win the lottery; the second clause is intended to represent the idea that there is one guy, y, who has a higher chance of winning than everybody else. In [15]

Halpern examines various formalisms for quantifying uncertainty in a logical reasoning context and assesses which ones can provide a consistent and sensible truth-value evaluation for this expression.

Evaluating the truth value of this expression using indefinite probabilities, with PCL=0.8 and e=0.3 for the ThereExists rule and the complement values of PCL=0.2 and e=0.7 for the ForAll rule, we obtain the following results:

Premises	Truth Value
Winner(x)	<[0.05, 0.1], 0.9, 10>
Winner(y)	<[0.25, 0.5], 0.9, 10>
[Winner(x) Winner(y)] → Winner(y)	<[0.8, 0.9], 0.9, 10>
Conclusion	Truth Value
Clause 1: ~ThereExists x : Winner(x) → false	<[0.0517, 0.2642], 0.9, 10>
Clause 2: ThereExists y : [ForAll x : (Winner(x) Winner(y)) → Winner(y)]	<[0.7312, 0.9976], 0.9, 10>
Clause 1 AND Clause 2	<[0.0307, 0.2470], 0.9, 10>

For our second example, we consider an extension of the example we used for the standard PLN deduction rule:

Many women are beautiful.
 Almost all beautiful things bring happiness.
 |-
 Many women bring happiness

In order to use the indefinite probabilities formalism, we first need to determine appropriate values for the parameters LPC and UPC to represent the fuzzy concepts “many” and “almost all.” In practice, in the case where these rules are used within an integrative AGI system such as the NCE, appropriate values for these fuzzy concepts will be determined by the context in which they appear. In one context, for example, the interval [0.8, 0.9] might represent the idea “many,” but in a different situation, we may wish for [0.6, 0.95] to represent “many.”

For our example we set e=0.1. Let us suppose that “many” is represented by the interval [LPC, UPC]=[0.4, 0.95], and “almost all” by the interval [0.9, 0.99]. We will also assume identical truth-values to those in the previous example. The sequence of conclusions is then illustrated in the following tables.

Premises	Truth Value
Women	<[0.45, 0.55], 0.9, 10>
An individual woman is beautiful	<[0.8, 0.95], 0.9, 10>
Conclusion	Truth Value
Many women are beautiful	<[0.35451, 0.63574], 0.9, 10>

Premises	Truth Value
Beautiful things	<[0.4, 0.8], 0.9, 10>
A beautiful thing brings happiness	<[0.8, 0.95], 0.9, 10>

Conclusion	Truth Value
Almost all beautiful things bring happiness	<[0.03906, 0.37464], 0.9, 10>

Premises	Truth Value
Women	<[0.45, 0.55], 0.9, 10>
Many women are beautiful	<[0.35451, 0.63574], 0.9, 10>
Beautiful things	<[0.4, 0.8], 0.9, 10>
Almost all beautiful things bring happiness	<[0.03906, 0.37464], 0.9, 10>
Happiness	<[0.4, 0.9], 0.9, 10>
Conclusion	Truth Value
Many women bring happiness	<[0.41308, 0.53068], 0.9, 10>

7. Conclusions

The issues we have considered here are specific and technical, yet they address a problem that is key to the overall project of creating powerful artificial intelligence. If one wishes to create an AI system that carries out explicit probabilistic estimations, and also explicit logical reasoning, then one is faced with the problem of unifying probability and logic in an elegant and easily controllable way. The PLN framework represents our general solution to this problem; and, in order to be applied in a general way, PLN's truth value formulas must handle quantifiers both crisp and fuzzy. The PLN mathematics must make it possible to propagate uncertain truth values (in whatever representation is chosen, currently indefinite probabilities) through every logical inference rule utilized, including those involving abstract constructs such as quantifiers. Here we have explained how this may be accomplished.

By incorporating a third level of distributions, we have extended the PLN indefinite probabilities method to handle a wide variety of both crisp and fuzzy quantifiers, including the quantifiers that occur in natural language. This is interesting from a purely theoretical perspective, and also from a pragmatic AGI perspective. During the next year or two we expect to put the formulas presentd here to work in carrying out inferences on logical relationships derived from natural language understanding and perceptual data analysis within the NCE. Due to the harmony of these formulas with the overall PLN inference framework, we don't expect inference control to be a major issue: bringing quantifier-based truth value estimation within the scope of PLN, means bringing it within the scope of PLN's powerful inference-control methodology, which uses the history of prior inferences conducted to adaptively prune backwards and forwards chaining inference trees.

Finally, a note on computational tractability may be worthwhile. Although mathematically abstract, the formulas given here are actually not among the more computationally intensive of the indefinite probabilities formulas. The term logic deduction formula presents much greater computational challenges. On modern computers the implementation of the procedures given here yields code that is far from being a bottleneck in the context of a system such as PLN, which has fairly expensive

inference control mechanisms going on as well as the execution of inference steps. By all indications the results described here have the potential to be a valuable part of pragmatic probabilistic logical inference in the Novamente Cognition Engine and potentially other AI approaches as well.

References

- [1] Christopher D. Manning, Hinrich Schuetze, Foundations of Statistical Natural Language Processing, Cambridge, MA, MIT Press, 1999.
- [2] Sebastian Thrun, Wolfram Burgard and Dieter Fox, Probabilistic Robotics, Cambridge, MA, MIT Press, 2005.
- [3] John N. Holmes, Wendy J. Holmes, W.J. Holmes, Speech Synthesis and Recognition, Reading, UK, Taylor & Francis Ltd, 2002.
- [4] John Coleman, Introducing Speech and Language Processing, Cambridge, UK, Cambridge University Press, 2005.
- [5] Judea Pearl, Probabilistic Reasoning in Intelligent Systems, San Mateo, Morgan Kaufmann, 1988.
- [6] Ben Goertzel, Moshe Looks, Cassio Pennachin, “Novamente: An Integrative Architecture for Artificial General Intelligence,” Proceedings of AAAI Symposium on Achieving Human-Level Intelligence through Integrated Systems and Research, Washington DC, August 2004.
- [7] Joseph Y. Halpern, Reasoning About Uncertainty, Cambridge, MA, MIT Press, 2003.
- [8] progic07: The Third Workshop on Combining Probability and Logic
- [9] Ben Goertzel, Matthew Iklé, Izabela Goertzel, Ari Heljakkala, Probabilistic Logic Networks: A Comprehensive Conceptual, Mathematical and Computational Framework for Uncertain Inference, Berlin; New York, Springer Verlag, 2008.
- [10] Matthew Iklé, Ben Goertzel, Izabela Goertzel, “Indefinite Probabilities for General Intelligence,” Advances in Artificial General Intelligence, IOS Press, 2007.
- [11] Hans-Jürgen Zimmerman, Fuzzy Set Theory - and its Applications, 4th ed., Berlin; New York, Springer Verlag, 2007.
- [12] Lofti A. Zadeh, George J. Klir, Bo Yuan (eds.), Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems: Selected Papers by Lofti A. Zadeh, Singapore; River Edge, N.J.: World Scientific Press, 1996.
- [13] Aron Culotta, Andrew McCallum, Practical Markov Logic Containing First-order Quantifiers with Application to Identity Uncertainty.. HLT Workshop on Computationally Hard Problems and Joint Inference in Speech and Language Processing, 2006.
- [14] Peter Walley, Statistical reasoning with imprecise probabilities, London ; New York : Chapman and Hall, 1991.
- [15] Nir Friedman, Joseph Y. Halpern, and Daphne Koller, “First-order Conditional Logic Revisited,” AAAI-96, 1996.

Comirit: Commonsense Reasoning by Integrating Simulation and Logic

Benjamin JOHNSTON and Mary-Anne WILLIAMS

Faculty of Information Technology, University of Technology, Sydney, Australia

Abstract. Rich computer simulations or quantitative models can enable an agent to realistically predict real-world behavior with precision and performance that is difficult to emulate in logical formalisms. Unfortunately, such simulations lack the deductive flexibility of techniques such as formal logics and so do not find natural application in the deductive machinery of commonsense or general purpose reasoning systems. This dilemma can, however, be resolved via a hybrid architecture that combines tableaux-based reasoning with a framework for generic simulation based on the concept of ‘molecular’ models. This combination exploits the complementary strengths of logic and simulation, allowing an agent to build and reason with automatically constructed simulations in a problem-sensitive manner.

Keywords. Commonsense reasoning, simulation, tableaux methods

Introduction

Comirit is the name of both a software system and a corresponding research project that seeks to enhance robotic and software agents with commonsense awareness and reasoning capabilities. The *Comirit* project is exploring the feasibility of rapidly constructing intelligent systems from existing technologies—that is, *Comirit* is a response to the challenge of engineering commonsense-enabled systems from scratch, on minimal budgets and in short time-frames. The objective of this paper is to introduce the core integrative architecture of *Comirit*, in the context of the inspiration of the architecture: the problem of integrating simulation and logic. While the objective of *Comirit* is nominally commonsense reasoning, the project has clear connections to artificial general intelligence and this project is intended as the first in a series of efforts aiming to pragmatically engineer, within a resource-constrained context, systems that are capable of increasingly more general reasoning and learning.

We see an opportunity in identifying well-established and robust techniques within computer science and integrating these in novel combinations that, in harmony, afford powerful new modes of reasoning. In particular, our early efforts have primarily focused on the compatibility of simulation and logical reasoning:

1. While simulations and computer games do not, in themselves, demonstrate commonsense intelligence or enable complex reasoning, they do provide extremely rich and realistic models of ‘commonsense’ scenarios and behavior.
2. Computable expressive logics, in contrast, allow for flexible and powerful modes of reasoning, but lack the rich models of commonsense situations often found in simulations.

The intent of combining these two methods is to unite the flexibility and power of expressive logics with the richness of simulation, and ideally to do so in a way that remains open

to later inclusion of other forms of reasoning.

It turns out that tableaux approaches to automated reasoning provide an effective basis for successful integration of logic and simulation. Tableaux systems are constructive methods for finding contradictions—they attempt to find possible models or worlds in which a logical formula is inconsistent. It is these worlds that form the crux of integration: rather than merely limiting the tableaux search to worlds that lead to logical inconsistency, these possible worlds can also be tested for inconsistency under simulation (or, in fact, inconsistency under a wide range of techniques).

Unfortunately, the technicalities of implementing such an integrated system can overwhelm the simplicity of this idea. In this paper we describe a strategy for managing this complexity: we generalize the basic principles of tableaux systems so that a branch of the tableaux can contain not just logical terms but also simulation data, notes, functions and even the tableaux expansion rules.

In the following two sections (Section 1 and Section 2), we comprehensively motivate the architecture and briefly review the preliminaries. We then describe the conceptual details of the architecture in Section 3. In Section 4 we describe our experiences with the prototypes that we have built, concluding with a brief discussion of future directions in Section 5.

1. Background

While our objective is commonsense intelligence, we do not intend to belabor the philosophical question of what precisely is meant by ‘commonsense’ or ‘intelligence’. Generally speaking, our intent is to develop software and robotic agents with sufficient know-how to appropriately respond to novel problems caused by the open-ended constraints of the real-world. That is, our emphasis is not on capturing commonly known factual knowledge (“Who is the Queen of England?”), but to create systems with real-world ‘know-how’ (“How can I safely rescue this person?”).

In lieu of a rigorous definition of commonsense intelligence, we make use of benchmark problems and analyze system performance in applied situations to evaluate and motivate our work. Morgenstern [1] provides a range of commonsense benchmark problems, contributed by many researchers, that involve naïve or commonsense knowledge about matters including planning, physics and psychology. While many of these ‘challenge problems’ require general purpose problem solving skills, they are presented within restricted domains and with only moderate complexity such that they are useful for benchmarking formalisms, theories and components (as opposed to entire systems). Entire systems may be benchmarked in realistic open-ended scenarios such as competitions like RoboCup Rescue, RoboCup @ Home and the DARPA Grand Challenges. Physical and spatial reasoning play a significant part in all of these problems, however such problems also include aspects of naïve human psychology, economics, game theory, agent behavior, planning and potentially general purpose problem solving. Existing approaches to these benchmarks and challenges tend to rely on significant human engineering: either in the form of large scale manual knowledge elicitation (*e.g.*, [2,3]) or by manually engineering implicit (and task specific) commonsense ‘know-how’ into the low-level faculties of the robot architecture (*e.g.*, [4]). The effort required and the brittleness of these existing approaches is unacceptable for our purposes.

We observe, however, that modern simulations—computer games, computer animations and virtual worlds—have increasingly detailed, life-like environments that sometimes resemble the very situations described in benchmark problems. While simulations

are currently difficult to directly exploit as a resource for commonsense reasoning, they do present a rich resource of implicit commonsense ‘know-how’. Some projects have attempted to exploit this resource indirectly [5]—placing an agent within a simulated environment to explore and learn about human settings free of the wear, cost, time and concurrency constraints of the real world. Such approaches are interesting, but are limited by progress in machine learning and the ability to automatically represent, generalize and specialize knowledge acquired from simulated experiences. In contrast, our initial objective is to *directly* exploit simulation as a representation and reasoning mechanism (rather than merely as a training environment).

Of course, with (typically) only a forward mode of reasoning and many critical limitations, simulations are not immediately useful as a mechanism for commonsense reasoning. However, in combination with a suitable automatic reasoning system for an expressive logic, these limitations can be avoided. Proof search facilities can be used to reverse the natural ‘arrow of time’ by postulating possible causes and then simulating to test whether the outcomes match observations. Such proof search facilities can also be expanded to generate the necessary detail to allow partially defined and unground objects to be numerically simulated, and search facilities may even be used to manage the process of converting symbolic scene descriptions into numerical simulations. Furthermore, the reasoning system can solve those sub-problems that are poorly suited to encoding in simulations, such as abstract logical deduction and random-access recall of simple factual data.

In principle, if we are integrating simulation and logic, we have many options for the choice of underlying technologies; the selection of which can have a dramatic influence on the capabilities and ease of integration of the complete hybrid system. Although it is entirely possible (and effective) to simultaneously use a range of simulation engines in the pluggable hybrid architecture we are proposing in this paper, our preference is for a single simulation platform with sufficient flexibility to model the vast range of scenarios that an agent may encounter. The Slick architecture [6]—a ‘molecular’ or ‘ball-and-stick’ approach to simulating a wide range of physical and non-physical phenomena—suits this criteria and has, in fact, been designed specifically for commonsense reasoning. Similarly, successful integration requires a reasoning mechanism that is efficient and flexible and that provides suitable ‘scaffolding’ upon which other methodologies may be added. Tableaux based reasoning systems are ideal in this regard: they have demonstrated efficiency as the basis of modern Semantic Web reasoners [7], and their search strategy is based on the construction of counter-models or counter-worlds to which simulations can be attached. The following section includes a brief review of both Slick simulation and tableaux based reasoning in the context of hybrid commonsense reasoning.

2. Preliminaries

2.1. Slick Simulation

The Slick architecture [6] is a general purpose approach to simulation designed for commonsense reasoning systems (for related approaches see [8,9]). Slick is designed for simulating a wide range of phenomena including domains as diverse as physical solids and liquids, naïve psychology and economic behavior. In Slick, the state of a simulation is represented as a hypergraph in which every vertex and hyperedge is annotated with a frame. A set of functions perform iterative stepwise update to the state of the simulation during

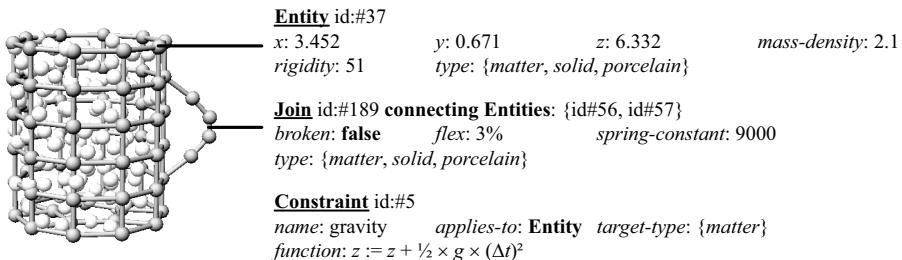


Figure 1. Sample Entity, Join and Constraint illustrating the frame-like data structures of a Slick simulation. each tick of a global, adaptive¹ clock. With appropriate update functions and annotations, Slick can be applied to any environment where *global* behavior is governed by or can be predicted by *local* laws or rationality assumptions. For example, in the case of physical environments: the hypergraph is structured to approximate the shape of physical objects; annotations are used to describe the local physical properties of the object (the position of vertex, local weight density, local temperature, appearance, *etc.*); and update rules correspond to discrete-time variants of the laws of Newtonian mechanics. Consider Figure 1 as an example of how Slick might be used to represent a cup of coffee.

Note that this method of simulation is characteristically non-symbolic. We do not need to specify how a cup of coffee behaves, but derive its behavior from the interactions and forces that occur over the simplified ‘molecular’ structure. That is, macroscopic properties and behaviors emerge from simple microscopic update rules, simplifying the knowledge engineering process and maximizing the generality of the technique.

We have previously demonstrated [6] how this simple architecture can be implemented in a concrete system and applied to established benchmark problems such as the Egg Cracking problem [10]. A concrete implementation of the Slick architecture includes the following three critical classes/types:

1. **Entity**. An annotated vertex with operations for vertex-specific parameters to be stored and retrieved by attribute name.
2. **Join**. An annotated hyperedge (with a set of end vertices) with operations for hyperedge-specific parameters to be stored and retrieved by attribute name.
3. **Constraint**. A stepwise simulation update function that queries the current simulation hypergraph and updates the annotations, in addition to possibly forking the simulation or invalidating the simulation if either multiple or no valid future-states exist.

The Slick architecture, as originally proposed, also incorporates a control language for instantiating and manipulating simulations, in addition to a database mechanism that stores generic models for instantiation and manages the relationship between abstract symbols and the simulation hyper-graph. These not only enable the integration of simulation and symbolic methods, but allow for simulations to be automatically constructed from purely symbolic queries. We will see in Section 3 that while the database and control language remain essential in a hybrid architecture, they are subsumed and generalized by the use of expressive logics.

2.2. Tableaux Reasoning

The method of analytic tableaux is an approach to automatically proving (or disproving) the truth of a logical statement by searching for models or worlds in which the negation of the logical statement is satisfiable. The technique is over 50 years old but has experienced

¹ Clock rate is variable and adapts to ensure numerical precision remains within acceptable bounds.

Table 1. Basic tableaux rules.

Rule	Condition	Action	Explanation
R1a:	$\{A \wedge B\}$	$\rightarrow \{A\}$	(extend the branch)
R1b:	$\{A \wedge B\}$	$\rightarrow \{B\}$	(extend the branch)
R2:	$\{A \vee B\}$	$\rightarrow \{A\}, \{B\}$	(fork into two child branches)
R3:	$\{A, \neg A\}$	$\rightarrow *$	(close the branch due to contradiction)

a recent surge in interest because of its applicability to modal logics, description logics and Semantic Web reasoning. A thorough overview of the method and applications to reasoning with propositional, first order, modal, description and other logics can be found in D'Agostino *et al.*'s (eds) handbook [11] or other standard references for formal computer science. For convenience, we present a brief review of the technique for the propositional case below.

A tableaux system is a method of proof by contradiction: it proves validity by showing that the negation of a formula is unsatisfiable. The method works by updating a tree structure whose nodes are labeled with logical formulae. The algorithm starts with a negated formula as the root of a tree and repetitively applies a set of update rules that close or expand branches of the tree as appropriate. The basic algorithm is simple: when a branch contains a disjunction, the branch is forked into two child branches corresponding to each of the disjuncts; when a branch contains a conjunction, each conjunct is added to the leaf of the branch. The effect is that a formula is converted into a tree where the parent-child relationship can be read conjunctively and the sibling relationship can be read disjunctively. Because the parent-child relationship is read conjunctively, we look along branches (paths from the root to a leaf along the ancestor/descendent axis) for contradiction. If every branch contains a contradiction, then there is no consistent counter-model and therefore the original un-negated formula must be valid. The update rules of this basic algorithm are summarized in Table 1.

The following example illustrates how the tableaux method is used to reason that $((a \vee b) \wedge \neg a) \Rightarrow b$ is universally valid.

To show that $((a \vee b) \wedge \neg a) \Rightarrow b$ is valid, we negate it: $\neg((a \vee b) \wedge \neg a) \Rightarrow b$, simplify to negation normal form: $(a \vee b) \wedge \neg a \wedge \neg b$, and then place it at the root of a tree. We then attempt to show that this negated formula at the root of the tree is unsatisfiable by applying the basic tableaux rules until all branches can be closed. Figures 2a–2c illustrate the results of the iterative rule application.

A contradiction can be seen along the left-hand branch of the completed tableau: node 5 is an ancestor of node 6 so should be read conjunctively, however $\neg a$ and a obviously

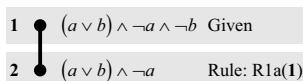


Figure 2a. Tableau after the first step (applying rule R1a on the first node)

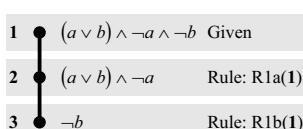


Figure 2b. Tableau after the second step (applying rule R1b on the second node).

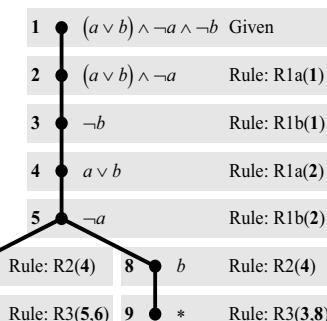


Figure 2c. Completed tableau for $((a \vee b) \wedge \neg a) \Rightarrow b$ (reached after eight steps).

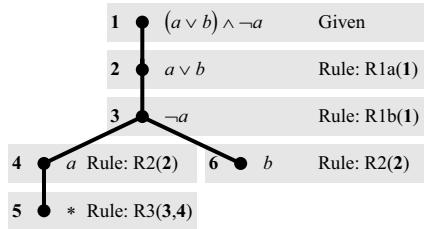


Figure 3. Completed tableau for $(a \vee b) \Rightarrow a$.

form a contradiction. Similarly, node 3 is an ancestor of node 8 and when read conjunctively, trivially forms a contradiction. All branches are contradictory and closed, therefore the original formula is valid.

Similarly, consider the effect if we begin with a formula that is not valid, such as: $(a \vee b) \Rightarrow a$. We first negate it: $\neg((a \vee b) \Rightarrow a) \equiv (a \vee b) \wedge \neg a$, and apply the tableaux rules, then we obtain a completed tableau per Figure 3.

A contradiction can be seen along the left-hand branch: node 3 (a) is an ancestor of node 4 ($\neg a$) so they should be read conjunctively, but are in contradiction. The right-hand branch, however, remains open with neither contradictions nor the possibility of applying a rule (without repeating). In fact, the right-hand branch (when read conjunctively) is a counter model for our original formula: $(a \vee b) \Rightarrow a$ does not hold when we have $\neg a$ and b . The tableaux method is an efficient way of searching for such counter models.

We have assumed conversion into negation normal form (NNF) and the three operators \wedge , \vee and \neg . However, by extending the tableaux system with rules for NNF conversion, it is possible to accept any formula of propositional logic. Through the addition of yet further rules (and additional types of structures), a range of other operators and logics can be supported [11].

We have provided only a brief overview, ignoring a range of issues concerning rule selection, search strategy, heuristics and optimizations, quantification and modalities, cycles, managing repeated rule application, completeness, correctness and the complexities of undecidable logics (such as first order logic). While these matters are highly significant in any domain, including applications for commonsense reasoning, we have not yet had the opportunity to comprehensively consider every issue but have designed a system that through configurability (and modularity) of tableaux rules, meta-strategies and data-types remains agnostic to the particulars of a given tableaux algorithm. However, the critical observation here is that tableaux systems can be made very efficient, can support a wide range of logics and, as we will see in the following section, elegantly interface with other forms of ‘reasoning’ such as simulation.

3. Integration

In combining simulation and logic, we hope to create a system greater than merely the sum of its two parts. In combination, the two have the capability for a powerful union: logic unlocks the full potential of the implicit ‘know-how’ in simulations, and simulation dramatically enhances the capabilities and real-life applicability of logic. However, in creating a hybrid system we must carefully address the mismatch between paradigms and the inherent increase in complexity that the integration introduces. Furthermore, our goal is to create an open-ended system that can admit other technologies and thereby develop sophistication over time. Aside from our primary motivation of creating systems with greater commonsense awareness, we are therefore guided by three important design

objectives: cohesion, simplicity and open-endedness. Cohesion is achieved by using a well defined interface between tableau reasoning and simulation, described in Section 3.1. This interface is implemented using highly regularized data-structures, described in Section 3.2, that maximize simplicity and open-endedness.

3.1. Tableaux-Simulation Interface

Clear semantics assist in maximizing the cohesion in the integration of simulation and tableaux based reasoning. Without a clear and unifying abstraction, the mapping between paradigms can be highly convoluted, requiring complex ad-hoc code to monitor, analyze and translate data-structures and data-updates between representations. Instead, a single, well-defined coupling is preferred, such as with the concept of possible worlds.

Both simulation and tableaux reasoning can be seen as processes concerned with dividing and subsequently ‘closing’ the space of possible worlds. In tableaux-based reasoning systems, broad regions of the space of possible worlds are divided and closed by restrictions with logical prepositions. Simulations, in contrast, work only on narrowly defined spaces of possible worlds, but also ultimately divide and ‘close’. This correspondence forms the basis of our integration strategy, and is more clearly illustrated by way of example. Consider the following problem:

A commonsense enabled real-estate matching system knows that John values his time but doesn't have a car: he cycles or catches a bus depending on the weather. A convenient home is less than 15 minutes by bus (60km/hour) and by bicycle (12km/hour) from his workplace. We want to know whether a home 3km from work or 4km by the bus's circuitous route is convenient.

We assume a hybrid system, constructed from simplified components:

- A tableaux reasoner using rules for unquantified predicate logic, equality ($=$), simple relations of order ($<$, \leq) and a rule for expanding a predicate from its definition (\equiv).
- A simple numerical ‘simulation’ of the process of commuting to work via different mechanisms. For illustrative purposes we use the highly simplified and abstracted simulation of Figure 4.

In the context of this system, we might then encode the sample problem as the logical formula $(bus \vee bike) \Rightarrow conv$, where $bus = speed=60 \wedge distance=4$, where $bike = speed=12 \wedge distance=3$ and where $conv = duration \leq 15$.

We negate the formula and convert it to negation normal form: $(bus \vee bike) \wedge \neg conv$, and apply the tableaux rules until no more rules can be applied (*i.e.*, the tableau has ‘stalled’), resulting in the tableau of Figure 5a. Since open branches remain, we attempt simulation: each branch is read as a narrowly defined space of possible worlds, and the simulation engine is accordingly invoked on each branch. Doing so expands the tableaux with the output of the simulation engine per Figure 5b.

Inputs	<i>speed, distance</i>
Outputs	<i>duration</i>
Algorithm	<pre>set x := 60 × distance ÷ speed return x as duration</pre>

Figure 4. Simplified simulation algorithm. Note that while highly simplified, this algorithm has similar constraints to real-life simulations: the inputs are assumed to be numerical and fully specified; and the algorithm can only be used in the ‘forward’ direction (that is, it cannot be directly used to compute speed, given the duration and distance).

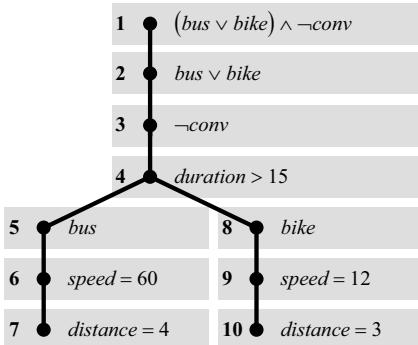


Figure 5a. Stalled tableau

Note that nodes 4, 6, 7, 9 and 10 were created by expanding (from definition) $\neg\text{conv}$, bus and bike .

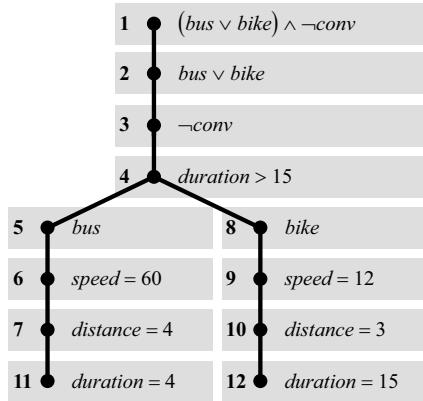


Figure 5b. Tableau after simulation.

Finally, we can revert back to the tableaux rules, and close both branches using the contradictions that occur with *duration* in nodes 4 (*duration* > 15) and 7 (*duration* = 4) in the left-hand branch and in nodes 4 (*duration* > 15) and 7 (*duration* = 15) in the right-hand branch. Because all branches can be closed, we have proven the original query: $(\text{bus} \vee \text{bike}) \Rightarrow \text{conv}$, and have therefore shown the house to be ‘convenient’.

The general execution strategy of the *Comirit* hybrid system follows exactly this process of alternating tableaux rule application and simulation. However, practical complexities are introduced by the non-determinism that we have left implicit. It is often possible to apply both simulation and tableaux rules at any given point in time, and there may in fact be multiple applicable tableaux rules for the tableaux system. Sophisticated heuristics may be developed to guide the execution strategy—our research prototypes generate simple cost-estimates and select based on a greedy cheapest-first strategy, but more mature techniques such as a measure of expected benefit could be used. In the following section we outline the generic architecture which allows for pluggable replacement of selection strategies.

Integrating Slick simulation into this framework presents its own challenges. A Slick simulation has potentially millions of outputs—the value of every attribute of every Entity or Join at every point in time—including these as logical terms in the tableau would incur significant computational and memory costs for little benefit. Instead, additional *linking Constraints* are added to the Slick simulation to manage the connection between symbolic and ‘molecular’ attributes of objects: at each tick of the clock, they calculate the state of an object in aggregate over its ‘molecular’ representation and generate corresponding abstract symbolic terms for inclusion in the tableau. For example, an ‘object broken’ *linking Constraint* might report that an object is broken at a given point in time if any of its Joins have ‘snapped’ at that time. Such constraints can be added to every simulation, however for further performance gains, it suffices to use a simple strategy of only including the *linking Constraints* if they appear to be relevant (that is, the Constraint is only active if its output symbols are referenced by logical terms in the tableau). Such performance optimizations are valid because a simulation can be restarted and executed multiple times to generate necessary values for the tableau if a missing Constraint is later deemed to be relevant.

Finally, tableaux methods are not only applicable to true/false theorem proving—they may be used for query answering. By allowing unification during rule application and unground Prolog-style variables, a tableau can generate output in the form of variable bindings. Syntactically, this is achieved by including a query term in the right-hand side of an implication. For example, if we have a wff, $F(x)$ that constrains the value of x , we can discover the legal values of x by posing a query $F(x) \Rightarrow x = \mathbf{X}$, where \mathbf{X} is an unground

Prolog-style variable. When this formula is negated and converted into negation normal form, the tableau will contain query term $x \neq \mathbf{X}$ that will allow the branch to be closed by unification of \mathbf{X} in the contradiction rule with the true assignment for x .

3.2. Data-Structure Regularity

Combining the two reasoning strategies in a flexible hybrid system requires a large range of data structures and therefore introduces the data management challenge of maintaining internal and mutual consistency. A selection of some of these data structures follow:

Deductive Data Structures. Logical terms, tableaux, Slick molecular representations, database/external-procedure backed predicates.

Strategic Data Structures. Heuristic information and scores, search queues, work queues.

Supportive Data Structures. Cached data, indexes.

Configuration Data. Currently active heuristics, tableau rules, simulation ‘laws’, meta-strategy.

A systematic approach is required, and this can be found by applying object oriented design principles and through the unification of all reasoner state into a single repository.

We generalize the unit of integration—a space of worlds—into a collection object. This collection, known as a **WorldSpace**, corresponds to both a branch of a tableau and an instance of a simulation. A **WorldSpace** has three major operations—division (forking), expansion (addition) and closure—corresponding to the tableau operations for disjunction, conjunction and contradiction. Aside from meta-strategy, *all* state and configuration is stored within a **WorldSpace**.

A set of **WorldSpaces** are grouped into a collection known as a **Problem**. Each query is initially represented by a single **Problem** with a single **WorldSpace** containing the query formula and configuration. The system then uses tableaux rules and simulation to reason about the **WorldSpace**; dividing, expanding and closing until all **WorldSpaces** are closed or no further progress can be made. Aside from a list of current **WorldSpaces**, the only system state stored in a **Problem** relates to meta-strategy: the processes of selecting a focus **WorldSpace** and interpreting the final state of the **Problem**. While focus selection has little impact on the theoretical capabilities of the hybrid system, it has great computational significance. Focus selection drives the high-level search across **WorldSpaces**. For example, a stack based focus selection strategy results in depth first search, a queue based focus selection results in breadth first search, and a priority queue may be used for heuristic, best-first or A*-style searching.

The key to minimizing the complexity of the system, and yet allowing significant modularity and extensibility lies in the regularization of all state and data. Rather than storing tableau rules, heuristic scores, simulation configuration and simulation state in separate linked data structures, they are all stored as terms in the tableau. That is, a tableau branch (a **WorldSpace**) can contain not only logical terms, but a wide range of objects all implemented as subclasses of the class **Node**. Such objects include: simulation objects such as **Entity**, **Join** and **Constraint**; logical objects such as **Term** and **Binding**, dynamic objects such as **Function** and **Task**; and book-keeping objects such as **Note**. In this setting, the tableaux algorithm is generalized so that in each step a branch is searched for not just matching logical terms, but also the appropriate rules to apply. In fact, the tableaux algorithm is itself stored within a **WorldSpace** as a dynamic object—as a **Function**.

Reasoning within a given **WorldSpace** follows a simple interpretation process. When a **Node** is added to a **WorldSpace**, the **WorldSpace** is searched for **Functions** with matching parameter lists. These **Functions** are then invoked to either perform triv-

ial state updates or to generate **Task** objects that are in turn added to the **WorldSpace**. Because **Task** objects also extend **Node**, the creation of a **Task** can result in a cascade of **Function** calls to perform other minor state updates (such as setting priority) or create new **Tasks**. When focus is eventually given to a **WorldSpace**, a **Task** (or set of **Tasks**) is chosen from the **WorldSpace** based on the priority of the **Task**, and subsequently executed. The typical execution process is illustrated in Figure 6.

A **WorldSpace** begins as an empty collection. It is initialized by adding (one-by-one) an appropriate set of **Functions** for reasoning. The query term is then added to the collection as a **Term** (extending **Node**), resulting in a cascade of new **Tasks**, that drive the division, expansion and closure of the **WorldSpace**, the creation of new **Nodes** and in turn the ongoing creation of new **Tasks**. When all **Tasks** have been exhausted, the system attempts to execute any remaining ‘default’ **Functions** (*i.e.*, parameterless functions), and if they in turn are unable to make progress, the branch is deemed to have stalled (*i.e.*, no further progress can be made).

Within this architecture, tableau based reasoning is implemented by a **TableauxReasonerFunction** that responds to logical **Terms**. When a **Term** is added, it is deconstructed and an appropriate division/extension action is chosen and created as a new **Task**.

Simulation-based reasoning is made possible by a set of **Functions** that recognize **Terms** that constrain the state of a simulation. If, for example, a **Term** stating that *x is a Coffee-Mug* is inserted into a **WorldSpace**, then a **SimulationConstructorFunction** will expand the **WorldSpace** with new **Entitys** and **Joins** appropriately describing the shape of a *Coffee-Mug*. When no further progress can be made using other reasoning methods, a **SimulatorFunction**, implemented as a ‘default’ **Function** is executed, creating a new **Task** that will apply every **Constraint** object (*i.e.*, every physical law) in the **WorldSpace** to every **Entity** and **Join** in the **WorldSpace**. While we have not done so, other forms of simulation can be integrated using an identical process.

A range of other reasoning modes are also possible. Some of the modes that we are currently using are described below:

- Heuristics and prioritization are implemented by **Functions** that compute a cost or expected benefit and update the priority of **Tasks** when either the **Task** is added (*i.e.*, before they are executed) or in response to the addition of other kinds of objects to the **WorldSpace**.
- Methods of informal default reasoning are implemented in two ways: as **Functions** that generate plausible extensions in response to the addition of logical terms; and ‘default’ (parameter-less) **Functions** that generate assumptions

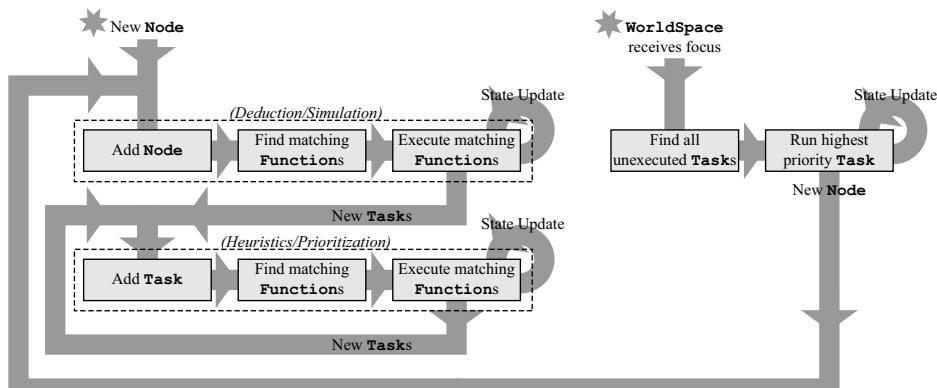


Figure 6. Typical execution/interpretation process for a **WorldSpace**.

when ordinary logical deduction has stalled.

- Search *within* a problem space is implemented by ‘default’ **Functions** that generate **Tasks** to divide a **WorldSpace** into a covering set of mutually exclusive conditions.
- Database-driven predicates and ‘abbreviated’ expressions are implemented as Functions that respond to the addition of a **Term**, invoking external routines that test the new **Term** or generate expanded forms for addition into the **WorldSpace**.

4. Results and Discussion

This architecture is the culmination of our experiences in integrating simulation and logical reasoning in novel combinations; and the only architecture in our experience that so elegantly achieves cohesion, simplicity and open-endedness. We are currently exploring and implementing different logics, simulation laws and heuristics to maximize the usefulness of the system.

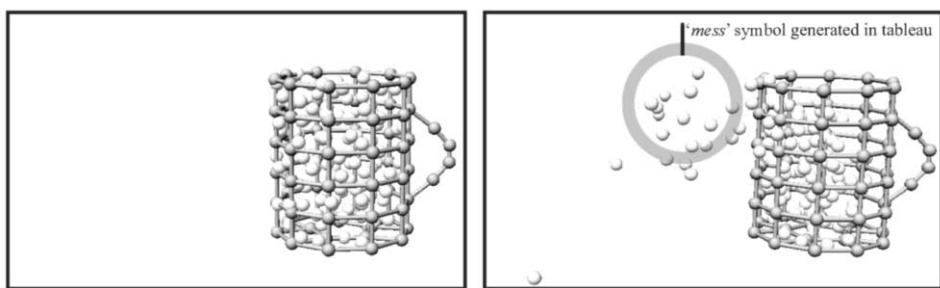
The best way to illustrate the full power of this approach is through example. Consider a simple benchmark-style challenge of a reasoning whether it is better to use a slow action or fast action to move a cup of coffee. We can pose the problem as a logical formula (using \square as the modal ‘always’ operator):

$$\text{problem} \equiv \left(\begin{array}{l} (\text{action} = \text{slow-move}(x) \vee \text{action} = \text{fast-move}(x)) \\ \wedge x \text{ is a Mug} \wedge x \text{ contains Coffee} \wedge y \text{ is a Table} \wedge x \text{ on } y \\ \wedge \text{standard-physics} \wedge \text{can-assume-defaults} \end{array} \right) \Rightarrow \square(\neg\text{mess} \wedge \neg\text{damage})$$

Alternately, we might pose the problem as a query to discover an appropriate action: (*i.e.*, $\text{problem} \Rightarrow \text{action} = \mathbf{A}$).

Given this query, the system builds a tableau with two branches corresponding to the two possible actions. In both branches, the $x \text{ is a Mug}$ and $x \text{ contains Coffee}$ are expanded into a set of **Entitys** and **Joins**; the *standard-physics* term is expanded into a set of **Constraints** corresponding to the laws of physics; and the *can-assume-defaults* is expanded into **Functions** that generate feasible placements, in 3D space, for objects such as the *Table* and the *Mug*. A 3D rendering of the mid-action state of the two branches can be seen in Figure 7.

In the right-hand branch we have a contradiction caused by the *mess* generated in the simulation and the $\neg\text{mess}$ constraint of our original query. The conclusion that our hybrid reasoner draws is that the preferred action is to use the *slow-move* action on the cup of coffee.



While this example is somewhat contrived, a great deal of generality is evident. Little modification is required to query the feasibility of different kinds of actions, different materials or liquids or even the behaviors of entirely novel objects. The advantage of simulation is that it is not necessary to perform comprehensive knowledge elicitation to describe the behavior of every object in every situation; rather, simulation can automatically compute real-world consequences simply from the physical parameters of objects. Combined with the flexibility of logical deduction, this makes for a powerful commonsense reasoning system.

5. Conclusion and Future Work

Though this architecture represents a first step for the *Comirit* project, and remains a long way from true artificial general intelligence, our early results have demonstrated both significant promise as a tool for commonsense reasoning and an enormous capability for extension and modularity. Reasoning in this hybrid architecture combines both the flexibility and power of formal logics with the ease of specification and the implicit commonsense ‘know-how’ of simulations: even in our simple examples, the architecture elegantly and efficiently solves problems that are difficult to express using other methods.

There is ample scope for future development. In the short term, we intend to explore the potential for including new tableau rules for more efficient and different forms of logical deduction, for including entirely new reasoning mechanisms (integrating along the ‘possible worlds’) and for developing new heuristics. Our longer term vision includes the integration of the reasoning and simulation mechanisms with machine learning and robot vision systems, and subsequently deploying the technology in real-world robot systems. Given the modularity of the architecture and the generality of integration via ‘possible worlds’, we believe that the potential is enormous.

References

- [1] Morgenstern, L. & Miller, R. 2006, *The Commonsense Problem Page*, viewed 10 May 2006 <<http://www-formal.stanford.edu/leora/commonsense/>>.
- [2] Barker, K., Chaudhri, V., Chaw, S., Clark, P., Fan, J., Israel, D., Mishra, S., Porter, B., Romero, P., Tecuci, D. & Yeh, P., 2004, ‘A question-answering system for AP chemistry: assessing KR&R technologies’, KR2004.
- [3] Lenat, D.B., Guha, R.V., Pittman, K., Pratt, D. & Shepherd, M. 1990, ‘Cyc - toward Programs with Common-Sense’, *Communications of the ACM*, vol. 33, no. 8, pp. 30-49.
- [4] Bryson, J. 2003, ‘The behaviour-oriented design of modular agent intelligence’ in *Agent Technologies, Infrastructures, Tools and Applications for e-Services*, LNCS 2592/2003.
- [5] Goertzel, B., Heljakka, A., Bugaj, S. & Pennachin, M. 2006, ‘Exploring android developmental psychology in a simulation world’, Proc. of ICCCS-2006.
- [6] Johnston, B. & Williams, M-A. 2007, ‘A generic framework for approximate simulation in common-sense reasoning systems’, 8th Int. Sym. on Logical Formalizations of Commonsense Reasoning.
- [7] Horrocks, I. & Sattler, U. 2001, ‘Ontology reasoning in the SHOQ(D) description logic’, IJCAI 2001.
- [8] Gardin, F. & Meltzer, B. 1989, ‘Analogical representations of naïve physics’, *Artificial Intelligence*, vol. 38, no. 2, pp. 139-159.
- [9] Hoyes, K. 2007, ‘3D simulation: the key to AI’ in Goertzel, B. & Pennachin, C. (eds) *Artificial General Intelligence*, Springer, Berlin.
- [10] Morgenstern, L. 2001, ‘Mid-sized axiomatizations of common-sense problems: a case study in egg-cracking’, *Studia Logica*, vol. 67, no. 3, pp. 333-384.
- [11] D’Agostino, M., Gabbay, D., Hähnle, R. & Posegga J. (eds) 1999, *Handbook of Tableau Methods*, Kluwer, Netherlands.

Learning from Inconsistencies in an Integrated Cognitive Architecture

Kai-Uwe KÜHNBERGER, Peter GEIBEL, Helmar GUST, Ulf KRUMNACK,
Ekaterina OVCHINNIKOVA, Angela SCHWERING, and Tonio WANDMACHER

*University of Osnabrück, Institute of Cognitive Science, Albrechtstr. 28,
49076 Osnabrück, Germany*

Abstract. Whereas symbol-based systems, like deductive reasoning devices, knowledge bases, planning systems, or tools for solving constraint satisfaction problems, presuppose (more or less) the consistency of data and the consistency of results of internal computations, this is far from being plausible in real-world applications, in particular, if we take natural agents into account. Furthermore in complex cognitive systems, that often contain a large number of different modules, inconsistencies can jeopardize the integrity of the whole system. This paper addresses the problem of resolving inconsistencies in hybrid cognitively inspired systems on both levels, in single processing modules and in the overall system. We propose the hybrid architecture I-Cog as a flexible tool, that is explicitly designed to reorganize knowledge constantly and use occurring inconsistencies as a non-classical learning mechanism.

Keywords. Integrated Cognition, Hybrid Systems, Inconsistencies, Learning

Introduction

Natural agents, in particular, humans master a large variety of cognitively important challenges. They can perform a variety of different reasoning tasks, store a large number of different kinds of information and knowledge, they have the ability to learn from noisy and sparse data, and show a remarkable potential of creativity in problem solving. An attempt to model such abilities in its broad range with machines necessarily results in a large number of computing paradigms, specialized modules, competing computations, different representation formalisms etc. Consequently coherence problems of the overall system, as well as inconsistency clashes in the single modules are natural side-effects of such integrated architectures. The problem is even worse if the system is hybrid in nature containing symbolic processing devices and connectionist-inspired modules.

Before focusing on the coherence problem in more detail, let us at first make a detour in discussing some aspects concerning neuro-inspired models and symbolic models for intelligent systems. In artificial intelligence, there is a certain tension between symbolic approaches for modeling higher cognitive abilities and neural approaches for learning patterns in noisy environments. As a matter of fact, the two approaches have different strengths and weaknesses: whereas symbolic theories have problems in learning from noisy data, controlling elementary behaviors of artificial agents, or detecting patterns in

perception input, connectionist systems are not well-suited to perform deduction steps in reasoning processes, to generate plans for a planning agent, or to represent complex data structures. This gap is not only obvious with respect to different application domains, but also with respect to the underlying methodology. Connectionist systems are usually based on analytic tools, whereas most symbolic systems use logical or algebraic theories for realizing computations.

The idea to combine the strengths of the two modeling paradigms in a hybrid architecture is a natural way to cover the complementary applications domains. Nevertheless there are at least two non-trivial problems for hybrid architectures:

- On which level should learning be implemented?
- What are plausible strategies in order to resolve occurring inconsistencies in single modules, as well as in the overall system?

This paper proposes to bring the two mentioned aspects together (consistency problems and learning tasks in integrative architectures) by using occurring inconsistencies in modules of an architecture and in the overall systems as a mechanism for learning. We propose the I-Cog architecture [1,2] as a model for addressing these problems. I-Cog is a hybrid architecture consisting of three modules: An analogy engine (*AE*) as a reasoning device, an ontology rewriting device (*ORD*) as a memory module for coding ontological background knowledge, and a neuro-symbolic learning device (*NSLD*) for learning from noisy data and drawing inferences in underdetermined situations. The overall claim of this paper is that inconsistencies should not be considered solely as a problem for hybrid systems, but rather as a crucial tool to make (cognitive) learning possible in the first place: Inconsistencies make the adaptation of knowledge and the creative establishment of new knowledge necessary and are therefore triggers for learning new facts. We claim that resolving inconsistencies is not only a problem for hybrid systems, but for every realistic system. Therefore strategies to resolve inconsistencies need to be implemented in all modules, not only with respect to the interaction of the involved modules.

The paper has the following structure: in Section 1, we sketch some ideas of the I-Cog architecture. Section 2 discusses exemplarily resolution and learning strategies from occurring inconsistencies in *AE*, *ORD*, and the overall system. Section 3 summarizes related work and Section 4 concludes the paper.

1. The I-Cog Architecture

I-Cog is a hybrid architecture for integrated cognition. In this section, we mention the overall idea very briefly.¹ I-Cog consists out of three main modules:

- An analogy engine (*AE*) is used to cover various forms of classical and non-classical reasoning. This module is based on heuristic-driven theory projection (HDTTP) [3], a mathematically sound theory for computing analogical relations between a target and a source domain. HDTTP is an extension of the theory of anti-unification [4,5]. The analogy engine was applied to compute analogical relations in a variety of domains like metaphoric expressions [3], qualitative physics [6], or in the geometry domain [7].

¹In [1] and [2], the overall system is described more precisely.

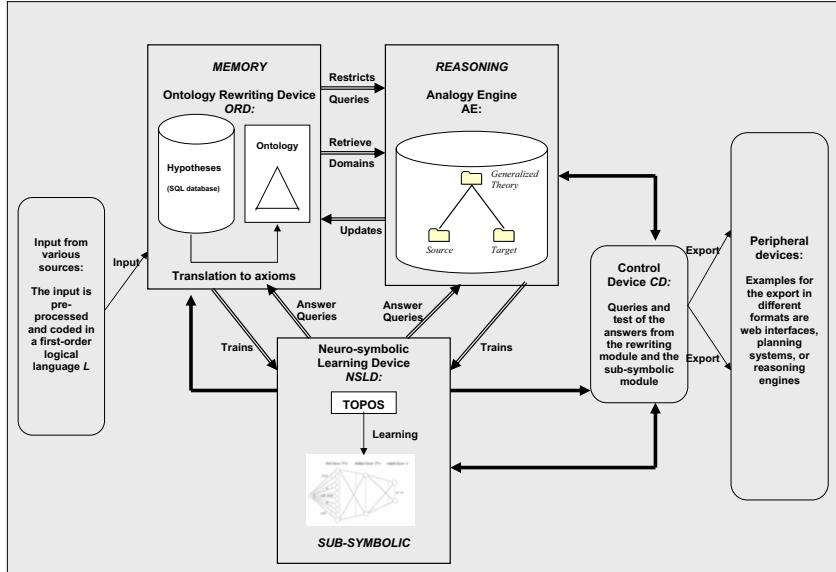


Figure 1. The overall architecture for an integration of the different modules. Whereas the modules *ORD* and *NSLD* are adapting new ontological axioms to an existing ontology, the analogy engine *AE* computes analogical relations based on background knowledge provided by the other two modules. The control device *CD* is intended to choose answers from all three modules.

- The ontology rewriting device (*ORD*) is a knowledge base that is intended to resolve inconsistencies. The system is based on rewriting algorithms designed for several types of description logics [8,9]. The rewriting algorithms were applied to prototypical ontologies for resolving undergeneralization, overgeneralization, and polysemy problems in ontology design. Due to the limited expressive strength of description logics, it is currently not possible to code complex theories in *ORD*, e.g. theories that are based on full first-order logic.
- The neuro-symbolic learning device (*NSLD*) consists of a feedforward neural network that takes as input first-order logical expressions that are transformed into a variable-free representation in a topos [10]. This input is used to learn an approximation of a logical theory, i.e. the input plus an approximation of all possible inferences is learned [11]. The approach was applied to simple and complex logical theories.

An integration based on a non-trivial interaction of the seemingly rather incompatible modules can be achieved as follows: symbolic and sub-symbolic processes can be integrated, because *NSLD* is trained on symbolic data (i.e. on first-order logical expressions) and it is able to learn a model of a given logical theory. Although it is currently not possible to extract symbolic information directly from *NSLD*, a competition of *ORD* and *NSLD* can be implemented by querying both and evaluating their answers (governed by the Control Device *CD*). Additionally, both modules can directly interact with each other, because input from *ORD* can be used for training and querying *NSLD*. In a very similar way, the integration of *AE* and *NSLD* can be justified. Here, the situation is similar, because both modules are operating on a symbolic level (although the expressive

strengths of the respectively underlying knowledge representation formalisms are different). Figure 1 depicts the overall architecture of the system.²

- The input may originate from various sources, e.g. from resources based on structured data, unstructured data, or semi-structured data. The input needs to be available in an appropriate (subset) of a first-order language \mathcal{L} , in order to be in an appropriate format for the other modules. Therefore, ORD generates appropriate logical formulas from hypotheses.
- An important aspect is the interaction of ORD and $NSLD$: on the one hand, ORD trains $NSLD$, on the other ORD queries $NSLD$. Although $NSLD$ can only give an approximate answer in terms of a classification, this can improve the performance of ORD in time-critical situations.
- With respect to the interaction of AE and ORD , ontological knowledge can naturally be used to constrain the computation of possible analogies [12]. Furthermore newly generated analogies can be used to update and therefore rewrite background knowledge [13].
- Similarly to the relation between ORD and $NSLD$, AE is used to train $NSLD$, whereas query answering can be performed in the other direction.
- The control device CD is intended to implement a competition of the feedback of the three modules with respect to queries. Feedback may be in accordance to each other or not. In the second case, a ranking of the corresponding hypotheses is decided by CD (see below).

I-Cog is based on some crucial assumptions: First, the modules are intended to cover a variety of different computing paradigms. For example, the analogy engine AE is covering several forms of classical and non-classical reasoning, like deductive reasoning, inductive reasoning, or vague reasoning. Furthermore, analogy making is a mechanism for getting new (generalized) concepts [14]. Second, all modules are learning devices in the sense that they are dynamically adapting to new information or results of internal processes. Third, the system is crucially hybrid due to the interaction between symbolic and neural modules.

2. Learning from Inconsistencies

2.1. General Remarks

Human learning and memorizing differ in various aspects from classical machine learning algorithms. Whereas humans are able to learn from sparse data, machine learning algorithms usually need a large data sample, in order to produce reliable generalizations. Furthermore, learning new knowledge seems to be strongly based on creative problem solving. Additionally, human memory (i.e. learned knowledge) is not comparable with a fixed knowledge base, but is constantly reorganized. In the I-Cog architecture, we propose that learning is realized decentralized in each module namely by constantly reorganizing the memory content in AE , ORD , and $NSLD$. In other words, learning is a side-effect of other processes and not explicitly based on a single learning mechanism.

²Whereas the three main modules of I-Cog are implemented and evaluated in several scenarios, the I-Cog architecture as a whole is currently not implemented yet.

Inconsistencies are classically connected to logical theories: If for a given set of axioms Γ (relative to a given logical language \mathcal{L}) a formula ϕ can be entailed ($\Gamma \models \phi$) and similarly $\neg\phi$ can be entailed ($\Gamma \models \neg\phi$), then the axiomatic system Γ is inconsistent. In this paper, we are using the term inconsistency rather loosely by discussing some special forms of inconsistent data, that are not necessarily equivalent to logical inconsistency in the sense above. Rather some forms could be better described by incoherent data or incompatible conceptualizations. Here are three simple examples:

- Every analogy establishes a relation between two domains that may be similar in certain aspects, but usually are far from being coherent. For example, analogies in qualitative physics like "Current is the water in the electric circuit." or "Electrons are the planets of the atom." are simply statements that are semantically nonsense: A computation of the conventional meaning of these natural language sentences (relative to a conventional lexicon) would result in *false*, hence they are unsatisfiable and therefore contradictions. Nevertheless humans can make sense out of these sentences and particularly in teaching situations, students can learn new conceptualizations of a formerly unknown domain by such analogies.
- In automatic ontology generation, it happens quite often that polysemy problems can occur due to the missing disambiguation of different word senses. Whether the word *forest* denotes a collection of plants or an abstract data structure, depends on the context in the text. These two interpretations of *forest* need to be described in the ontology by two different concepts with two different identifiers (e.g. *ForestPlant* and *ForestStructure*).
- Concepts in ontology design are often overgeneralized. The most famous example is *Tweety*, the penguin that is a bird, but cannot fly:

$$\forall x : Bird(x) \rightarrow CanFly(x) \quad \forall x : Penguin(x) \rightarrow Bird(x) \wedge \neg CanFly(x)$$
 Obviously, a contradiction can be derived because "Birds can fly" is too general. This inconsistency can be resolved by introducing a slightly more specific concept like *FlyingBird* (cf. Subsection 2.3).

In the following three subsections, we will give some examples how learning from inconsistencies is realized in I-Cog.

2.2. Learning from Inconsistencies in Analogy Making

Establishing an analogical relation between two different domains can be seen as the task to make incompatible conceptualizations compatible with each other: For example, in the analogy between the atom model and the solar system ("Electrons are the planets of the atom"), two incompatible domains need to be associated with each other. That means that an adaptation process is required, in order to resolve a seemingly incoherent association of information. We will see that not only on this level learning by analogies is based on inconsistencies, but also the analogical reasoning process itself is crucially driven by resolving inconsistencies, and adapting pieces of information.

The framework we use for computing analogical relations is heuristic-driven theory projection (HDTTP) [3]. HDTTP represents the source and target domains by sets of first-order formulas. The corresponding source theory Th_S and target theory Th_T are then generalized using an extension of anti-unification [4]. Here are the key elements of HDTTP:

Source Th_S : Addition	Target Th_T : Multiplication
$\alpha_1 \quad \forall x : add(x, 0) = x$	$\beta_1 \quad \forall x : mult(x, s(0)) = x$
$\alpha_2 \quad \forall x \forall y : add(x, s(y)) = s(add(x, y))$	$\beta_2 \quad \forall x \forall y : mult(x, s(y)) = add(x, mult(x, y))$
Generalized Theory Th_G	
$\gamma_1 \quad \forall x : \mathbf{Op}_1(x, \mathbf{E}) = x$	
$\gamma_2 \quad \forall x \forall y : \mathbf{Op}_1(x, s(y)) = \mathbf{Op}_2(\mathbf{Op}_1(x, y))$	

Table 1. A formalization of addition and multiplication by primitive recursion. The generalized theory Th_G is a simplified but abstract formalization of primitive recursion.

- Two formulas $p_1(a, b)$ and $p_2(a, c)$ can be anti-unified by $P(a, X)$, with substitutions $\Theta_1 = \{P \rightarrow p_1, X \rightarrow b\}$ and $\Theta_2 = \{P \rightarrow p_2, X \rightarrow c\}$.
- A theorem prover allows the re-representation of formulas.
- Whole theories can be generalized, not only single terms or formulas.
- The process is governed by heuristics on various levels.

The idea behind HDTDP is to compute generalizations of (logical or arithmetical) formulas (axioms), in order to establish a generalization of the source and target domain. A heuristic used is the minimization of the complexity of substitutions for generalized pairs of axioms. The generalization process leads to anti-instances that are structural descriptions of the input theories of source and target.

We want to exemplify HDTDP using a simple example without specifying the formal details of the approach.³ Table 1 shows the standard axiomatization of addition and multiplication by primitive recursion (source and target). The generalized theory depicted in Table 1 introduces (existentially quantified) new variables for constants and operators. It is possible to gain the corresponding source theory Th_S and target theory Th_T from the generalized theory Th_G by applying the following substitutions:

$$\begin{aligned} \Theta_1 : \quad & \mathbf{E} \mapsto 0, \mathbf{Op}_1 \mapsto add, \mathbf{Op}_2 \mapsto s \\ \Theta_2 : \quad & \mathbf{E} \mapsto s(0), \mathbf{Op}_1 \mapsto mult, \mathbf{Op}_2 \mapsto \lambda z.add(x, z) \end{aligned}$$

The goal of computing an analogy between source and target is a generalized theory associating pairs of formulas from the source and the target. If the input is given as depicted in Table 1, HDTDP starts with axiom β_1 (heuristics based on complexity), searches a possible candidate in the set $\{\alpha_1, \alpha_2\}$ for anti-unification and finds α_1 as the best candidate (based on a further heuristics minimizing the substitution complexity). The anti-instance γ_1 is straightforwardly constructed. β_2 is chosen next and anti-unified with α_2 in a very similar way.

For a computation of the generalizations γ_1 and γ_2 no inconsistencies occur making backtracking unnecessary. The situation changes, if we change the conceptualization of the target domain slightly: Assume we have the axiom $\forall x : mult(s(0), x) = x$ instead of β_1 (permutation of arguments). Then, a simple anti-unification is not possible, and HDTDP backtracks based on this inconsistency. Provided there is background knowledge available stating that $\forall x : mult(s(0), x) = mult(x, s(0))$ holds, then the internal theorem prover in HDTDP re-represents the input β_1 , in order to anti-unify β_1 and α_1 as above resulting in the generalized theory γ_1 and γ_2 .

³This example is intended to make the mechanisms more transparent, not to give a thorough introduction into HDTDP. The interested reader is referred to [3,15,5] for the theoretical background.

Source Th_S : Addition	Target Th_T : Multiplication
$\alpha_1 \quad \forall x : add(0, x) = x$	$\beta_1 \quad \forall x : mult(0, x) = 0$
$\alpha_2 \quad \forall x \forall y : add(s(y), x) = add(y, s(x))$	$\beta_2 \quad \forall x \forall y : mult(s(y), x) = add(x, mult(y, x))$
Generalized Theory Th_G	
$\gamma_1 \quad \forall x : \mathbf{Op}_1(\mathbf{E}, x) = x$	

Table 2. A formalization of addition and multiplication by a different recursive axiomatization.

Consider a slightly more complicated situation, where the conceptualization is given as depicted in Table 2. The two axioms systems of source and target cannot be anti-unified in a straightforward way. By using the theorem prover, we can derive the following fact (using axioms α_1 and α_2 of the source to derive a fact on the target):

$$mult(s(0), x) = add(x, mult(0, x)) = add(x, 0) = \dots = add(0, x) = x$$

Hence, we can derive $\beta_3 : \forall x : mult(s(0), x) = x$ and the anti-unifier γ_1 can be established with the following substitutions:

$$\begin{aligned} \Theta_1 : \quad \mathbf{E} &\mapsto 0, \quad \mathbf{Op} \mapsto add \\ \Theta_2 : \quad \mathbf{E} &\mapsto s(0), \quad \mathbf{Op} \mapsto mult \end{aligned}$$

The established analogy expresses that addition corresponds to multiplication and that there are (different) unit elements for addition and multiplication. It is important to notice that the generation of abstract knowledge in form of a generalization of source and target is based on occurring inconsistencies between two seemingly incompatible axiomatizations.

Based on the given axiomatization of Table 2 it is not possible to derive a general commutativity law $\forall x \forall y : \mathbf{Op}(x, y) = \mathbf{Op}(y, x)$. But it is possible to extend the coverage of the generalized theory by arbitrary instances of formulas $\mathbf{Op}(s^n(0), s^m(0)) = \mathbf{Op}(s^m(0), s^n(0))$ without changing the relevant substitutions.

Notice that natural extensions of the sketched simple arithmetical theories are possible. Just to mention one case, if we extend elementary arithmetic on natural numbers to elementary arithmetic on rational numbers, additional laws on the source domain can be introduced. For example, the law α_3 , that guarantees the existence of an inverse element, could be added: $\alpha_3 : \forall x \exists y : add(x, y) = 0$. An analogical projection (transfer) of α_3 to the target side would result in a law $\beta_3 : \forall x \exists y : mult(x, y) = 1$, which is obviously a contradiction to β_1 , because one can deduce the inconsistency $\exists y : mult(0, y) = 1$. Clearly β_3 does only hold for rational numbers unequal to 0. An appropriate restriction of the domain for which β_3 holds can fix this problem.⁴ On the level of the generalized theory a new concept *invertible element* can be introduced.

2.3. Learning from Inconsistencies of Ontological Background Knowledge

It is commonly assumed that background knowledge of human agents quite often contains incoherent information. Furthermore, human memory is not fixed, but rather con-

⁴There are many non-trivial examples in mathematics, where some properties do only hold for a subset of objects: Consider, for example, quaternions \mathbb{H} , a non-commutative extension of complex numbers, where only the basis quaternions i, j, k are non-commutative with $ij = k \neq ji = -k$, whereas the restrictions to real number $a = a_1 + 0i + 0j + 0k$ and $b = b_1 + 0i + 0j + 0k$ results in $ab = ba$.

stantly updated in an incremental way. Humans can reorganize their memory content and have the ability to adapt incoherent information on-the-fly. From a more technical perspective this translates into an adaptive knowledge base for artificial systems. In the standard conception a knowledge base is considered to be static, although this is a rather counterintuitive assumption. In the I-Cog architecture, the memory module *ORD* is intended to be highly adaptive, extending incrementally the knowledge base constantly, and – if necessary – adapting it, provided inconsistencies occur.

The resolution of inconsistencies is assessed by a rewriting device on knowledge formalized in description logics (DL), which are the current state-of-the-art standard for coding ontological knowledge. Ontologies usually contain a terminological component and an assertion component. A description logic terminology consists of a set of terminological axioms defining concepts either by formulas of the form $\forall x : C(x) \rightarrow D(x)$ (partial definition) or by $\forall x : C(x) \leftrightarrow D(x)$ (total definition), where C is a concept name and D is a concept description.⁵ Additionally to the terminological component an assertion component contains information about the assignment of particular individuals to concepts and relations from the terminology. Axioms are interpreted model theoretically by an interpretation function mapping concept descriptions to subsets of the domain. A model of an ontology is an interpretation satisfying all axioms. An ontology is inconsistent if it does not have a model.

There are several possibilities why inconsistencies can occur in ontologies. In [16], structural inconsistencies, usage-defined inconsistencies, and logical inconsistencies are distinguished. The last type of inconsistency – potentially caused by dynamic updates of the knowledge base – is of particular interest in our context. We mention some forms of occurring logical inconsistencies that can be handled with *ORD*. One aspect of logical inconsistency problems concerns polysemy: If an ontology is updated automatically, then different concepts may happen to have the same name. Suppose, the concept named *tree* is declared to be a subconcept both of *plant* and of *data structure* (where *plant* and *data structure* are disjoint concepts). Both of these two interpretations of *tree* are correct, but it is still necessary to describe two different concepts in the ontology with different identifiers (e.g. *TreePlant*, *TreeStructure*).

Another important aspect of logical inconsistency problems concerns generalization mistakes and is strongly connected to non-monotonic reasoning, extensively discussed in the relevant AI literature.

Example 1 Assume the following axioms are given:

$$\begin{aligned} \forall x : Bird(x) &\rightarrow CanFly(x) & \forall x : CanFly(x) &\rightarrow CanMove(x) \\ \forall x : Canary(x) &\rightarrow Bird(x) & \forall x : Penguin(x) &\rightarrow Bird(x) \wedge \neg CanFly(x) \end{aligned}$$

In Example 1, the statement “birds can fly” is too general. If an exception occurs (*penguin*), the ontology becomes unsatisfiable, since *penguin* is declared to be a bird, but it cannot fly.

In order to resolve the inconsistency, notice that the definition of the concept *Bird* is overgeneralized. Therefore we need to rewrite it. Nevertheless, we wish to keep as much information as possible in the ontology. Example 2 specifies a solution:

⁵Compare [17] for an exhaustive definition of description logics.

Example 2 Adapted ontology from Example 1:
$$\begin{aligned} \forall x : Bird(x) &\rightarrow CanMove(x) \\ \forall x : CanFly(x) &\rightarrow CanMove(x) \\ \forall x : Canary(x) &\rightarrow FlyingBird(x) \\ \forall x : Penguin(x) &\rightarrow Bird(x) \wedge \neg CanFly(x) \\ \forall x : FlyingBird(x) &\rightarrow Bird(x) \wedge CanFly(x) \end{aligned}$$

In the definition of the concept *Bird* (subsuming the unsatisfiable concept *Penguin*), we want to keep a maximum of information not conflicting with the definition of *Penguin*. Conflicting information is moved to the definition of the new concept *FlyingBird*, which is declared to subsume all former subconcepts of *Bird* (such as *Canary*) except *Penguin*.

An algorithmic solution to the problem is formally described in [8], [18], and [9]. In this framework, ontologies are extended with additional axioms conflicting with the original knowledge base, i.e. given a consistent ontology O (possibly empty) the procedure adds a new axiom A to O . If $O^+ = O \cup \{A\}$ is inconsistent, then the procedure tries to find a polysemy or an overgeneralization and repairs O^+ . First, problematic axioms that cause a contradiction are detected, then the type of the contradiction (polysemy or overgeneralization) are defined, and finally an algorithm repairs the terminology by rewriting parts of the axioms that are responsible for the contradiction. Detected polysemous concepts are renamed and overgeneralized concepts are split into more general and more specific parts.

The sketched rewriting for a constant adaptation process of background knowledge is a first step towards a general theory of dynamification and adaptation of background knowledge. The framework has been developed primarily for text technological applications, but the approach can be extended to a wider range of applications.⁶

2.4. Inconsistencies in the Overall System

As mentioned in the Introduction, a standard problem of hybrid architectures is the problem of how inconsistencies between potential outputs of the different modules can be resolved. Even in the case a competition mechanism is implemented between the modules, in order to accept or reject certain potential outputs of the respective modules, a control device needs to assess these outputs appropriately. A plausible way for an implementation is to use heuristics in the first place to resolve potential conflicts, and on top of it a learning device for updating these heuristics.

In the I-Cog architecture, the control device *CD* is the module that arbitrates between the main modules. *CD* needs to assess possible answers of the three main modules and needs to implement a competition process. First, we exemplify possible situations with respect to *ORD* and *NSLD*. Concerning underdetermined situations, where *ORD* is not able to answer queries, *NSLD* can be used for answering a query.⁷ In such cases, the usage of *NSLD* is clearly preferred by the heuristic. On the other hand, if *ORD* contains

⁶The crucial algorithms for resolving overgeneralization, undergeneralization, and polysemy problems, are implemented and prototypically tested in example domains [9].

⁷Simple examples in which *NSLD* can answer queries in underdetermined situations can be found in [11]: If certain objects of the domain are not sufficiently defined or even not defined at all, a knowledge base or a theorem prover cannot provide any interesting information about such objects. A neural reasoning device like *NSLD* can, because it can give answers to any query.

(or can prove) a particular fact, for example, that a certain subsumption relation between two concepts A and B holds, then this result should be tentatively preferred by CD in comparison to the output of $NSLD$. In cases, where time-critical reactions are necessary and ORD is not able to compute an answer in time, the natural heuristic would be to use $NSLD$ instead. Finally, it could happen that the answers of ORD and $NSLD$ are contradicting each other. In this case, CD cannot base the decision on an *a priori* heuristic. Possible solutions may be either the training of $NSLD$ by the answer of ORD or the implementation of a reinforcement learning mechanism on CD itself. The latter can be used to learn preferred choices of the knowledge modules involved. In both cases, occurring inconsistencies function as triggers for learning.

Very similarly to the interaction between ORD and $NSLD$ we sketch some ideas controlling the interaction between AE and $NSLD$. If the reasoning device AE can prove a fact or can successfully establish an analogical relation between a source and a target domain, AE is clearly preferred in comparison to $NSLD$. In time-critical or underdetermined situations, the answer of $NSLD$ is heuristically preferred. Finally, if contradictions between AE and $NSLD$ occur, the solutions mentioned above can be again applied, namely the implementation of a reinforcement learning mechanism or the training of $NSLD$ by AE .

3. Related Work

Analogical reasoning was discussed in many domains like proportional analogies in string domains [19] and analogies between geometric figures [20]. Further discussions were based on the relation between analogies and metaphors [14] and on analogical problem solving [21]. Concerning underlying methods for modeling analogies algebraic [14], graph-based [22], and similarity-based approaches [23] can be found.

Rewriting systems for knowledge representations are described in [16]. A collection of approaches aiming at resolving inconsistencies in knowledge representation is related to non-monotonicity. Examples are extensions by default sets [24] or by belief-revision processes [25]. A family of approaches is based on tracing techniques for detecting a set of axioms that are responsible for particular ontological contradictions [26], [27].

There is a number of attempts to resolve the gap between symbolic and subsymbolic computations. We just mention some newer approaches: An example to solve the neuro-symbolic integration problem is described in [28] in which a logical deduction operator is approximated by a neural network. Another approach is [29], where category theoretic methods are used for neural constructions. In [30], tractable fragments of predicate logic are learned by connectionist networks.

Recently, some endeavor has been invested to approximate a solution to human-level intelligence. [31] proposes a so-called cognitive substrate in order to reduce higher cognition and the profusion of knowledge to a basis of low computational complexity. Further approaches that resemble the integration idea presented here follow the tradition of cognitive architectures. Examples are the hybrid AMBR/DUAL model [32], which is modeling neuro-symbolic processing and analogical reasoning, the ICARUS architecture [33], which is focusing primarily on learning, or the NARS architecture [34], which is intended for integrating many different types of reasoning and representation formats. Nevertheless the set-up for and the theories behind the involved modules in I-Cog, as

well as the integration idea of a constant reorganization based on resolving inconsistencies in I-Cog fundamentally differ from the mentioned approaches.

4. Conclusions

Standardly it is assumed that inconsistencies in hybrid systems are problematic and a source for difficulties in applications. We used the I-Cog architecture to support the claim that on the contrary, inconsistencies in hybrid cognitive systems should be considered as a cue for (cognitive) learning, and not only as a problem for integrating incoherent information. In I-Cog, all main modules – the analogy engine, the ontology rewriting device, and the neuro-symbolic integration device – are either able to learn from inconsistent information or can deal with vague and noisy input, quite similar to abilities shown by natural agents. In other words, learning should not be realized in a separated module, but should be an integral part of every computation device. Furthermore, inconsistencies do not only occur in the interaction between modules of an architecture, but are an important source for guiding computations and learning on several levels: In reasoning modules for computing various types of inferences, in updates of the knowledge base for adapting background knowledge to new input, and in the interaction of different modules for optimal results in a particular application.

References

- [1] K.-U. Kühnberger, T. Wandmacher, A. Schwering, E. Ovchinnikova, U. Krumnack, H. Gust and P. Geibel, I-Cog: A Computational Framework for Integrated Cognition of Higher Cognitive Abilities, *Proceedings of 6th Mexican International Conference on Artificial Intelligence*, LNAI 4827, Springer (2007) pp. 203–214.
- [2] K.-U. Kühnberger, T. Wandmacher, E. Ovchinnikova, U. Krumnack, H. Gust and P. Geibel, Modeling Human-Level Intelligence by Integrated Cognition in a Hybrid Architecture, in P. Hitzler, T. Roth-Berghofer, S. Rudolph (eds.): *Foundations of Artificial Intelligence (FAInt-07)*, Workshop at KI 2007, CEUR-WS 277 (2007), 1–15.
- [3] H. Gust, K.-U. Kühnberger and U. Schmid, Metaphors and Heuristic-Driven Theory Projection (HDTPT), *Theoretical Computer Science* **354** (2006) 98–117.
- [4] G. Plotkin, A note of inductive generalization, *Machine Intelligence* **5** (1970), 153–163.
- [5] U. Krumnack, A. Schwering, H. Gust, K.-U. Kühnberger, Restricted Higher-Order Anti-Unification for Analogy Making, in *Proceedings of Twenties Australian Joint Conference on Artificial Intelligence*, LNAI 4830, Springer (2007) pp. 273–282.
- [6] H. Gust, K.-U. Kühnberger and U. Schmid, Solving Predictive Analogies with Anti-Unification, in: P. Slezak (eds.): *Proceedings of the Joint International Conference on Cognitive Science*, 2003, pp. 145–150.
- [7] A. Schwering, U. Krumnack, K.-U. Kühnberger and H. Gust, Using Gestalt Principles to Compute Analogies of Geometric Figures, in: D. S. McNamara & J. G. Trafton (Eds.), *Proceedings of the 29th Annual Conference of the Cognitive Science Society*, Cognitive Science Society, 2007, pp. 1485–1490.
- [8] E. Ovchinnikova and K.-U. Kühnberger, Adaptive \mathcal{ALC} -TBox for Extending Terminological Knowledge. In A. Sattar, B. H. Kang (eds.): *Proceedings of the 19th ACS Australian Joint Conference on Artificial Intelligence*, LNAI 4304, Springer, 2006, pp. 1111–1115.
- [9] E. Ovchinnikova, T. Wandmacher and K.-U. Kühnberger, Solving Terminological Inconsistency Problems in Ontology Design. *International Journal of Interoperability in Business Information Systems*, **2(1)** (2007), 65–80.
- [10] R. Goldblatt, *Topoi: The Categorial Analysis of Logic*. Studies in Logic and the Foundations of Mathematics, **98**, North-Holland, Amsterdam (1979).

- [11] H. Gust, K.-U. Kühnberger and P. Geibel, Learning Models of Predicate Logical Theories with Neural Networks based on Topos Theory. In P. Hitzler and B. Hammer (eds.): *Perspectives of Neuro-Symbolic Integration*, Studies in Computational Intelligence (SCI) 77, Springer, (2007) pp. 233–264.
- [12] H. Gust, K.-U. Kühnberger and U. Schmid, Ontological Aspects of Computing Analogies. *Proceedings of the Sixth International Conference on Cognitive Modeling*, Mahwah, NJ: Lawrence Erlbaum, (2004) pp. 350–351.
- [13] H. Gust, K.-U. Kühnberger and U. Schmid, Ontologies as a Cue for the Metaphorical Meaning of Technical Concepts, in A. Schalley, D. Khlebtzov (eds.): *Mental States: Evolution, Function, Nature*, Volume I, John Benjamins Publishing Company, Amsterdam, Philadelphia, (2007) pp. 191–212.
- [14] B. Indurkha, *Metaphor and Cognition*, Dordrecht, The Netherlands, Kluver (1992).
- [15] H. Gust, U. Krummack, K.-U. Kühnberger and A. Schwering, An Approach to the Semantics of Analogical Relations, in S. Vosniadou, D. Kayser, A. Protopapas (eds.): *Proceedings of EuroCogSci 2007*, Lawrence Erlbaum, pp. 640–645.
- [16] P. Haase, F. van Harmelen, Z. Huang, H. Stuckenschmidt and Y. Sure, A framework for handling inconsistency in changing ontologies. *Proc. of the Fourth Internation Semantic Web Conference*, LNCS, Springer (2005).
- [17] F. Baader, D. Calvanese, D. McGuinness, D. Nardi and P. Patel-Schneider (eds.), *Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press (2003).
- [18] E. Ovchinnikova and K.-U. Kühnberger, Automatic Ontology Extension: Resolving Inconsistencies, *GLDV Journal for Computational Linguistics and Language Technology* (to appear).
- [19] D. Hofstadter and The Fluid Analogies Research Group, *Fluid concepts and creative analogies*. New York: Basic Books (1995).
- [20] M. Dastani, Languages of Perception, ILLC Dissertation Series 1998–05, 1998, <http://www.illc.uva.nl/Publications/Dissertations/DS-1998-05.text.ps.gz>.
- [21] J. Anderson and R. Thompson, Use of Analogy in a Production System Architecture, in Vosniadou, Ortony (eds.): *Similarity and analogical reasoning*, Cambridge (1989) 267–297.
- [22] B. Falkenhainer, K. Forbus and D. Gentner, The structure-mapping engine: Algorithm and example, *Artificial Intelligence* 41 (1989) 1–63.
- [23] D. Gentner, The Mechanisms of Analogical Learning, in: S. Vosniadou & A. Ortony (editors): *Similarity and Analogical Reasoning*, New York, Cambridge University Press.
- [24] S. Heymans and D. Vermeir, A Defeasible Ontology Language, In Meersman, R. et al. (eds.): *On the Move to Meaningful Internet Systems, 2002 – Confederated International Conferences: CoopIS, DOA, and ODBASE 2002*, Springer, pp. 1033–1046.
- [25] G. Flouris, D. Plexousakis and G. Antoniou, Updating DLs Using the AGM Theory: A Preliminary Study, in: *Description Logics*, 2006.
- [26] F. Baader and B. Hollunder, Embedding defaults into terminological knowledge representation formalisms. *J. Autom. Reasoning*, 14(1) (1995) 149–180.
- [27] A. Kalyanpur, Debugging and Repair of OWL Ontologies. Ph.D. Dissertation, (2006).
- [28] S. Bader, P. Hitzler, S. Hölldobler and A. Witzel, A Fully Connectionist Model Generator for Covered First-Order Logic Programs. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence* (2007), pp. 666–671.
- [29] M. Healy and T. Caudell, Neural Networks, Knowledge and Cognition: A Mathematical Semantic Model Based upon Category Theory. University of New Mexico, (2004), EECE-TR-04-020.
- [30] A. D'Avila Garcez, K. Broda and D. Gabbay, *Neural-Symbolic Learning Systems. Foundations and Applications*. Springer (2002).
- [31] N. Cassimatis, A Cognitive Substrate for Achieving Human-Level Intelligence, *AI Magazine* 27(2) (2006) 45–56.
- [32] B. Kokinov and A. Petrov, Integrating Memory and Reasoning in Analogy-Making: The AMBR Model, in D. Gentner, K. Holyoak, B. Kokinov (eds.): *The Analogical Mind. Perspectives from Cognitive Science*, Cambridge Mass. (2001).
- [33] P. Langley, Cognitive Architectures and General Intelligent Systems, *AI Magazine*, 27(2) (2006), 33–44.
- [34] P. Wang, *Rigid Flexibility: The Logic of Intelligence*, Springer, 2006.

Extending the Soar Cognitive Architecture

John E. Laird

Division of Computer Science and Engineering, University of Michigan

Abstract. One approach in pursuit of general intelligent agents has been to concentrate on the underlying cognitive architecture, of which Soar is a prime example. In the past, Soar has relied on a minimal number of architectural modules together with purely symbolic representations of knowledge. This paper presents the cognitive architecture approach to general intelligence and the traditional, symbolic Soar architecture. This is followed by major additions to Soar: non-symbolic representations, new learning mechanisms, and long-term memories.

Keywords. Soar, cognitive architecture, reinforcement learning, episodic memory, semantic memory, clustering, mental imagery, emotion.

Introduction

Through the years, there has been substantial evolution and refinement of the Soar architecture [1], with eight major versions between 1982 and 2007. During this evolution, the basic approach of pure symbolic processing, with all long-term knowledge being represented as production rules, was maintained and Soar proved to be a general and flexible architecture for research in cognitive modeling across a wide variety of behavioral and learning phenomena [2]. Soar also proved to be useful for creating knowledge-rich agents that could generate diverse, intelligent behavior in complex, dynamic environments [3, 4].

In spite of these successes, it became clear that Soar was missing some important capabilities that we take for granted in humans, many of which have also been ignored by the larger cognitive architecture community. In response, we have made substantial extensions to Soar, adding new learning mechanisms and long-term memories as well as new forms of non-symbolic processing. The motivation for these extensions is to increase the *functionality* of Soar for creating artificial general intelligent systems, but we also expect that these changes will significantly expand the breadth of human behavior that can be modeled using Soar. Before presenting these extensions, we start with our underlying methodology for understanding and creating artificial generally intelligent systems: cognitive architecture. We then present the traditional version Soar, without the extensions, followed by the extended version of Soar. We conclude with an analysis of this new version of Soar and discussion of future directions.

1. Cognitive Architecture

During the last twenty years, the field of AI has successfully pursued specialized algorithms for specific problems. What distinguishes generally intelligent entities is their ability to solve not just a single problem using a specific method, but the ability to pursue a wide variety of tasks, including novel tasks, using large bodies of diverse knowledge, acquired through experience, in complex, dynamic environments. This leads us to study the fixed infrastructure that supports the acquisition and use of knowledge: the cognitive architecture. A cognitive architecture consists of:

- memories for storing knowledge
- processing units that extract, select, combine, and store knowledge
- languages for representing the knowledge that is stored and processed

Cognitive architectures distinguish between knowledge that is acquired over time and the fixed cognitive architecture that is common across all tasks. One of the most difficult challenges in cognitive architecture design is to create sufficient structure to support initial coherent and purposeful behavior, while at the same time providing sufficient flexibility so that an agent can adapt (via learning) to the specifics of its tasks and environment. Cognitive architectures must embody strong hypotheses about the building blocks of cognition that are shared by all tasks, and how different types of knowledge are learned, encoded, and used, making a cognitive architecture a software implementation of a general theory of intelligence.

The hypothesis behind cognitive architectures such as Soar and ACT-R [5] is that there are useful abstractions and regularities above the level of neurally-based theories. This hypothesis plays out both in longer time scales of modeled behavior and in the symbolic representations of knowledge about the world. A related hypothesis is that the structures and discoveries of the symbolic architectures will be reflected in the neurally-based architectures. This is the approach taken in Neuro-Soar [6] and the ACT-R/Libre [7] hybrid architectures. Probably the most interesting question is whether the extra detail of the neurally-based architectures (and the concurrent additional processing requirements) is necessary to achieve generally intelligent agents, or whether the more abstract architectures sufficiently capture the structures and regularities required for intelligence.

2. Traditional Soar

As shown in Figure 1, up until five years ago, Soar has consisted of a single long-term memory, which is encoded as production rules, and a single short-term memory, which is encoded as a symbolic graph structure so that objects can be represented with properties and relations. Symbolic short-term memory holds the agent's assessment of the current situation derived from perception and via retrieval of knowledge from its long-term memory. Action in an environment occurs through creation of motor commands in a buffer in short-term memory. The decision procedure selects *operators* and detects *impasses*, both of which are described below.

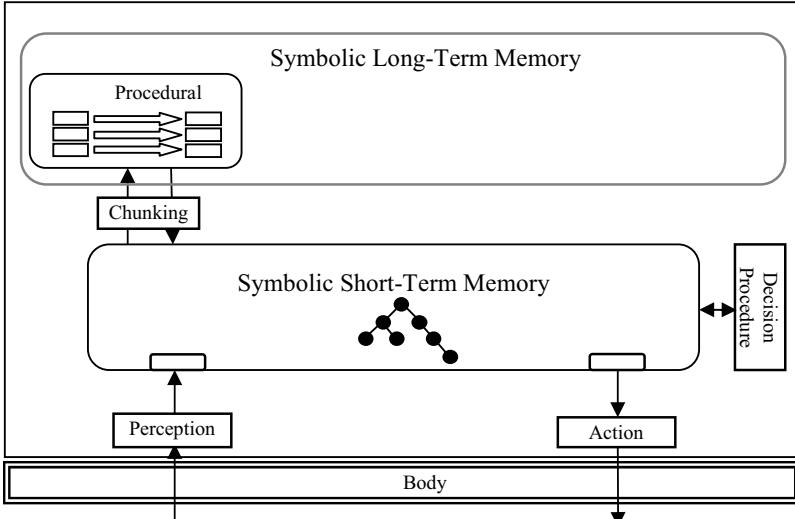


Figure 1: Structure of Soar 9

At the lowest level, Soar's processing consists of matching and firing rules. Rules provide a flexible, context-dependent representation of knowledge, with their conditions matching the current situation and their actions retrieving information relevant to the current situation. Most rule-based systems choose a single rule to fire at a given time, and this serves as the locus of choice in the system – where one action is selected instead of another. However, there is only limited knowledge available to choose between rules, namely the conditions of the rules, the data matched by the rules, and possibly meta-data, such as a numeric score, associated with the rules. There is no ability to use additional context-dependent knowledge to influence the decision. Soar allows additional knowledge to influence a decision by introducing *operators* as the locus for choice and using rules to propose, evaluate, and apply operators. Rules act as an associative-memory that retrieves information relevant to the current situation, so there is no need to select between them and thus, in Soar, rules fire in parallel.

The concept of operator is common in AI, but usually involves a monolithic data structure containing the operator's preconditions and actions. However, in Soar, the definition of an operator is distributed across multiple rules. Thus, in Soar, there are rules that *propose* operators that create a data structure in working memory representing the operator and an *acceptable preference* so that the operator can be considered for selection. There are also rules that *evaluate* operators and create other types of preferences that prefer one operator to another or provide some indication of the utility of the operator for the current situation. Finally, there are rules that *apply* the operator by making changes to working memory that reflect the actions of the operator. These changes may be purely internal or may initiate external actions in the environment. This approach supports a flexible representation of knowledge about operators – there can be many reasons for proposing, selecting, and/or applying an operator – some that are very specific and others that are quite general. This representation also makes it possible to incrementally build up operator knowledge structures, so that the definition of an operator can change over time as new knowledge is learned for proposal, selection, and application [8].

To support the selection and application of operators and to interface to external environments, Soar has the processing cycle shown in Figure 2.

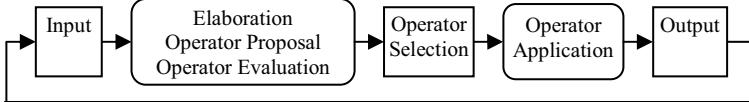


Figure 2: Soar's Processing Cycle

1. Input. Changes to perception are processed and sent to short-term memory.
2. Elaboration. Rules compute entailments of short-term memory. For example, a rule might test if the goal is to grasp an object, the object's distance, and the agent's reach, and then create a structure signifying whether the object is within reach.
3. Operator Proposal. Rules propose operators that are appropriate to the current situation based on features of the situation tested in the condition of the rules.
4. Operator Evaluation. Rules create preferences for which of the proposed operators should be preferred, based on the current situation and goal. The preferences can be symbolic (A is better than B), or numeric (the estimated utility of A is .73).
5. Operator Selection. A fixed decision procedure combines the generated preferences and selects the current operator. If the preferences are insufficient for making a decision, an *impasse* arises and Soar automatically creates a substate in which the goal is to resolve that impasse. In the substate, Soar recursively uses the same processing cycle to select and apply operators, leading to automatic, reactive meta-reasoning. The impasses and resulting substates provide a mechanism for Soar to deliberately perform any of the functions (elaboration, proposal, evaluation, application) that are performed automatically/reactively with rules.
6. Operator Application. The actions of an operator are performed by rules that match the current situation and the current operator structure. Multiple rules can fire in parallel and in sequence providing a flexible and expressive means for encoding operator actions. If there is insufficient application knowledge, an impasse arises with a substate. This leads to dynamic task decomposition, where a complex operator is performed recursively by simpler operators.
7. Output. Any output commands are passed on to the motor system.

Soar's procedural knowledge, decision-making, and subgoaling provide a strong foundation on to which to build. It has a shared short-term memory where knowledge from perception and long-term memory are combined to provide a unified representation of the current situation. It has a lean decision procedure that supports context-dependent reactive behavior, but also supports automatic impasse-driven subgoals and meta-reasoning. Chunking is Soar's learning mechanism that converts the results of problem solving in subgoals into rules – compiling knowledge and behavior from deliberate to reactive. Although chunking is a simple mechanism, it is extremely general and can learn all the types of knowledge encoded in rules [9].

3. Extended Soar

In extending Soar, we had two goals: 1. Retain the strengths of the original Soar: a flexible model of control and meta-reasoning along with the inherent ability to support reactive and deliberative behavior and the automatic conversion from deliberate to reactive behavior via chunking. 2. Expand the types of knowledge Soar could represent, reason with, and learn, inspired by human capabilities, but with the primary goal of additional functionality. The extensions fall into two, partially overlapping categories:

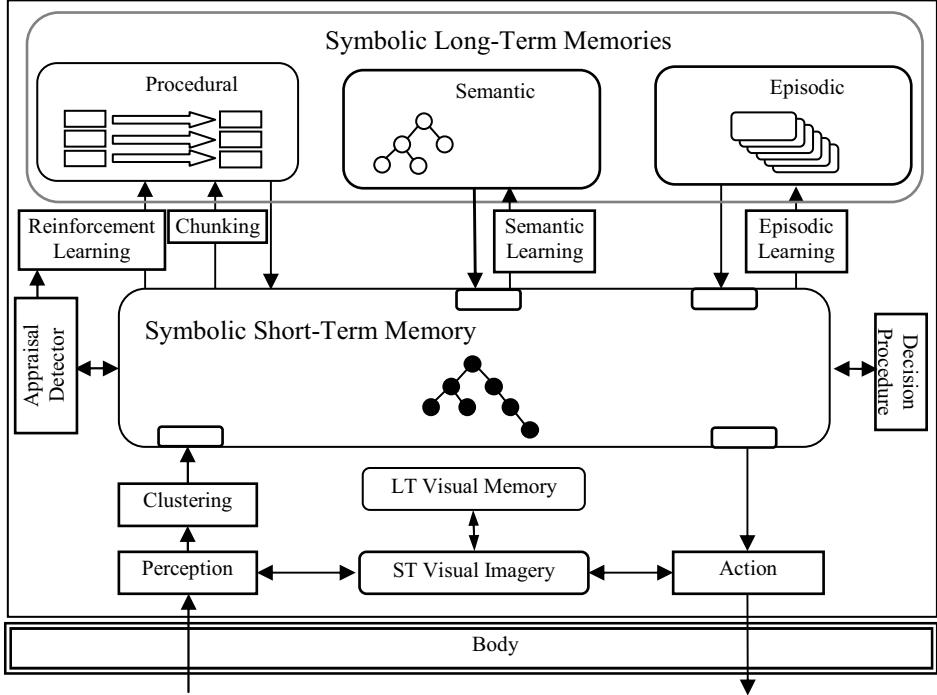


Figure 3: Soar 9

new non-symbolic representations of knowledge along with associated processing and memory modules, and new learning and memory modules that capture knowledge that is cumbersome to learn and encode in rules.

Figure 3 shows the structure of Soar, version 9. All of the new components have been built, integrated, and run with the traditional Soar components; however, as of yet there is not a single unified system that has all the components running at once. The major additions include: working memory activation, which provides meta-information about the recency and usefulness of working memory elements; reinforcement learning, which tunes the numeric preferences of operator selection rules; the appraisal detector, which generates emotions, feelings, and an internal reward signal for reinforcement learning; semantic memory, which contains symbolic structures representing facts; episodic memory, which contains temporally ordered “snapshots” of working memory; a set of processes and memories to support visual imagery, which includes depictive representations in which spatial information is inherent to the representation; and clustering, which dynamically creates new concepts and symbols.

Soar’s processing cycle is still driven by procedural knowledge encoded as production rules. The new components influence decision making indirectly by retrieving or creating structures in symbolic working memory that cause rules to match and fire. In the remainder of this section, we will give descriptions of these new components and discuss briefly their value and why their functionality would be very difficult to achieve by the existing mechanisms.

3.1. Working Memory Activation

Inspired by ACT-R [5], we added activation to Soar's working memory [10, 11]. Activation provides meta-information in terms of the recency of a working memory element and its relevance, which is computed based on when the element matched rules that fired. This information is not used to determine which rules to fire, as Soar fires all rules that match, but it is stored as part of episodic memories, biasing their retrieval so that the episode retrieved is the most relevant to the current situation. Empirical results verify that working memory activation significantly improves episodic memory retrieval [12]. In the future, we expect that working memory activation will be used in semantic memory retrieval and emotion. Working memory activation requires architectural changes because it must access and maintain information that is only available to the architecture (when working memory elements are created and matched).

3.2. Reinforcement Learning

Reinforcement learning (RL) involves adjusting the selection of actions in an attempt to maximize reward. In early versions of Soar, all preferences for selecting operators were symbolic, so there was no way to represent or adjust such knowledge; however, we recently added numeric preferences, which specify the expected value of an operator for the current state [13]. During operator selection, all numeric preferences for an operator are combined, and an epsilon-greedy algorithm is used to select the next operator. This makes RL in Soar straightforward – it adjusts the actions of rules that create numeric preferences for selected operators [14]. Thus, after an operator applies, all of the rules that created numeric preferences for that operator are updated based on any new reward and the expected future reward, which is simply the summed numeric value of the numeric preferences for the next selected operator. RL in Soar applies across all goals, including impasse-generated subgoals. One intriguing aspect of RL in Soar is that the mapping from situation and operator to expected reward (the value-function) is represented as collections of rules. Only those rules that match the current situation participate in the selection of an operator, and there can be many rules contributing estimates of future reward for a single operator. This representation supports varying degrees of coverage and hierarchical representations, which can greatly speed up learning [15].

RL would be very difficult to implement in Soar using only chunking. RL applies to every operator selection, on every decision, even when there is no impasse, while chunking only learns rules through impasses. In addition, RL modifies existing rules by changing the values of numeric preferences, while chunking only adds new rules. In fact, RL and chunking are quite complementary because when there are no selection rules, an impasse arises and problem solving in a subgoal can generate initial preferences for the tied operators. Chunking then creates rules to generate these initial preferences in the future, and RL then tunes the values as experience accumulates.

3.3. Emotion

The functional and computational role of emotion is open to debate; however, in the last twenty years there has been substantial research on appraisal theories of emotion [16]. These theories propose that an agent continually evaluates a situation and that evaluation leads to emotion. The evaluation is hypothesized to take place along

multiple dimensions, such as goal relevance (is this situation important to my goals?), goal conduciveness (is this situation good or bad for my goals?), causality (who caused the situation?), control (can I change the situation?), and so on. These dimensions are exactly what an intelligent agent needs to compute as it pursues its goals while interacting with an environment. Thus, we have created a computational implementation of a specific appraisal theory [17] in Soar, represented by the appraisal detector in Figure 3. In Soar, appraisals lead to emotions, emotions influence mood, and mood and emotion determine feelings [18]. Individual appraisals produce either categorical or numeric values, which combine to form an intensity of the current feeling. This intensity becomes the intrinsic reward [19] for reinforcement learning, which significantly speeds learning [20]. A major goal of our future work is to explore how emotion, mood, and feeling can be used productively with other modules (such as retrieval from long-term memory and decision making), as well as in interacting with other agents.

It is possible to implement similar theories of emotion in Soar without modifying the architecture [21]; however, in these approaches, Soar is used more as a programming language than as a cognitive architecture – all of the important functionality of the system comes from procedural knowledge encoded in Soar as opposed to the structure of the architecture. This makes it impossible for emotions, mood, and feelings to directly influence other architectural modules, which we expect is critical to many aspects of emotion, mood, and feelings.

3.4. Semantic Memory

In addition to procedural knowledge, which is encoded as rules in Soar, there is declarative knowledge, which can be split into things that are known, such as facts, and things that are remembered, such as episodic experiences. Semantic learning and memory provides the ability to store and retrieve declarative facts about the world, such as tables have legs, dogs are animals, and Ann Arbor is in Michigan. This capability has been central to ACT-R's ability to model a wide variety of human data and adding it to Soar should enhance our ability to create agents that reason and use general knowledge about the world. In Soar, semantic memory is built up from structures that occur in working memory [22]. A structure from semantic memory is retrieved by creating a cue in a special buffer in working memory. The cue is then used to search for the best partial match in semantic memory, which is then retrieved into working memory.

There has been a significant body of research on acquiring semantic knowledge through Soar's chunking mechanism, under the label of *data chunking* [23]. Although it is possible, it is not easy. Data chunking requires pre-existing knowledge about the possible structures that can exist in working memory, and the agent has to interrupt its current task processing to deliberately force an impasse in which chunking can learn the appropriate rules. Moreover, because the knowledge is encoded in rules, retrieval requires an exact match of the cue, limiting the generality of what is learned. These factors made it difficult to use data chunking in new domains, begging the question as to how it would naturally arise in a generally intelligent agent.

3.5. Episodic Memory

In contrast to semantic memory, which contains knowledge independent of when and where it was learned, episodic memory contains memories of what was experienced over time [24]. In Soar, episodic memory includes specific instances of the structures that occur in working memory at the same time, providing the ability to remember the context of past experiences as well as the temporal relationships between experiences [12]. An episode is retrieved by the deliberate creation of a cue, which is a partial specification of working memory in a special buffer. Once a cue is created, the best partial match is found (biased by recency and working memory activation) and retrieved into a separate working memory buffer (to avoid confusion between a memory and the current situation). The next episode can also be retrieved, providing the ability to replay an experience as a sequence of retrieved episodes.

Although similar mechanisms have been studied in case-based reasoning, episodic memory is distinguished by the fact that it is task-independent and thus available for every problem, providing a memory of experience not available from other mechanisms. Episodic learning is so simple that it is often dismissed in AI as not worthy of study. Although simple, one has only to imagine what life is like for amnesiacs to appreciate its importance for general intelligence [25]. We have demonstrated that when episodic memory is embedded in Soar, it enables many advanced cognitive capabilities such as internal simulation and prediction, learning action models, and retrospective reasoning and learning.

Episodic memory would be even more difficult to implement using chunking than semantic memory because it requires capturing a snapshot of working memory and using working memory activation to bias partial matching for retrieval.

3.6. Visual Imagery

All of the previous extensions depend on Soar's existing symbolic short-term memory to represent the agent's understanding of the current situation and with good reason. The generality and power of symbolic representations and processing are unmatched and our ability to compose symbolic structures is a hallmark of human-level intelligence. However, for some constrained forms of processing, other representations can be much more efficient. One compelling example is visual imagery [26], which is useful for visual-feature and visual-spatial reasoning. We have added a set of modules to Soar that support visual imagery [27], including a short-term memory where images are constructed and manipulated; a long-term memory that contains images that can be retrieved into the short-term memory; processes that manipulate images in short-term memory, and processes that create symbolic structures from the visual images. Although not shown, these extensions support both a depictive representation in which space is inherent to the representation, as well as an intermediate, quantitative representation that combines symbolic and numeric representations. Visual imagery is controlled by the symbolic system, which issues commands to construct, manipulate, and examine visual images.

With the addition of visual imagery, we have demonstrated that it is possible to solve spatial reasoning problems orders of magnitude faster than without it, and using significantly less procedural knowledge. Visual imagery also enables processing that is not possible with only symbolic reasoning, such as determining which letters in the alphabet are symmetric along the vertical axis (A, H, I, M, O, T, U, V, W, X, Y).

3.7. Clustering

One additional capability that has been missing from Soar is a mechanism that detects statistical regularities in the stream of experiences and automatically creates new symbolic structures that represent those regularities, providing a mechanism for automatically generating new symbols and thus concepts that can be used to classify perception. Few existing cognitive architectures have the capability to create new symbolic structures – they are a prisoner of the original encodings provided by a human programmer. To remedy this shortcoming, we have added a clustering module to Soar that is based on research by Richard Granger [28]. The underlying algorithms are derived from the processing of thalamocortical loops in the brain, where it appears there is clustering and successive sub-clustering of inputs using winner-take-all circuits. Although we do not yet have a general implementation of clustering for all types of perception in Soar, we have used clustering to create new symbolic structures that enrich that state representation and speed reinforcement learning.

3.8. Integration of New Modules in Soar

As mentioned earlier, the processing cycle of Soar did not require any changes to accommodate the new modules beyond invoking them at appropriate times. Working memory activation is updated when rules fire, while reinforcement learning is invoked when a new operator is selected and clustering is invoked during input processing. The retrieval for semantic and episodic memories is performed during the input phase, so that the results are available during the next processing cycle. The visual imagery system is treated as if it was an external module, with commands to the system creating by rules and then executed during the output phase, with new interpretations of the images created during the input phase. Although the modules are invoked at specific points in the processing cycle, they could run asynchronously. For example, when a cue is created for episodic memory, the retrieval processes could run in parallel with rule matching and firing. We have maintained the more rigid approach because it simplifies development and debugging, and because there was no computational advantage to having asynchronous threads before the advent of multi-core processors.

Although knowledge encoded in rules still controls behavior through the selection and application of operators, the new modules influence behavior through the working memory structures they create. For example, if a list of instructions is stored in semantic memory, with the appropriate rules, they can be retrieved, and lead to the selection of operators that interpret and execute them. Reactivity is maintained by rules that propose and prefer operators to process unexpected external events.

4. Discussion

4.1. Non-symbolic processing in Soar

One of the major changes in extending Soar has been the introduction of non-symbolic processing. In Soar, clustering is *subsymbolic*, where non-symbolic perceptual structures are combined together to create symbols. All the other non-symbolic processing is co-symbolic – it either controls symbolic processing (similar to ACT-R’s use of non-symbolic processing), or in the case of visual imagery, provides an

alternative to symbolic processing by providing a representational media for reasoning about the world. In contrast, Clarion has extensive subsymbolic processing [29] where it supports neural networks processing for some of its reasoning in addition to symbolic processing. Below is a chart of the different functional uses of non-symbolic processing in Soar.

Non-symbolic Processing	Function
Numeric Preferences	Control operator selection
Reinforcement Learning	Learn operator selection control knowledge
Working memory activation	Aid long-term memory retrieval
Visual Imagery	Represent images and spatial data for reasoning
Appraisals: Emotions & Feelings	Summarize intrinsic value of situation – aid RL
Clustering	Create symbols that capture statistical regularities

4.2. Analysis of Memories and Learning in a Cognitive Architecture

The second major change is the addition of new memories and learning systems. Below we summarize the major dimensions of variations in the memory and learning systems:

Memory/Learning System	Source of Knowledge	Representation of knowledge	Retrieval of knowledge
Chunking	Traces of rule firings in subgoals	Rules	Exact match of rule conditions, retrieve actions
Clustering	Perception	Classification networks	Winner take all
Semantic Memory	Working memory existence	Mirror of working memory object structures	Partial match, retrieve object
Episodic Memory	Working memory co-occurrence	Episodes: Snapshots of working memory	Partial match, retrieve episode
Reinforcement Learning	Reward and numeric preferences	Numeric preferences	Exact match of rule conditions, retrieve preference
Image Memory	Image short-term memory	Image	Deliberate recall based on symbolic referent

4.3. Future work

The addition of these new modules only scratches the surface of the potential research, with explorations of the interactions and synergies of these components being the next order of business. Some of the interactions are obvious, such as using RL to learn the best cues for retrieving knowledge from episodic memory, or using episodic memory for retrospective learning to train RL on important experiences. Other interactions require deeper integrations, such as how emotion and visual imagery are captured and used in episodic memory, how knowledge might move between episodic and semantic memories, how emotions and feelings influence decision making, or how emotion and working memory activation might influence storage and retrieval of knowledge in episodic and semantic memory. Finally, we need to study the impact of these extensions on Soar's status as a unified theory of cognition [30].

4.4. Final thoughts

So far, the explorations in psychology and AI through the space of cognitive architectures have been extremely fruitful. Research on ACT-R has led to comprehensive computational theories of a wide variety of human phenomena, including brain activity [5], while research on EPIC has made great strides in modeling the detailed interactions of cognition with perception and action [31]. However, these architectures and others have ignored many of the cognitive capabilities we are now studying in Soar (episodic memory, emotion, visual imagery), and it has taken us close to twenty years to study them. Why is this? The simple answer is that our architectures are prisoners of the tasks we study. To date, most of the research on cognitive architecture has arisen from studying short-term tasks – similar to those that arise in a psychology laboratory setting where there is minimal emotional engagement and little need for personal history. The challenge for the future is to develop general artificial systems that can pursue a wide variety of tasks (including novel ones) over long time scales, with extensive experiential learning in dynamic, complex environments. As is evidenced from this paper, my belief is that research in cognitive architecture provides the building blocks to meet this challenge.

Acknowledgments

The author wishes to thank the current and former graduate students and research programmers who have contributed to the extensions to Soar described in this paper: Ron Chong, Karen Coulter, Michael James, Scott Lathrop, Robert Marinier, Shelley Nason, Andrew Nuxoll, Jonathan Voigt, Yongjia Wang, and Samuel Wintermute.

The author acknowledges the funding support of the National Science Foundation under Grant No. 0413013 and the DARPA “Biologically Inspired Cognitive Architecture” program under the Air Force Research Laboratory “Extending the Soar Cognitive Architecture” project award number FA8650-05-C-7253.

References

- [1] Laird, J. E., & Rosenbloom, P. S. (1996) The evolution of the Soar cognitive architecture. In T. Mitchell (ed.) *Mind Matters*, 1-50.
- [2] Rosenbloom, P. S., Laird, J. E., & Newell, A. (1993) *The Soar papers: Research on Integrated Intelligence*. MIT Press, Cambridge, MA.
- [3] Jones, R. M., Laird, J. E., Nielsen, P. E., Coulter, K. J., Kenny, P., & Koss, F. V. (1999) Automated intelligent pilots for combat might simulation. *AI Magazine*, 20(1), 27-41.
- [4] Wray, R. E., Laird, J. E., Nuxoll, A., Stokes, D., & Kerfoot, A. (2005) Synthetic adversaries for urban combat training. *AI Magazine*, 26(3), 82-92.
- [5] Anderson, J. R. (2007) *How Can the Human Mind Occur in the Physical Universe?* Oxford University Press.
- [6] Cho, B., Rosenbloom, P. S. & Dolan, C. P. (1991) Neuro-Soar: A neural-network architecture for goal-oriented behavior, *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*, 673-677. Chicago, IL.
- [7] Taatgen, N. A., Juvina, I., Herd, S., Jilk, D., & Martens, S. (2007) Attentional blink: An internal traffic jam? Eighth International Conference on Cognitive Modeling.
- [8] Pearson, D. J., & Laird, J. E., Incremental Learning of Procedural Planning Knowledge in Challenging Environments, *Computational Intelligence*, 2005, 21(4), 414:439.

- [9] Steier, D.S., Laird, J.E., Newell, A., Rosenbloom, P.S., Flynn, R., Golding, A., Polk, T.A., Shivers, O., Unruh, A., Yost, G.R. (1987) Varieties of Learning in Soar. Proceedings of the Fourth International Machine Learning Workshop.
- [10] Chong, R. (2003) The addition of an activation and decay mechanism to the Soar architecture. *Proceedings of the Fifth International Conference on Cognitive Modeling*, Pittsburgh, PA.
- [11] Nuxoll, A. M., Laird, J. E., & James, M. (2004) Comprehensive working memory activation in Soar, *Sixth International Conference on Cognitive Modeling*, 226-230.
- [12] Nuxoll, A. M., & Laird, J. E. (2007) Extending cognitive architecture with episodic memory. *Proceedings of the 21st National Conference on Artificial Intelligence*.
- [13] Wray, R. E & Laird, J.E. (2003) Variability in Human Behavior Modeling for Military Simulations. *Proceedings of the 2003 Conference on Behavior Representation in Modeling and Simulation*. Scottsdale, AZ.
- [14] Nason, S., & Laird, J. E. (2005) Soar-RL: Integrating reinforcement learning with Soar. *Cognitive Systems Research*, 6(1), 51-59.
- [15] Wang, Y., and Laird, J.E. (2007) The Importance of Action History in Decision Making and Reinforcement Learning. *Proceedings of the Eighth International Conference on Cognitive Modeling*. Ann Arbor, MI.
- [16] Roseman, I. & Smith, C. A. (2001) Appraisal theory: Overview, Assumptions, Varieties, Controversies. In Scherer, Schorr, & Johnstone (Eds.) *Appraisal processes in Emotion: Theory, Methods, Research*, 3-19. Oxford University Press.
- [17] Scherer, K. R. (2001) Appraisal considered as a process of multi-level sequential checking. In K. R. Scherer, A. Schorr, & T. Johnstone (Eds.) *Appraisal processes in Emotion: Theory, Methods, Research*. 92-120. Oxford University Press.
- [18] Marinier, R. P., & Laird, J. E. (2007) Computational Modeling of Mood and Feeling from Emotion. *Proceedings of 29th Meeting of the Cognitive Science Society*. 461-466. Nashville: Cognitive Science Society.
- [19] Singh, S., Barto, A. G., & Chentanez, N. (2004) Intrinsically motivated reinforcement learning. *18th Annual Conference on Neural Information Processing Systems (NIPS)*.
- [20] Marinier, R. P., & Laird, J. E. (2008) Emotion-Driven Reinforcement Learning. CCA-TR-2007-02, Center for Cognitive Architecture, University of Michigan.
- [21] Gratch, J. & Marsella, S. (2004) A Domain-independent Framework for Modeling Emotion. *Cognitive Systems Research*, 5:269-306, 2004.
- [22] Wang, Y., and Laird, J.E. 2006. Integrating Semantic Memory into a Cognitive Architecture. CCA-TR-2006-02, Center for Cognitive Architecture, University of Michigan.
- [23] Rosenbloom, P. S. (2006) A cognitive odyssey: From the power law of practice to a general learning mechanism and beyond. *Tutorials in Quantitative Methods for Psychology*, 2(2), 38-42.
- [24] Tulving, E. (1983) *Elements of Episodic Memory*. Oxford: Clarendon Press.
- [25] Nolan, C. (2000) *Memento*, New Market Films.
- [26] Kosslyn, S. M., Thompson, W. L., & Ganis, G. (2006). *The Case for Mental Imagery*. New York, New York: Oxford University Press.
- [27] Lathrop, S.D., and Laird, J.E. (2007). Towards Incorporating Visual Imagery into a Cognitive Architecture. *Eighth International Conference on Cognitive Modeling*.
- [28] Granger, R. (2006) Engines of the brain: The computational instruction set of human cognition. *AI Magazine* 27: 15-32.
- [29] Sun, R. (2006). The CLARION cognitive architecture: Extending cognitive modeling to social simulation. In: Ron Sun (ed.), *Cognition and Multi-Agent Interaction*. Cambridge University Press, New York.
- [30] Newell, A. (1990) *Unified Theories of Cognition*. Harvard University Press.
- [31] Kieras, D. & Meyer, D. E. (1997) An overview of the EPIC architecture for cognition and performance with application to human-computer interaction. *Human-Computer Interaction*, 12, 391-438.

Temporal Action Logic for Question Answering in an Adventure Game

Martin MAGNUSSON and Patrick DOHERTY

Department of Computer and Information Science

Linköping University, 581 83 Linköping, Sweden

E-mail: {marma, patdo}@ida.liu.se

Abstract. Inhabiting the complex and dynamic environments of modern computer games with autonomous agents capable of intelligent timely behaviour is a significant research challenge. We illustrate this using our own attempts to build a practical agent architecture on a logicist foundation. In the ANDI-Land adventure game concept players solve puzzles by eliciting information from computer characters through natural language question answering. While numerous challenges immediately presented themselves, they took on a form of concrete and accessible problems to solve, and we present some of our initial solutions. We conclude that games, due to their demand for human-like computer characters with robust and independent operation in large simulated worlds, might serve as excellent test beds for research towards artificial general intelligence.

Keywords. Temporal Action Logic, computer games, natural language understanding, artificial general intelligence, natural deduction, planning, epistemic reasoning

1. Introduction

Two topics that have seen a recent boost of interest are research on artificial general intelligence (AGI) and the use of modern computer games as AI research test beds. There is much to say in favour of combining these trends, though we confine ourselves to two important observations. First, games are readily accessible both for the scientist who can use existing games with exposed APIs, or relatively easily implement entirely new games, and for the peer researcher or student who can download *and experiment with* the software themselves. Second, their demand for human-like behaviour in complex environments necessitates a certain amount of generality in any proposed solution. Game environments are much more complex than classical benchmark problems such as the blocks world, which are often criticised for their limited scope (e.g. by Hayes [1]). In contrast, most computer games are incompatible with simplifying assumptions such as the (in)famous closed world assumption and call for many of the capabilities needed for general intelligence such as an agent architecture that integrates everything from perception to action, robustness and responsiveness in sometimes unpredictable environments, goal-directed action planning, multi agent communication, reasoning about knowledge and how to obtain it, and natural language understanding for dialog interaction.

Our own work involves research on topics relevant to an adventure game project where a human player solves simple puzzles through natural language question answer-

ing dialogs with ANDIs, agents with Automated Natural Deduction based Intelligence, who inhabit ANDI-Land. Present day games almost universally adopt straight jacketed exchanges typically featuring a choice between three canned sentences, two of which are humorous sidetracks and one that will move the dialog forward to the next set of sentences. Our aim is to eliminate the forced linearity of scripted dialogs through artificial intelligence technology. Rather than mindlessly trying all alternatives, we would have the player think¹. The reader is encouraged to evaluate the results we describe below through experimentation with our demonstrator available for download (as a Windows binary) at www.andi-land.com.

The long term aim is wide coverage natural language understanding, which requires both extensive knowledge of the topics under discussion and the capability to reason with it. Such demands can ultimately only be satisfied by true AGI, while our efforts to date are certainly not in that ballpark. But our initial experiences with ANDI-Land indicate that a computer game setting enables an incremental approach where reasonably difficult challenges can be attacked while keeping the long term goal in mind. The work presented below is, for this reason, based on a logicist foundation. We believe the best way to approach general intelligence is by formulating most types of reasoning in a unified proof system for deductive and non-monotonic types of inference in a, not necessarily purely classical, logical formalism expressive enough to capture all the subtleties and distinctions that humans make in their reasoning. If successful, such an endeavour will allow the use of efficient specialized reasoning processes when applicable, yet always providing the option to fall back on more general but less efficient methods of proof in new and unforeseen situations.

Rather than expanding further on this nebulous conjecture we will discuss the specific research problems that immediately suggested themselves when we initiated work on our question answering adventure game concept, in Section 2. Section 3 presents example dialogs from an in-game scenario that illustrate some capabilities of the architecture built in response to the challenges. A hopelessly inadequate selection of related work, squeezed into Section 4, will have to make do for orienting our efforts in relation to others'. Finally, Section 5 concludes with a look towards the future.

2. ANDI-Land

ANDI-Land consists of an isometric graphical representation of a forest that can be explored by a player through a keyboard controlled avatar. The forest is inhabited by intelligent agents with which the player can initiate question answering conversations, and who sometimes proactively do so themselves in order to further their own goals. There is a puzzle element to make interaction interesting, but unlike most other adventure type games, solving puzzles through a process of eliminating all alternatives is not feasible since the natural language input is not sufficiently restrictive. Implementing this concept requires providing ANDI-Land agents with a genuine understanding of questions posed to them and equipping them with knowledge of their virtual world from which to deduce answers. Only the coordination of linguistic and semantic processing can make this possible.

¹Although whether this constitutes an enjoyable game experience depends, of course, on the player.

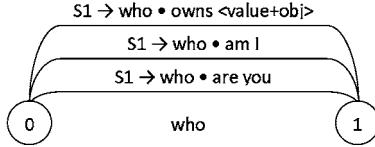


Figure 1. Active edges in the chart resulting from parsing “who” show the possibilities for the next word in the sentence, “owns”, “am”, and “are”.

2.1. Interactive Natural Language Input

As we pointed out in Section 1, it would be unrealistic to expect both broad and deep natural language understanding sooner than the development of AGI. First, no one has yet been able to construct a grammar with adequate coverage of the entire English language. The problem is not just that players of the game will not be able to express themselves in whatever way they want, but worse, nor will they receive any hints as to how to rephrase themselves in a way that the parser will understand. Second, the knowledge of the game characters will, while aiming at generality, realistically start out as downright narrow. Most sentences would simply be outside the area of competence of the game characters, and they would have to respond with an honest “I don’t know”.

These two problems threaten to reduce the adventure to a sort of guessing game where the player, more or less blindly, would have to search for sentences that both avoid the equivalent of a “parse error” message and whose semantic meaning happens to produce something other than a puzzled look on the respondent’s face. Using a very large wide coverage grammar like the English Resource Grammar [2] would seem to help alleviate the first problem, but at the cost of worsening the second. The semantical form it produces is not detailed enough to suffice for automated reasoning and question answering. Our project would be quite stranded if these problems left no room for incremental progress building on a modest start.

ANDI-Land incorporates a unique form of interactive natural language input that deal with both problems by guiding the player towards sentences that parse correctly and fall within the AIs competence areas. For example, the sentence “who is the lumber’s owner” is not supported by our current grammar, but the sentence “who owns the lumber” is. Even though this restriction is arbitrary, the player is spared much frustration due to the interactive parser. As soon as the player starts typing, a chart parser (as described in [3])² starts producing partial parses that cover as many words as possible. Though the initial word “who” does not constitute a complete sentence, the resulting parse chart still contains useful information. Specifically, by looking at active chart edges we can collect all words that would advance the parse if they occurred immediately succeeding the current input. According to Figure 1, the words “owns”, “am”, and “are” constitute all the possible continuations of the current input, and choosing among them effectively circumvents all sentences that have no chance of resulting in a complete parse. The process could be likened to the widespread T9 mobile phone text input, except that the system understands the grammar of entire sentences rather than just the correct spelling of words. Furthermore, by limiting the grammar to concepts covered by the agent’s background knowledge we can ensure that most of the input sentences are answered intel-

²The parser and grammar are extremely simple. We would like to improve them at a later date, perhaps modifying the Linguistic Knowledge Builder [2] for our interactive mode.

Natural language question	Who owns the lumber?
Input logical form	$\exists ans [value(\text{now}, \text{owner}(\text{lumber})) = ans]$
Answer variable binding	$ans = djak$
Answer logical form	$value(\text{now}, \text{owner}(\text{lumber})) = djak$
Natural language response	Djak owns the lumber.

Figure 2. The process from natural language question to natural language response is largely symmetrical, thanks to a “reversible” grammar.

ligently. Compared to scripted dialogs, the interactive input method presents the player with a multitude of choices, even using a very small grammar, while also allowing for gradual improvements in language coverage.

2.2. Reversible Natural Language Grammar

Another challenge, natural language generation, is presented by communication in the opposite direction. Each grammar rule has an associated lambda expression that represents its meaning. Meaning fragments are combined by lambda application that eventually, after a complete parse of the sentence, results in formulas of first-order logic. These formulas are directly amenable to automated reasoning to produce an answer expression, encoding the response in first-order logic. Of course, we would prefer the game characters to reply in plain English. Shieber’s uniform architecture [4] for both parsing and generation addresses this difficulty with minimal machinery. It effectively makes the natural language grammar reversible with relatively small modifications to the basic chart parsing algorithm. There is no need for separate generation algorithms. Furthermore, extensions to the input grammar that help the ANDI-Land inhabitants understand new words or grammar automatically increase their proficiency in using them in speech too. Figure 2 illustrates the question answering process using the example sentence from the previous section. A question is parsed into a formula that contains an *answer variable*. Its value, found through the theorem proving techniques described in Section 2.6, can be used to instantiate the query to form an answer expression. Finally, the chart parser is run in “reverse” to produce a natural language response to the original question.

2.3. Temporal Action Logic

We said that the logical forms resulting from parsing were amenable to automated reasoning. Work within the methodology of formal logic provides a comprehensive tool set for correct reasoning. However, the standard philosophical logic turns out to be inadequate to support the thinking processes of *active* characters in *dynamic* environments. Researchers in cognitive robotics are therefore creating new powerful logics that are applicable to commonsense reasoning about action and change as well as more traditional logical reasoning. We have chosen to work with one such logic, the Temporal Action Logic (TAL), which adopts an intuitive explicit time line to describe actions and their effects in a changing environment. The origins of TAL can be found in the Features and Fluents framework developed by Sandewall [5], but it was a new characterization in terms of first-order logic with circumscription, by Doherty [6], that made automated reasoning possible. Many extensions since have turned TAL into a very expressive language capable of representing, among other things, actions with durations, context-dependent and non-deterministic actions, concurrency, and action side-effects.

But the most important feature of TAL might be its *occlusion* concept that serves as a flexible tool to deal with important aspects of the frame problem, which has long haunted logical approaches to AI. Properties and relations that may change over time are modelled by *fluents*, and the value v of a fluent f can be linked to a time point t on the time line using a function $\text{value}(t, f) = v$. Some agent a (denoted *self* when doing the thinking) carrying out an action c during time interval i is specified by $\text{Occurs}(a, i, c)$. The following formula relates a fluent f 's value at the starting and ending time points of a time interval i , unless the fluent is occluded, as specified by $\text{Occlude}(i, f)$:

$$\forall i, f [\neg \text{Occlude}(i, f) \rightarrow \text{value}(\text{start}(i), f) = \text{value}(\text{finish}(i), f)] \quad (1)$$

The role of circumscription is the minimization of action occurrences and occlusion to implement the blanket assumption that no unexpected actions occur and fluents' values persist over time. Exceptions are specified by explicit action occurrences and their occlusion of fluents they affect, thus releasing them from the frame assumption that their values remain unchanged. E.g., if the game character Djak was to sell the lumber he possesses in Figure 2, the fluent *owner(lumber)* would be occluded during any interval that overlaps the interval during which the selling occurs, and Formula 1 would not be applicable.

2.4. Reasoning and Planning

However, one of the most important forms of reasoning is not supported by the TAL framework as described above, namely proactive *planning* to achieve goals. E.g., if Djak's (modest) goal in life is the possession of lumber, he could reason that going to a shop and buying lumber is one possible plan to satisfy it. But his reasoning must allow the consideration of different actions before committing to any particular plan. Djak should not commence going to the store before considering whether the store actually sells lumber or not, since if it does not he might have to resort to an alternative sequence of actions such as cutting down some trees himself. However, committed knowledge about the set of actions is a prerequisite to automated reasoning using the circumscription account of TAL [7]. In contrast, we would like the set of actions to be a *consequence* of reasoning. This was accomplished in previous work [8] in a constraint logic programming setting. Rather than circumscribing a fixed set of actions, we use constraints to keep track of assumptions that depend on the set of actions, and reevaluate those assumptions when the set of actions change. The mechanism was cast as deduction, but the same principles are recast as abduction in a new first-order theorem proving setting described in Section 2.6. Thus equipped, Djak is both able to answer questions and plan his actions through automated reasoning with TAL.

2.5. Epistemics

Game agents, however, must face an additional complication when planning their actions. Inhabitants of the game world can not reasonably be assumed to possess complete knowledge of their entire world, and even if they did, the dynamic nature of game environments would quickly make this knowledge obsolete. The closed world assumption that is at the foundation of many classical planning systems is not applicable. Instead, an intelligent agent must reason with incomplete information and, significantly, plan to

obtain additional information when needed. E.g., suppose another ANDI-Land agent, Keypr, owns a shop. Although Keypr is all out of lumber, he could sell Djak an axe to use to cut down a tree with. Being an intelligent and proactive fellow, Djak might come up with the following plan fragment (excluding the tree cutting part):

$$\exists i_1, i_2 [Occurs(\text{self}, i_1, \text{walk}(\text{value}(\text{start}(i_2), \text{location}(\text{keypr})))) \wedge \\ Occurs(\text{self}, i_2, \text{buy}(\text{axe}, \text{keypr})) \wedge \\ \text{finish}(i_1) = \text{start}(i_2)]$$

Though, what if Djak does not *know* Keypr's location? The plan is still correct in the sense that if Djak executed it, the intended effects would manifest. The problem is that it is not *executable*. There is no way Djak can (willingly) walk to Keypr's location without knowing what that location is, but we have as of yet no means to express this additional *knowledge precondition*. What is needed is an epistemic logic that includes a notion of knowledge.

The most common way of introducing such a notion of knowledge is in the form of a modal operator *Knows* with a possible worlds semantics. This can be done while remaining in classical logic by encoding the possible worlds and the accessibility relation between them explicitly in the object language, as e.g. in Moore's pioneering work [9]. But these approaches are associated with some limitations that make them unsuitable as *general* frameworks of epistemic reasoning, as pointed out e.g. by Morgenstern [10]. She proposes an alternative treatment that introduces *Knows* as a "syntactic" predicate, which accepts quoted formulas as arguments. Quotation can be seen as an extreme form of reification where any formula can be turned into a term. It appears to be both simpler and more intuitive than possible world semantics in many contexts. Unfortunately, quotation is associated with the risk of paradoxes. While it is true that unrestricted quotation leads to the paradox of the Knower [11], there are methods for avoiding these problems (a particularly interesting one is Perlis' [12], which still allows for self-referential formulas). Our work, although adopting the syntactic quotation framework in anticipation of requirements of generality, has not yet proceeded far enough to utilize the additional expressivity afforded by syntactical treatments of knowledge over modal variants, a fact that guarantees consistency [13] and allows us to remain uncommitted as to which more general treatment to give preference to.

Equipped with the ability to represent knowledge explicitly we add a precondition to walking that one should know where the destination is. We can also make use of the *Knows* predicate in action effects, thereby formalizing knowledge producing actions and putting us in a position where planning for knowledge acquisition is possible. Adding an action for asking another agent (such as the player!) about a fluent's value enables Djak to come up with a plan that is both executable and that has the intended effect:

$$\exists i_1, i_2, i_3 [Occurs(\text{self}, i_1, \text{askValue}(\text{player}, \text{location}(\text{keypr}))) \wedge \\ Occurs(\text{self}, i_2, \text{walk}(\text{value}(\text{start}(i_3), \text{location}(\text{keypr})))) \wedge \\ Occurs(\text{self}, i_3, \text{buy}(\text{axe}, \text{keypr})) \wedge \\ \text{finish}(i_1) = \text{start}(i_2) \wedge \text{finish}(i_2) = \text{start}(i_3)]$$

2.6. Natural Deductive Theorem Proving

Many agent architectures are built on a logic programming foundation, as was our previous work [8]. Logic programs incorporate some of the power of theorem proving while remaining relatively simple and allowing a high degree of control over the inference mechanism. But a fundamental limitation of Prolog is the assumption of complete knowledge, which, as we noted in Section 2.5, is unreasonable in complex computer games. In the interest of overcoming this limitation one can augment Prolog with meta-interpreters or other add-ons. Though when setting the sights for general intelligence it seems to us that augmenting Prolog will, over time, gradually approach general first-order theorem proving but in a roundabout and unnecessarily complicated way.

An alternative approach is to start with a first-order resolution theorem prover and complement it with special purpose modules that make some types of reasoning highly efficient. This is the method taken by the Cyc team, who have gone one step further and given up completeness in favour of efficiency and expressiveness [14]. Our (limited) experience with resolution suggests to us that it is not quite the natural fit with common-sense reasoning that one would hope. For example, the need to compile the knowledge base into clause form destroys potentially useful structural information that was previously implicit in the syntactic form of knowledge and rules, and the use of a single proof rule based on *reductio ad absurdum* could be incompatible with the defeasible reasoning that has turned out to be so important to commonsense reasoning [15].

Still, resolution completely dominates the field of automated theorem proving, but it is not the only contender. One particularly interesting alternative is *automated natural deduction*. Rather than compiling the agent's knowledge into clause form, such a theorem prover works with the "natural form" directly. And the rule set is extensible, thereby supporting the addition of special purpose rules, e.g. for defeasible reasoning. Moreover, whether the term "natural" is grounded in any relation between the deductive system and human reasoning is an exciting prospect explored by Rips, who argues a positive verdict [16].

In light of these considerations we have opted for our ANDI-Land inhabitants to "think" using an automated natural deduction theorem prover. Input formulas use the quantifier free form described by Pollock [17] and Rips [16]. This eliminates the somewhat cumbersome natural deduction rules for quantifier elimination and introduction while still preserving the knowledge base's natural form to a large extent. Most importantly, it provides the opportunity to work with unification and enables the use of *answer extraction* for question answering by binding answer variables to values as exemplified in Figure 2. Rather than a select few inference rules there is a set of *forward* rules (four at the moment), which are applied whenever they become applicable, and a set of *backward* rules (currently eleven of them), which are used in a goal-directed search for a proof. Finally, equality is dealt with through a system of rewrite rules, and temporal relations are added to a general temporal constraint network [18], exemplifying the use of special purpose reasoning mechanisms for efficiency.

A novel proof rule worthy of mention is a special abduction rule that allows relations from a set of *abducibles* to be assumed rather than proven, as long as doing so does not lead to inconsistency. This "natural abduction" rule forms the basis of the mechanism for non-monotonic reasoning and planning. As an example, consider the following natural deduction proof fragment (where the justifications in the right margin denote (P)remises, (H)ypotheses, the agents background (K)nowledge, and row numbers):

1	$value(12:00, \text{location}(\text{self})) = \text{loc}(1, -1)$	P
2	$start(i_{37}) = 12:00$	P
3	$finish(i_{37}) = 13:00$	P
4	$\neg Occlude(i_{37}, \text{location}(\text{self}))$	H
5	$value(13:00, \text{location}(\text{self})) = \text{loc}(1, -1)$	1 – 4, K
6	$\neg Occurs(\text{self}, i_{38}, \text{walk}(\text{loc}(0, 0)))$	H
7	$value(finish(i_{38}), \text{location}(\text{self})) = \text{loc}(0, 0)$	6, K
8	$\forall i [\neg Occlude(i, \text{location}(\text{self})) \rightarrow \neg Overlap(i, i_{38})]$	6, K
9	$\neg Overlap(i_{37}, i_{38})$	4, 8

The agent starts at the location with coordinate $\langle 1, -1 \rangle$ at noon, as in Row 1. Suppose the agent needs to remain at the same location at 1 p.m. One way of proving this would be to use persistence. The location fluent is only persistent if it is not occluded, and while the agent has no knowledge about whether it is occluded or not, $\neg Occlude$ is an abducible and may thus be *assumed*. Rows 2-4 introduces a new interval constant and indicates the assumption using a natural deduction style vertical line in the margin. Suppose further that the agent, for some other reason, needs to visit location $\langle 0, 0 \rangle$. The only way of proving this would be if a walk action destined for that coordinate occurred. When planning, *Occurs* is also abducible, so the agent assumes such an action in Row 6. The effect on the agent's location is recorded by Row 7. Walking should occlude the location fluent, but instead of stating that the fluent is occluded in any interval that overlaps the walk action, Row 8 uses the contra-position, stating that any interval that has assumed the location to be persistent must not overlap with the action of walking. This triggers the forward modus ponens rule to produce Row 9, partially ordering the two intervals to avoid any conflict between the persistence of the agent's location, and the agent's moving about. The non-overlap constraint is automatically added to the temporal constraint network. If it is impossible to order i_{37} and i_{38} so that they do not overlap in any way, the network becomes inconsistent, and the prover needs to backtrack, perhaps cancelling the most recent assumption. The abduction rule thus enables both defeasible conclusions about the persistence of fluents and, simultaneously, planning of new actions.

The use of the contrapositive form illustrates a case where two logically equivalent formulas have different effects in the system due to their surface form. If the occlusion had been expressed as $\forall i [Overlap(i, i_{38}) \rightarrow Occlude(i, \text{location}(\text{self}))]$, nothing would have triggered the non-overlap constraint. This, in turn, illustrates another important point. If the non-overlap constraint would make the temporal constraint network inconsistent, failing to trigger it could result in the agent failing to discover that one of its assumptions is unreasonable. This would not be a cause of unsoundness, since we are still within the sound system of natural deduction, but it might result in plans and conclusions that rest on impossible assumptions. A conclusion Φ depending on an inconsistent assumption would in effect have the logical form $\perp \rightarrow \Phi$, and thus be tautological and void. This is to be expected though since consistency is not even semi-decidable for first-order logic. The most we can hope for is for the agent to continually evaluate the consistency of its assumptions, improving the chances of them being correct over time, while regarding conclusions as tentative [15].

Another novelty is an execution rule linked to the agent's action execution mechanism, which is used to put plans into effect. Instead of sending the entire plan to a “dumb” execution module, we use the execution rule in “proving” that the plan is executed, thereby enabling the full reasoning power of the natural deduction prover to be

You steer your avatar Magni eastward and stumble upon another ANDI-Land character:

- M) Hello!
- K) Hello!
- M) Who are you?
- K) I am Keypr.
- M) What do you own?
- K) I own the axe.
- M) What is the axe's price?
- K) The axe's price is 5 gold.
- M) What is my wealth? (thinking)
- M) My wealth is 4 gold.
- M) Goodbye!
- K) Goodbye!

Dismayed by a sense of acute poverty, you continue to investigate the great forest. South-west lives another character, and as soon as he spots you, he comes running:

- D) Hello!
- M) Hello!
- D) Who owns the axe?
- M) Keypr owns the axe.
- D) What is Keypr's location?
- M) Keypr's location is 1 screen east and 2 screen north.
- D) Goodbye!

M) Goodbye!

Before you have a chance to ask his name, he hurries northward. Curious, you follow. At Keypr's, you observe the following dialog:

- D) Hello!
- K) Hello!
- D) Sell the axe to me.
- K) OK.
- D) Goodbye!
- K) Goodbye!

Somewhat envious of the axe-wielding stranger, you follow him back and watch him start applying the axe to the trunk of a tree. Determined to know his identity you confront him:

- M) Hello!
- D) Hello!
- M) Who are you?
- D) I am Djak.
- M) What happened?
- D) I bought the axe from Keypr.
- M) What do you own?
- D) I own the axe.
- M) Goodbye!
- D) Goodbye!

While you watch eagerly as Djak strikes the tree, it suddenly disap-

pears:

- M) Hello!
- D) Hello!
- M) What do you own?
- D) I own the axe and I own the lumber.
- M) What is the lumber's price?
- D) The lumber's price is 3 gold.
- M) Sell the lumber to me.
- D) OK.
- M) Goodbye!
- D) Goodbye!

Acting as a middle man, you revisit Keypr to try to sell the lumber:

- M) Hello!
- K) Hello!
- M) What is the lumber's price?
- K) The lumber's price is 6 gold.
- M) Buy the lumber from me.
- K) OK.
- M) What is my wealth (thinking)
- M) My wealth is 7 gold.

Intrigued by Djak and Keypr's limited displays of intelligence, but convinced that more must be possible, you vow to research AI in games!

Figure 3. This scenario from the ANDI-Land adventure game concept involves planned use of speech acts to satisfy the knowledge preconditions of buying an axe.

used in finding the exact action parameters for each step of the plan. Consider, e.g., Djak's plan to ask the player about Keypr's location in Section 2.5. Depending on the player's reply, further reasoning might be needed to convert this reply into a form that is suitable to pass as an argument to the action execution mechanism. This reasoning might depend on background knowledge about local geography and, in general, any amount of deliberation might be required during execution of a plan that involves knowledge acquisition, a fact respected by our execution proof rule.

3. A Dialog Scenario

Figure 3 illustrates all the components working together through example dialogs from ANDI-Land. The scenario revolves around our friends Djak and Keypr, from previous sections, but starts with the human player's avatar Magni in the middle of a thick forest. Djak's plan was automatically generated by the natural deductive theorem prover and its abduction rule, while the plan execution and all dialogs are straight from a running game session.

4. Related Work

In a spirit similar to ours, Amir and Doyle have proposed the use of text adventure games as a vehicle of research in cognitive robotics [19]. But instead of intelligent agents acting in supporting roles to enhance a human player's experience, they consider what challenges an agent would face if trying to solve the adventure itself. The agent would start out with severely limited knowledge, not knowing what actions are available to it, what fluents it should use to represent the environment, nor even the purpose of the game. These are some significant challenges, though they say a computer game "[...] allows us to examine them in a controlled environment in which we can easily change the problems to be solvable, and then gradually increase the difficulty step by step". However, their proposition does not endorse any specific formalism or system.

Shanahan [20] proposes a logicist agent architecture that incorporates planning, perception, and a sense-plan-act loop, all formalized in the Event Calculus and executed through proof using abductive logic programming. The unified approach makes it possible to proactively deal with unexpected percepts in a robotic mail delivery domain, due to humans unpredictably blocking pathways by closing office doors. The robotic agent is able to intelligently adapt its behaviour by first reasoning about all percepts using abductive proof, forming explanations for sensor values that deviate from expectations in terms of actions by other agents or humans, and then adapting its plans to incorporate the new knowledge. Hierarchical planning is accomplished through the same abductive proof mechanism and allows timely reactions by only instantiating the abstract plan enough to figure out a first action, while leaving the rest a sketchy idea of how to achieve the goal.

Pollock goes further towards general intelligence and differentiates between goal-oriented agents, that solve tasks for which a metric of success can be defined, and anthropomorphic agents, that solve tasks that are too complex for it to be possible to identify such a metric [15]. Such agents must be based on a "general theory of rational cognition", and Pollock's OSCAR agent architecture is an attempt to embody such a theory in an implemented system. The central component is a natural deduction theorem prover for first-order logic that is capable of planning, reasoning about percepts and attaching certainty factors to premises and conclusions. But its most important feature is the mechanism for defeasible reasoning that can be used to deal with default reasoning and the frame problem. Unlike most other formalisms, which are only applicable to problems conforming to explicit restrictions that ensure computability, Pollock's anthropomorphic architecture can be applied to any problem. The inference engine reports solutions based on defeasible assumptions, while a search for evidence contradicting these assumptions continues, for which there can be no guarantee of termination.

Wang's NARS system is similar in that the underlying assumption is the lack of knowledge and resources sufficient to give optimal answers or even any correctness guarantees [21]. Instead, the system continually evaluates the available evidence and may "change its mind" about the best answer to a given query. NARS is based on a novel *categorical logic* that differs significantly from classical first-order logic, incorporates uncertainty, and deals with conflicting evidence at a fundamental level. While clearly aiming towards general intelligence, results to date seem limited to small benchmark problems.

One example of the relatively recent surge of interest in the use of computer games for AI research is the Soar/Games project. They report uncovering new research challenges after coupling the Soar artificial general intelligence architecture to Quake 2 and

Descent 3 [22]. Their emphasis is on generality in their attempts to build reusable rule bases for agent behaviour. Laird's and van Lent's enthusiasm for the use of computer games in AGI research is evident in their paper "Human-level AI's Killer Application: Interactive Computer Games" [23].

Finally, the 1996 computer game Creatures is an example of AI from the game industry rather than of academic origins. Its artificial life forms use neural net "brains" that can be trained through interaction with a human player, learn from interaction with their simulated world, or even from other creatures [24]. The success of Creatures is an affirmation of the possibility of incorporating AI technology into a commercial computer game.

5. Conclusions

We hope to have given the impression that our game concept is far from complete. On the contrary, when working with games interesting problems abound, and many of them call for new research in artificial general intelligence. Some old but still open questions that figure in our work are how to include perception, reasoning, planning, execution, and failure recovery in an integrated agent architecture, what to do about conflicting information, and how to deal with the accumulation of perceptions and knowledge in persistent agents without their reasoning slowing down to a crawl. ANDI-Land is fundamentally a multi agent setting and could involve cooperation between multiple agents, delegation of goals, and intelligent use of communication. These (and more) topics have concrete instantiations in the structure of the game environment that make them easier to think about, discuss, and hopefully to solve.

Traditional AI benchmark problems play an important role in clearly highlighting specific difficulties that any sufficiently general AI system will have to address. Games can serve to complement them by forcing an integrated view of autonomous agents in complex environments, and they possess many positive attributes such as ease of access for both researchers and their peers, variable challenge level ranging all the way from simple puzzle games to wide coverage natural language understanding, and the possibility for applications in the future commercial game industry where academic AI technology has so far failed to migrate (the prototypical exception being A* search).

The demand for human-like computer characters is by itself incentive to study all the key technologies needed for artificial general intelligence, making games an excellent test bed for AGI research. Even some forms of self-awareness would seem to be desirable to agents acting as if they were "real" live inhabitants of some fictional reality game world. Such a setting is a sort of Turing test where human players are not necessarily aware of which characters are artificial and which are other humans. It seems to us that research on game AI could function as a much needed road map towards the fields original vision.

Acknowledgements

This work is supported in part by the National Aeronautics Research Program NFFP04 S4203 and the Strategic Research Center MOVIII, funded by the Swedish Foundation for Strategic Research, SSF.

References

- [1] Patrick J. Hayes. The naive physics manifesto. In Donald Michie, editor, *Expert Systems in the Micro-Electronic Age*, pages 242–270. Edinburgh University Press, 1979.
- [2] Ann Copestake and Dan Flickinger. An open source grammar development environment and broad-coverage English grammar using HPSG. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation LREC 2000*, Athens, Greece, 2000.
- [3] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Englewood Cliffs, NJ, 2nd edition, 2003.
- [4] Stuart M. Shieber. A uniform architecture for parsing and generation. In *Proceedings of the 12th International Conference on Computational Linguistics*, pages 614–619, Budapest, Hungary, 1988.
- [5] Erik Sandewall. *Features and Fluents: The Representation of Knowledge about Dynamical Systems*, volume 1. Oxford University Press, 1994.
- [6] Patrick Doherty. Reasoning about action and change using occlusion. In *Proceedings of the Eleventh European Conference on Artificial Intelligence ECAI'94*, pages 401–405, 1994.
- [7] Patrick Doherty and Jonas Kvarnström. *The Handbook of Knowledge Representation*, chapter 18. Elsevier, 2007. To appear.
- [8] Martin Magnusson. *Deductive Planning and Composite Actions in Temporal Action Logic*. Licentiate thesis, Linköping University, September 2007. <http://www.martinmagnusson.com/publications/magnusson-2007-lic.pdf>.
- [9] Robert Moore. Reasoning about knowledge and action. Technical Report 191, AI Center, SRI International, Menlo Park, CA, October 1980.
- [10] Leora Morgenstern. A first order theory of planning, knowledge, and action. In *Proceedings of the 1986 conference on Theoretical aspects of reasoning about knowledge TARK 1986*, pages 99–114, 1986.
- [11] David Kaplan and Richard Montague. A paradox regained. *Notre Dame Journal of Formal Logic*, 1(3):79–90, 1960.
- [12] Donald Perlis. Languages with self-reference I: Foundations (or: we can have everything in first-order logic). *Artificial Intelligence*, 25(3):301–322, 1985.
- [13] Jim des Rivières and Hector J. Levesque. The consistency of syntactical treatments of knowledge. In *Proceedings of the 1986 conference on Theoretical aspects of reasoning about knowledge TARK 1986*, pages 115–130, 1986.
- [14] Douglas Lenat. CYC: a large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38, 1995.
- [15] John Pollock. Rational cognition in OSCAR. In Nicholas Jennings and Yves Lesperance, editors, *Intelligent Agents VI: Agent Theories, Architectures, and Languages*, volume 1757 of *Lecture Notes in AI*, pages 71–90. Springer Verlag, 2000.
- [16] Lance J. Rips. *The psychology of proof: deductive reasoning in human thinking*. MIT Press, Cambridge, MA, USA, 1994.
- [17] John Pollock. Natural deduction. Technical report, Department of Philosophy, University of Arizona, 1999. <http://www.sambabike.org/ftp/OSCAR-web-page/PAPERS/Natural-Deduction.pdf>.
- [18] Itay Meiri. Combining qualitative and quantitative constraints in temporal reasoning. *Artificial Intelligence*, 87(1–2):343–385, 1996.
- [19] Eyal Amir and Patrick Doyle. Adventure games: a challenge for cognitive robotics. In *AAAI'02 workshop on Cognitive Robotics*, 2002.
- [20] Murray Shanahan. Reinventing shakey. In *Logic-Based Artificial Intelligence*, pages 233–253. Kluwer Academic Publishers, Norwell, MA, USA, 2000.
- [21] Pei Wang. The logic of intelligence. In Ben Goertzel and Cassio Pennachin, editors, *Artificial General Intelligence*, pages 31–62. Springer, 2007.
- [22] John E. Laird and Michael van Lent. Developing an artificial intelligence engine. In *Proceedings of the Game Developers' Conference*, pages 577–588, 1999.
- [23] John E. Laird and Michael van Lent. Human-level AI's killer application: Interactive computer games. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 1171–1178, 2000.
- [24] Stephen Grand, Dave Cliff, and Anil Malhotra. Creatures: Artificial life autonomous software agents for home entertainment. In *Proceedings of the First International Conference on Autonomous Agents (Agents'97)*, pages 22–29, 1997.

Artificial General Intelligence through Large-Scale, Multimodal Bayesian Learning

Brian MILCH^{a,1}

^a CSAIL, Massachusetts Institute of Technology, USA

Abstract. An artificial system that achieves human-level performance on open-domain tasks must have a huge amount of knowledge about the world. We argue that the most feasible way to construct such a system is to let it learn from the large collections of text, images, and video that are available online. More specifically, the system should use a Bayesian probability model to construct hypotheses about both specific objects and events, and general patterns that explain the observed data.

Keywords. probabilistic model, architecture, knowledge acquisition

Introduction

A long-standing goal of artificial intelligence is to build a single system that can answer questions as diverse as, “How can I get from Boston to New Haven without a car?”, “How many Nobel Prizes have been won by people from developing countries?”, and “In this scene showing people on a city street, about how cold is it?”. Answering such questions requires broad knowledge, on topics ranging from public transit to geography to weather-appropriate clothing. These questions also require deep reasoning, not just scanning for keywords or looking at simple features in an image.

So far, we do not have AI systems whose knowledge is both broad and deep enough to answer this range of questions. The most prominent efforts to acquire such knowledge are Cyc [1] and Open Mind [2], both of which have significant limitations. The knowledge they collect is primarily deterministic: it does not include quantitative measures of how often things tend to occur. Furthermore, adding new knowledge requires effort by humans, which limits the breadth of questions that can be answered.

Meanwhile, other branches of AI have focused on reasoning with probabilistic models that explicitly quantify uncertainty [3], and on learning probabilistic models automatically from data [4]. This probabilistic approach to AI has been successful in narrow domains, ranging from gene expression analysis [5] to terrain modeling for autonomous driving [6]. It has also seen domain-independent applications, such as sentence parsing [7] and object recognition [8], but these applications have been relatively shallow: they have not captured enough semantics to answer questions of the kind we posed above.

¹Corresponding Author: Brian Milch, MIT CSAIL, 32 Vassar St. Room 32-G480, Cambridge, MA 02139, USA; E-mail: milch@csail.mit.edu.

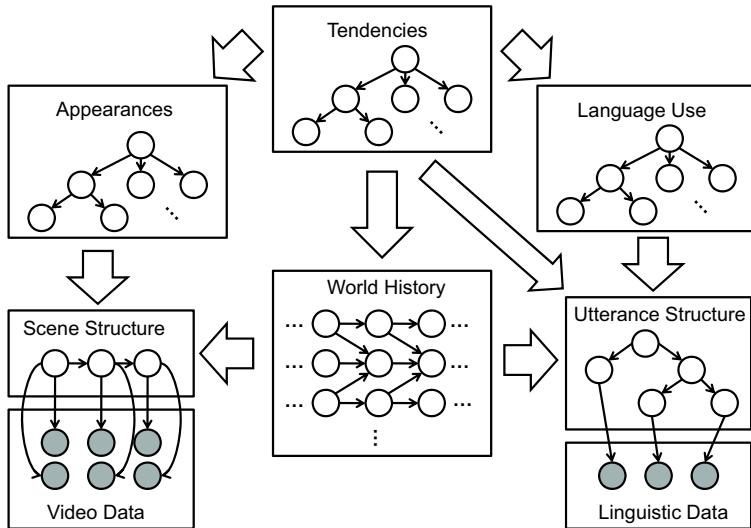


Figure 1. Sketch of a probabilistic graphical model for AGI. Nodes represent random variables and arrows represent probabilistic influences.

Thus, for the past two decades there has been an unfortunate divide between the probabilistic AI community and those actively working on deep, open-domain AI — what Goertzel has called *artificial general intelligence* or *AGI* [9]. There are some signs that this gap is narrowing: the Novamente Cognition Engine [9] is an AGI system that does probabilistic reasoning, and the probabilistic AI community is showing renewed interest in large-scale knowledge acquisition [10,11].

The purpose of this paper is to bridge this gap more completely. We propose a Bayesian AGI system that takes a large chunk of the web as input, including both hypertext and video data, and uses a probabilistic model to make inferences about what the world must be like. We also comment on integrating AGI into the ecosystem of probabilistic AI research.

1. A Bayesian Architecture for AGI

Figure 1 shows a schematic view of a probabilistic model suitable for an AGI system. The shaded nodes represent variables that the system can observe: pixel values in videos, and words in linguistic data. The unshaded nodes represent unobserved variables about which the system can form hypotheses. The variables in the top box represent general tendencies in the world, such as the way public transit tends to work, and the tendency of people to wear wool hats and scarves only in cold weather. The “world history” box contains variables representing specific facts about the world: for instance, the current public transit schedules between New Haven and Boston, and the history of Nobel prizes. The large arrow from the “tendencies” box to “world history” indicates that general tendencies influence the concrete facts. There are also arrows within the world history box, representing causal links between events and their successors.

In the top left and top right boxes, there are variables for how things tend to appear, and how people tend to use language. Finally, there are variables representing the 3-d

structures that underlie particular scenes and the syntactic structures that underlie particular utterances. These variables depend on the facts in the world: a scene or utterance typically provides a window on some piece of the world history. The additional arrow from tendencies to utterance structure allows utterances to be directly about tendencies: for instance, someone could say, “People wear scarves when it’s cold”. We discuss parts of this model in more detail in Sections 3 and 4.

Our system’s probability model can be seen as a joint distribution over all these variables. It can also be understood as a hierarchy of models: a hypothesis about the variables in the top three boxes defines the structure and parameters of a probability model for world histories, scenes, and utterances. In using a top-level probability model to define a distribution over sub-models, we are taking a Bayesian approach [12].

Bayesian modeling is attractive because it makes an automatic trade-off between the complexity of a hypothesized sub-model and how well it fits the data. Complex hypotheses may fit the data better, but they typically have lower prior probabilities than simple ones. Furthermore, hypotheses involving many free parameters are penalized by a Bayesian Ockham’s razor effect [13]: it is unlikely that all those parameters would happen to have values that match the data. Although similar trade-offs arise from Kolmogorov complexity [14] and the minimum description length principle [15], the Bayesian approach lets us express prior knowledge about the structure of the world, and also maintain uncertainty about sub-models rather than choosing a single best hypothesis. There are also close connections with Bayesian approaches in statistics [12], epistemology [16] and cognitive science [17].

2. Training Data

There are several ways in which one could hope to train a Bayesian AGI system. One is to train it solely on text, a huge amount of which is available online. However, this would leave the system unable to answer queries about images or video scenes. Furthermore, there is no evidence that it is possible to build a deep model of the world from linguistic input alone: human children all have access to visual, or at least tactile, input.

Alternatively, we might train our AGI system by hooking it up to one or more mobile robots. This would have several advantages: the system could learn from continuous streams of experience with multiple sensors, and it could actively move around and manipulate its environment to reduce its uncertainty. On the other hand, a robot’s experience would be fairly narrow, limited to the environments where its owners let it move around.

Fortunately, given the explosive growth of online video, we no longer have to choose between a large, diverse data set and a multimodal one. Videos available online include many everyday objects and activities (sometimes in the background), often correlated with speech. Meanwhile, the text on the web covers many kinds of objects and phenomena that are not readily visible in videos. Thus, we propose to train our AGI system on a web data set that includes hypertext, images, and video.

In contrast to a human child or a robot, this system still will not have a chance to actively explore its world. But acquiring multimodal data from the web promises to be easier and less expensive than acquiring it with a fleet of mobile robots — and even human children learn a lot from non-interactive media such as books, television, and movies.

3. Built-In Model Components

Although Figure 1 provides a rough sketch, the question of how to define the probability model for a Bayesian AGI system is still largely open. Some authors have argued that we should be able to use generic models that encode little or no prior knowledge about the physical world [18]. But finding good hypotheses to explain the world is a daunting search problem; to the extent that we can help our system along, it seems that we should do so. After all, it is unlikely that human children start from scratch: we seem to have an evolved infrastructure for interpreting visual input, understanding language, and interacting with other people.

Modeling the physical world is one area where we should be able to provide our AGI system with a good running start. The computer graphics and animation communities have done a great deal of work on modeling the world with enough fidelity to create compelling visual environments. Our appearance sub-models should be able to use standard mesh data structures for representing the 3-d shapes of objects, and also borrow skeleton-based techniques for modeling articulated objects. The system will still need to infer the meshes and skeletons of objects (and classes of objects) from data, but at least it will not need to invent its own approach to representing shape and movement.

It also makes sense to build in components for dealing with speech, hypertext, and language in general. The language-use sub-model should contain built-in representations for words and parse trees. As with the appearance model, all the specific content — the grammars of particular languages, and the words and phrases people tend to use for particular meanings — should be inferred by the system. But there is no reason for the system to re-invent the basic ideas of computational linguistics.

Another area that calls out for built-in infrastructure is reasoning about the beliefs and desires of other agents. This ability is crucial both for understanding human behavior in the training data, and for giving helpful answers to users' questions. The system must be able to put itself in another agent's shoes, using its own inference and planning capabilities to reason about what another agent probably believes and what it might do to achieve its goals. Thus the system's probability model must have some characteristics of a probabilistic epistemic logic theory [19] or a multi-agent influence diagram [20]. It is particularly hard to imagine how a Bayesian AGI system might invent this intentional view of other agents if it were not built in.

4. Learned Model Components

Of course, even with built-in capacities for physical, linguistic, and multi-agent reasoning, our system will have a lot to learn. It will lack such basic notions as eating, smiling, driving, owning objects, having a disease, and so on. Thus, the overall probability model must admit indefinitely many non-built-in concepts.

Fortunately, there has recently been a surge in research on probability models with an unbounded number of latent concepts, using a non-parametric Bayesian construct called the Dirichlet process (DP) mixture model [21,22,23]. A DP mixture can be thought of as a graphical model where the nodes are indexed by an infinite collection of objects, and these latent objects are “recruited” as necessary to explain the data. The Bayesian Ockham's razor effect [13] prevents the model from hypothesizing a separate latent object for

each data point. DP mixtures can serve as building blocks in quite complicated models, including those that model the relations among objects as well as their properties [24].

In addition to hypothesizing new concepts, our system must learn how they relate to one another. Often, these dependencies are probabilistic: for example, a person’s clothing choices depend probabilistically on the weather, and whether a person knows French depends probabilistically on where that person grew up. There has been considerable work on learning the dependency structure of probabilistic graphical models [25]. However, just learning dependencies among individual variables — Laura’s clothing choices depend on the weather in New York, Mary’s choices depend on the weather in London, etc. — is not sufficient. Instead, our system must learn dependency models that apply to whole classes of objects, specifying how probabilistic dependencies arise from underlying relationships (such as the relationship between a person and the city where she lives). Learning such dependency models is the domain of a young field called statistical relational learning [26]. Considerable progress has been made on learning certain classes of relational dependencies [27,28], although learning models in highly expressive languages such as Bayesian logic [29] remains an open problem.

5. Algorithms

Because we are taking a Bayesian approach, our proposed AGI system does make a strict separation between reasoning and learning: it learns by probabilistic inference. But of course, probabilistic inference can be very computationally expensive (a naive approach to answering a query would involve summing or integrating over all the unobserved, non-query variables). We have no silver bullet for this problem, but there are indications that performing inference on an AGI model is not as ridiculous as it may sound.

The first thing to note is that we will be satisfied with approximate inference methods, even if they do not provide bounds on the quality of the approximation — we will be able to judge if we are doing well enough based on how well the system answers queries. The major approaches to approximate inference today are Markov chain Monte Carlo (MCMC) [30], mean-field-based variational methods [31,32], and belief propagation methods [33,34]. These methods can take hours to run on sets of just a few hundred documents or images, so one might object that using them on a large chunk of the web is infeasible. But given a hypothesis about the world history and the appearance and language-use sub-models, the individual videos and documents in our training set are conditionally independent of each other. Thus, the work of parsing these data items can be parallelized across a large set of machines.

Perhaps the greatest challenge for inference in an AGI model is preventing the system from considering too many variables when evaluating a hypothesis or answering a query. For instance, consider trying to judge the ambient temperature in a street scene showing just one man, who is wearing a winter coat but no hat or scarf. In principle, a multitude of variables are relevant to judging the temperature: Is this person just stepping outside briefly, or is it warm enough that he can take a long walk in these clothes? What is his reason for being outside? Is he a local, or is he visiting from someplace warmer or colder? These variables are all within the purview of an AGI system — and indeed, with different data or a different query, it might be important to reason about them. But in this case, reasoning about them is bound to be unproductive: the system will not be

able to conclude anything strong enough about these variables to influence the query result. Thus, the inference algorithm must be able to deal with *partial* hypotheses, and sum out the remaining random variables without reasoning about them explicitly. Existing algorithms can exploit situations where certain variables are strictly irrelevant given the hypothesized values of other variables [35]. There are also algorithms that exploit interchangeability among objects to sum out large sets of variables in a single step [36,37]. But dealing with queries like the one above will require novel methods for exploiting approximate irrelevance and approximate symmetry among variables.

Another relevant line of research is the integration of probabilistic inference with logical reasoning. Many dependencies are nearly deterministic, and our system should be able to exploit near-determinism to speed up reasoning (by contrast, standard approximate inference algorithms tend to get stuck in local optima when determinism is present). One approach is to reduce probabilistic inference to weighted model-counting in propositional logic, and then exploit fast, deterministic SAT algorithms [38]. Another approach combines MCMC with SAT algorithms based on stochastic local search [39]. But for a full-blown AGI system, new inference algorithms are sure to be necessary. One positive side-effect of working with a theoretically simple Bayesian model is that the resulting inference problems can serve as fodder for the approximate inference community. It should even be possible to extract sub-networks from the AGI model to use as highly relevant benchmark problems.

6. Measures of Progress

In order for Bayesian AGI to be viable as a research program, there must be some ways of evaluating its progress — short of seeing it pass a Turing test. One advantage of training on open-domain web data, as opposed to in an artificial or restricted environment, is that the system should quickly gain the ability to at least *attempt* open-domain tasks of practical interest. Several such tasks are used for evaluation in the computer vision and natural language fields, including the CalTech-256 object recognition data set [8], the MUC-6 noun phrase coreference data set², the TREC question answering evaluation [40], and the PASCAL textual entailment challenge [41]. It is unlikely that the AGI system will immediately beat state-of-the-art but shallower approaches to these tasks. However, we should at least be able to measure our system’s performance and see it improving year by year.

Another way in which a Bayesian AGI system could succeed early on is by becoming a resource for shallower but more finely tuned systems. For instance, some current natural language systems use WordNet [42] or Cyc [1] to define features for machine learning algorithms, or to implement one part of a multi-stage heuristic system. If our AGI system came to play a similar role and improved the performance of the systems that used it, we would be providing a useful service. Eventually, of course, we would hope for the AGI system to surpass shallower systems that used it as a resource.

²<http://cs.nyu.edu/cs/faculty/grishman/muc6.html>

7. Conclusion

We have argued that Bayesian learning from large quantities of multimodal web data is the most feasible path toward artificial general intelligence. This position involves several strong claims: that an AGI system must learn most of its knowledge but should exploit built-in subsystems for visual, linguistic, and multi-agent reasoning; that learning from text alone is insufficient but text plus video is enough; and that Bayesianism provides an adequate theoretical foundation for AGI. These claims will be put to the test in a Bayesian AGI project.

This paper is by no means a complete design for an AGI system. The process of designing the probabilistic model, and algorithms to do inference on it, promises to involve much trial and error over a decade or two. But there seems to be a good chance of success. Furthermore, the existence of such a project (or more than one!) would be a boon to the probabilistic AI community, serving as a source of motivating problems and a testbed for new techniques. A flourishing Bayesian AGI project would bridge the disturbing gap between the successes of probabilistic AI and the goal of understanding deep, general intelligence.

References

- [1] D. B. Lenat. CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38, 1995.
- [2] P. Singh, T. Lin, E. T. Mueller, G. Lim, T. Perkins, and W. L. Zhu. Open Mind Common Sense: Knowledge acquisition from the general public. In *Proc. 1st Int'l Conf. on Ontologies, Databases, and Applications of Semantics for Large Scale Information Systems*, 2002.
- [3] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Francisco, revised edition, 1988.
- [4] M. I. Jordan, editor. *Learning in Graphical Models*. MIT Press, Cambridge, MA, 1998.
- [5] E. Segal, B. Taskar, A. Gasch, N. Friedman, and D. Koller. Rich probabilistic models for gene expression. *Bioinformatics*, 17(Suppl 1):S243–252, 2003.
- [6] S. Thrun, M. Montemerlo, and A. Aron. Probabilistic terrain analysis for high-speed desert driving. In *Proc. Robotics Science and Systems Conf.*, 2006.
- [7] M. Collins. Three generative, lexicalised models for statistical parsing. In *Proc. 35th Annual Meeting of the Assoc. for Computational Linguistics*, 1997.
- [8] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.
- [9] B. Goertzel. Patterns, hypergraphs and embodied general intelligence. In *IEEE World Congress on Computational Intelligence: Panel Discussion on “A Roadmap to Human-Level Intelligence”*, 2006.
- [10] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the Web. In *Proc. 20th Int'l Conf. on Artificial Intelligence*, 2007.
- [11] W. Pentney, M. Philipose, J. Bilmes, and H. Kautz. Learning large-scale common sense models of everyday life. In *Proc. 22nd AAAI Conf. on Artificial Intelligence*, 2007.
- [12] C. P. Robert. *The Bayesian Choice*. Springer, New York, 2nd edition, 2001.
- [13] E. T. Jaynes. Inference, method and decision: Towards a Bayesian philosophy of science (book review). *J. Amer. Stat. Assoc.*, 74(367):740–741, 1979.
- [14] M. Li and P. Vitanyi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer, 2nd edition, 1997.
- [15] J. Rissanen. *Information and Complexity in Statistical Modeling*. Springer, 2007.
- [16] C. Howson and P. Urbach. *Scientific Reasoning: The Bayesian Approach*. Open Court, La Salle, IL, 1989.
- [17] A. Gopnik and J. B. Tenenbaum. Bayesian networks, Bayesian learning, and cognitive development. *Developmental Science*, 10(3):281–287, 2007.

- [18] M. Hutter. *Universal Artificial Intelligence: Sequential Decisions Based on Algorithmic Probability*. Springer, 2005.
- [19] R. Fagin and J. Y. Halpern. Reasoning about knowledge and probability. *J. Assoc. for Computing Machinery*, 41(2):340–367, 1994.
- [20] D. Koller and B. Milch. Multi-agent influence diagrams for representing and solving games. *Games and Economic Behavior*, 45(1):181–221, 2003.
- [21] T. S. Ferguson. Bayesian density estimation by mixtures of normal distributions. In M. H. Rizvi, J. S. Rustagi, and D. Siegmund, editors, *Recent Advances in Statistics: Papers in Honor of Herman Chernoff on His Sixtieth Birthday*, pages 287–302. Academic Press, New York, 1983.
- [22] R. M. Neal. Markov chain sampling methods for Dirichlet process mixture models. *J. Computational and Graphical Statistics*, 9:249–265, 2000.
- [23] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *J. Amer. Statist. Assoc.*, 101:1566–1581, 2006.
- [24] C. Kemp, J. B. Tenenbaum, T. L. Griffiths, T. Yamada, and N. Ueda. Learning systems of concepts with an infinite relational model. In *Proc. 21st AAAI National Conf. on Artificial Intelligence*, 2006.
- [25] D. Heckerman, D. Geiger, and D. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20:197–243, 1995.
- [26] L. Getoor and B. Taskar, editors. *Introduction to Statistical Relational Learning*. MIT Press, 2007.
- [27] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *Proc. 16th International Joint Conference on Artificial Intelligence*, pages 1300–1307, 1999.
- [28] S. Kok and P. Domingos. Learning the structure of Markov logic networks. In *Proc. 22nd International Conf. on Machine Learning*, pages 441–448, 2005.
- [29] B. Milch, B. Marthi, S. Russell, D. Sontag, D. L. Ong, and A. Kolobov. BLOG: Probabilistic models with unknown objects. In *Proc. 19th International Joint Conference on Artificial Intelligence*, pages 1352–1359, 2005.
- [30] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, editors. *Markov Chain Monte Carlo in Practice*. Chapman and Hall, London, 1996.
- [31] M. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul. Introduction to variational methods for graphical models. *Machine Learning*, 37:183–233, 1999.
- [32] E. P. Xing, M. I. Jordan, and S. Russell. A generalized mean field algorithm for variational inference in exponential families. In *Proc. 19th Conf. on Uncertainty in Artificial Intelligence*, pages 583–591, 2003.
- [33] K. P. Murphy, Y. Weiss, and M. I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proc. 15th Conf. on Uncertainty in Artificial Intelligence*, pages 467–475, 1999.
- [34] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Generalized belief propagation. In *Advances in Neural Information Processing Systems 13*, pages 689–695. MIT Press, Cambridge, MA, 2001.
- [35] B. Milch and S. Russell. General-purpose MCMC inference over relational structures. In *Proc. 22nd Conf. on Uncertainty in Artificial Intelligence*, pages 349–358, 2006.
- [36] D. Poole. First-order probabilistic inference. In *Proc. 18th International Joint Conference on Artificial Intelligence*, pages 985–991, 2003.
- [37] R. de Salvo Braz, E. Amir, and D. Roth. Lifted first-order probabilistic inference. In *Proc. 19th International Joint Conference on Artificial Intelligence*, pages 1319–1325, 2005.
- [38] T. Sang, P. Beame, and H. Kautz. Performing Bayesian inference by weighted model counting. In *Proc. 20th AAAI National Conf. on Artificial Intelligence*, pages 475–482, 2005.
- [39] H. Poon and P. Domingos. Sound and efficient inference with probabilistic and deterministic dependencies. In *Proc. 21st AAAI National Conf. on Artificial Intelligence*, pages 458–463, 2006.
- [40] E. M. Voorhees. Overview of TREC 2004. In *The Thirteenth Text Retrieval Conference*. NIST, 2004.
- [41] I. Dagan, O. Glickman, and B. Magnini. The PASCAL recognising textual entailment challenge. In *Proc. PASCAL Challenges Workshop on Recognising Textual Entailment*, 2005.
- [42] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

A computational approximation to the AIXI model

Sergey PANKOV

*National High Magnetic Field Laboratory
Florida State University
Tallahassee, FL, 32306, USA*

Abstract.

Universal induction solves in principle the problem of choosing a prior to achieve optimal inductive inference. The AIXI theory, which combines control theory and universal induction, solves in principle the problem of optimal behavior of an intelligent agent. A practically most important and very challenging problem is to find a computationally efficient (if not optimal) approximation for the optimal but incomputable AIXI theory. We propose such an approximation based on a Monte Carlo algorithm that samples programs according to their algorithmic probability. The approach is specifically designed to deal with real world problems (the agent processes observed data and makes plans over range of divergent time scales) under limited computational resources.

Keywords. AIXI, Solomonoff induction, Kolmogorov complexity, Monte Carlo, algorithmic probability, control theory, intelligent agent

Introduction

There exists a universally optimal theory of artificial intelligence (AI), the AIXI theory [1,2], which combines universal induction [3] with control theory [4]. The AIXI deals with optimal performance (in terms of maximizing its reward) of an agent within an environment in the context of unlimited computational resources. Computationally universally optimal approaches have also been discussed, such as AIXItl [1,2] (which is an approximation to AIXI) or the self-referential Goedel machine [5]. Both approaches are asymptotically optimal, meaning that no other method can possibly outperform (in terms of required computational resources) them in the long run, up to a non-universal additive constant, which may be huge in practice. Both approaches use search for a proof that would justify a particular choice of an agent's strategy or a modification thereof. In the AIXItl the search algorithm is hard-coded, while the Goedel machine permits rewriting of its search algorithm itself, if a provably better algorithm is found. This ability to self-improve (rewrite any part of its code) may significantly reduce the above mentioned non-universal constant.

One could think [5] of using the AIXItl model to initialize the Goedel machine (that is to use it as an initial problem solver of the Goedel machine), which will

eventually self-improve far beyond the initial AIXItl. However, a few legitimate practical questions arise (if one to have in mind complex real world environments), such as: how long might one wait before a significant self-improvement happens? Will not the hidden constants render that time impractically long? Or, should not the initial solver, for example AIXItl, be (at least) "comparably intelligent" to its creator to begin with, if a significant self-improvement were to be expected in the foreseeable future?

Therefore, giving up computational universality would be justifiable at the current stage, provided one gains in low computational complexity on a certain class of problems, deemed (subjectively) important to the field of AI, or artificial general intelligence (AGI). The real value of the AIXI theory is that it provides a prescription for optimal (fastest in the number of agent's observations and actions) way of learning and exploiting the environment. This is analogous to how Solomonoff induction (which, like AIXI, is incomputable), gives a prescription for optimal (fastest in the number of observations) inductive inference. We, therefore, believe that any reasonable computational model of intelligence must recover the AIXI model in the limit of infinite computational resources.

In this paper we present a particular Monte Carlo (MC) algorithm for computational implementation of Solomonoff induction. The algorithm samples candidate programs, that produce a desirable output, according to their algorithmic probability. The idea behind the MC algorithm is to make a Turing machine run "in reverse", from its output to the input program. This guarantees finding some candidate programs, even with very limited computational resources. The agent then uses the discovered programs to predict its possible future observations and corresponding rewards (with predictions possibly reaching far ahead in time). These predicted outcomes are used by the agent to choose its immediate actions, employing a generalized expectation maximization procedure. We argue that our approach reduces to the AIXI model in the limit of infinite computational resources. Finally we discuss what representation of a Turing machine should be most suitable for the approach to be applicable to the real world problems.

The paper is organized as follows. In section 1, we introduce notations and basic concepts related to the universal induction and the AIXI. The main idea of our Monte Carlo algorithm is explained in section 2 in a simple setting of a conventional Turing machine (TM). In section 3, we generalize expectation maximization procedure to long term policies sampled probabilistically under limited computational resources. The general assumptions we make about possible environments relevant to AGI are discussed in section 4. In section 5, we outline a new TM representation, which is meant to accommodate the assumptions made in the previous section.

1. Basics of Solomonoff induction and AIXI model

In this section we introduce notations and briefly overview basic theoretical concepts that are used in this paper.

1.1. Notations

We use strings of symbols as an abstraction for various information transfer processes. Individual symbols in a string drawn from the alphabet \mathcal{X} are denoted x_i , with the subscript indicating their placement in the string. A string of length $l + 1$ may be written in one of the following ways: $x_m x_{m+1} \dots x_{m+l}$, $x_{m:m+l}$ or implicitly \mathbf{x} , if the subscript is not needed or its range is understood implicitly. The letter $p \equiv \mathbf{p}$ is reserved for a program string. In some cases we enumerate pairs of symbols (for example $x_1 y_1 x_2 y_2 \dots$), then a shorthand notation may be used $xy_i \equiv x_i y_i$, and correspondingly $xy_{i:j} = x_i y_i x_{i+1} \dots x_j y_j = x_i y_i x_{i+1} y_{i+1} \dots x_j y_j$. One should not confuse \mathbf{xy} and \mathbf{xy} : $x_{1:m} y_{m+1:n} = \mathbf{xy}$ while $xy_{1:m} = \mathbf{xy}$. The symbol $*$ represents any substring. The subscript in \mathbf{x}_i indicates string enumeration, not enumeration of individual symbols. The length of a binary representation of a string \mathbf{x} is denoted $l(\mathbf{x})$, so for $\mathcal{X} = \{0, 1\}$ one has $l(x_{1:m}) = m$. Probability of a string $x_{1:m}$ to be sampled from a probability distribution μ is denoted $\mu(x_{1:m})$. Conditional probability for a string $x_{1:n}$ to occur given its first $m < n$ symbols $x_{1:m}$ is defined as $\mu(x_{1:n}|x_{1:m}) \equiv \mu(x_{1:n})/\mu(x_{1:m})$.

1.2. Solomonoff induction

Consider the problem of predicting the symbol x_i given the knowledge of previous $i - 1$ symbols $x_{1:i-1}$ in a string sampled from some distribution. The Bayesian inference solves this problem, assuming that the prior probabilities of possible distributions generating the string are known. The problem of choosing the prior probabilities in an optimal way is solved [3] by the universal prior, also called Solomonoff prior. Solomonoff's universal prior $M(\mathbf{x})$ is defined as:

$$M(\mathbf{x}) = \sum_{p: U(p) = \mathbf{x}*} 2^{-l(p)} \quad (1)$$

The summation is over all possible programs p , which output strings starting with \mathbf{x} when run on a universal Turing machine U (see section 2). The inference with the universal prior is called Solomonoff induction (or universal induction). One of the central properties of universal induction is that the (expected) number of incorrectly predicted bits is bounded by the Kolmogorov complexity $K(\rho)$ of the generating distribution function ρ [6,7], (the Kolmogorov complexity [8] is defined, loosely speaking, as the shortest possible description of an object one can produce using a universal Turing machine - mathematical formalization of a general purpose computer). The universal prior can be equivalently presented [2] as a sum over all (semi-)computable (semi-)measures (distribution functions), weighted by their algorithmic probability ($\sim 2^{-K}$):

$$M(\mathbf{x}) \sim \xi(\mathbf{x}) = \sum_{\rho} 2^{-K(\rho)} \rho(\mathbf{x}) \quad (2)$$

1.3. AIXI

All or nearly all formalizable problems in the AI field can be cast in a form of an agent interacting with an environment and seeking to maximize its reward. The agent interacts with its environment in cycles: in i th cycle the agent receives a signal x_i from the environment, processes it, then sends a signal y_i (also called action) to the environment. The subscript i is also referred to as time. With every signal x_i comes a reward $v(x_i)$. Assume for now that the agent is in the beginning of the i th cycle and wants to maximize its expected reward for the next k cycles ahead in the environment described by a known measure μ [4]. To compute the maximized expected reward V_{ik}^* (counting the reward from cycle i to $i + k$), we introduce an auxiliary function V_{ik}^+ defined recursively as:

$$V_{ik}^+(xy_{i:j-1}) = \sum_{x_j} \mu(xy_{1:j-1}x_j|xy_{1:j-1}) \max_{y_j} V_{ik}^+(xy_{i:j}) \quad (3)$$

where $i < j \leq i + k$ and $V_{ik}^+(xy_{i:i+k}) = \sum_{j=i}^{j=i+k} v(x_j)$. Then $V_{ik}^*(x_i) = \max_{y_i} V_{ik}^+(xy_i)$. If the measure μ is unknown, the AIXI's prescription is to substitute it with ξ defined in Eq.(2).

2. Monte Carlo algorithm for a "conventional" Turing machine

By a conventional Turing machine we imply a machine described below. This should be contrasted with an effective representation of a Turing machine, discussed in section 5.

A Turing machine [9] is a computational device consisting of a read/write head and an infinite tape (or possibly several tapes). The head assumes some state $s \in S$ and is capable of simple actions, like (reading)writing a symbol $x \in \mathcal{X}$ (from)onto the tape, moving left or right, halting. The number of states $|S|$ and the size of the alphabet \mathcal{X} are finite. The machine's computation is governed by some fixed rules $F : \mathcal{X} \times S \rightarrow S \times A$, which defines a step of TM computation as follows. In a state s the machine reads a symbol x , maps the pair (x, s) to (s', a) according to F , assumes new state s' and performs the action $a \in A$. A machine capable of simulating any other machine is called universal and is denoted U .

In Solomonoff/AIXI theory one needs to sample programs p , producing a desired output $\mathbf{x}* = U(p)$, according to the probability $2^{-l(p)}$. In the following we devise a Monte Carlo algorithm for computing time-bound universal prior $M_\tau(\mathbf{x})$, which is constructed analogous to $M(\mathbf{x})$ (Eq.1) but from programs whose runtime does not exceed τ steps of a Turing machine computation.

Note that Solomonoff prior in Eq.(1) is defined via a monotone machine U (see section 3). In a simplified example considered in this section the TM will not be monotone, and only halting programs will be considered. Therefore, the sampled probability will not quite correspond to Eq.(1), and we will consider a single tape machine for illustrative purpose only. However, in the next section we will use monotone machines and will include non-halting computations in consideration, thus restoring exact correspondence to Solomonoff prior.

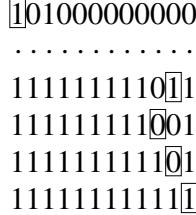


Figure 1. Figure represents a computation of a hypothetical single tape Turing machine, outputting a string of 1s. Each line shows content of the tape as computation progresses (from top to bottom). The transparent box represents the TM's head. The program length $l(p) = 3$. Only the first and a few concluding steps are shown. The machine halts in the last line.

For simplicity we consider a single tape machine with $\mathcal{X} = \{0, 1\}$. The string \mathbf{x}^τ represents the content of the tape in the τ th step of machine's computation. An action $a = (x_t, \Delta t)$ consists of writing x_t in the head's position t , then moving to a position $t' = t + \Delta t$, where $\Delta t = \pm 1$, (then reading a new symbol x'_t , that is understood implicitly). Fig.(1) shows a run of a hypothetical program on such a machine, outputting a string of 1s. The head is initially at $t = 1$ and in the last step τ_n it is at the end of the outputted string.

The triplet $C = (s, \mathbf{x}^\tau, t)$ completely describes the Turing machine in the τ th step and is called its τ th configuration, (we may also indicate the step index explicitly, as C_τ). We consider a version of a Markov chain Monte Carlo algorithm, known as Metropolis-Hastings algorithm [10,11]. Given a positive function $P(C)$, the algorithm will sample TM configurations with relative probability $P(C)/P(C')$, if run sufficiently long. Choosing

$$P(C = (s, \mathbf{x}, t)) = 2^{-l(\mathbf{x})}, \quad (4)$$

will then achieve our goal of sampling C s with their algorithmic probabilities, that is $2^{-l(\mathbf{x})}$. In our hypothetical example in Fig.(1) the leftmost and rightmost 1s can be thought of as the string \mathbf{x} delimiters, indicating the string's beginning and end.

The Monte Carlo procedure works as follows. Given a configuration C , a move to a configuration C' is proposed according to the distribution function (called proposal density) $Q(C' = (s', \mathbf{x}'^\tau, t')|C = (s, \mathbf{x}^\tau, t))$ defined as:

$$Q(C'|C) = \frac{\tilde{Q}(C'|C)}{\sum_{C'} \tilde{Q}(C'|C)},$$

$$\tilde{Q}(C'|C) = \begin{cases} 1, & \text{for } \begin{cases} \tau_{max} > \tau' = \tau + 1 > 1, & F(x_t^\tau, s) = (s', a' = (x_t^{\tau'}, t' - t)), \\ \tau_{max} > \tau = \tau' + 1 > 1, & F(x_{t'}^{\tau'}, s') = (s, a = (x_t^\tau, t - t')), \end{cases} \\ 0, & \text{otherwise,} \end{cases} \quad (5)$$

where τ_{max} is the limit on runtime of a program. The proposed move is then accepted with probability:

$$\min \left\{ \frac{P(C')}{P(C)} \frac{Q(C|C')}{Q(C'|C)}, 1 \right\}. \quad (6)$$

Proposing a MC move and then accepting or rejecting it constitutes a MC step. One can show that the prescribed MC procedure will sample C correctly provided: 1) $P(C)$ is invariant under a MC step (follows directly from Eq.(6)), 2) the procedure is ergodic (in practice it means that any C and C' can be connected by a series of MC moves).

The algorithm starts with $C = (s_h, \mathbf{x}^{\tau_{max}} = \mathbf{x}, t = l(\mathbf{x}))$, where s_h is the halting state and \mathbf{x} is program's desired output. It is assumed that the program starts at $t = 1$ and halts with the head over the last symbol of \mathbf{x} . The rationale of the construction of $Q(C'|C)$ as shown in Eq.(5) is that for any current C_τ the sequence of $C_{\tau'}$ for $\tau \leq \tau' \leq \tau_{max}$ represents (as in Fig.(1)) a computation of the Turing machine outputting \mathbf{x} . Thus for any C_τ with $t = 1$, corresponding \mathbf{x}^τ is a program p outputting \mathbf{x} . These programs are sampled with frequencies proportional to program's algorithmic probabilities. Notice that the time-bound universal prior can then be evaluated as $M_{\tau_{max}} = 1/\langle 2^{l(p)} \rangle_{MC}$, where $\langle \dots \rangle_{MC}$ is the averaging over the Monte Carlo samples.

We would like to note that one can slightly redefine how a program is supplied to the TM, so that \mathbf{x}^τ in any configuration C_τ could be regarded as a program (and not only when $t = 1$). Such redefinition will affect the sampling probability by at most a factor of $l(\mathbf{x})$. What one gains is that the MC simulation is guaranteed to produce a candidate program after any number of MC steps. This viewpoint of the algorithm is adopted in section 5.

Our method should be contrasted with a conceptually very different approach of Schmidhuber [12,13], in which one also samples candidate programs according to their algorithmic probability for the purpose of inductive inference. In that approach one uses the speed prior - a time bounded version of the universal prior. The speed prior takes into consideration computational complexity of a program, by allocating computational resources (time) in proportion to its algorithmic probability. So a program p of length $l(p)$ taking time $t(p)$ to run is sampled with frequency $\propto 2^{-l(p)}/t(p)$. This implies that the speed prior sampling (closely related to Levin's universal search [14]) will take time $\mathcal{O}(2^{l(p)}t(p))$ to sample a program p . And although the approach is optimal for finding the simplest program (here, simplest = minimal $l(p) + \log_2 t(p)$), the search time becomes prohibitively large if the simplest program producing a desired output is not short.

Similarly to the speed prior sampling, our MC method will also sample short programs more frequently. The difference is that the correct algorithmic probability is only guaranteed in the long run, because the output string \mathbf{x} may be much longer than the program p outputting it. In such a case the initial configurations C will be highly unlikely in the long run of the MC. This initial stage of sampling unlikely configurations can be viewed as a search for short programs (or at least sufficiently short; here and throughout the paper "sufficiently short" = sufficiently short for discovering interesting regularities), dominating the sampling in the long run. The MC search, which is not universally optimal, in certain cases will quickly find a sufficiently short program. In which cases this is possible will depend on ruggedness of the search landscape, that is on how much $l(\mathbf{x}_\tau)$ varies

along the MC chains (chain = sequence of MC steps). The crucial idea is that the landscape depends on the choice of the Turing machine. We will hypothesize that there is a particular machine choice that makes the MC search very efficient on an important class of problems.

3. Policy computation

We will not discuss computational efficiency of our approach in this paper, we only notice that it is expected to work fast in certain environments described in section 4. For now we simply assume that the MC succeeds in finding sufficiently short programs outputting the data observed by the agent.

A ubiquitous problem faced by the agent is the exponentially large space of states. Only an exponentially small fraction of the states can be sampled in practice when computing the agent's policy. And because in a typical environment of interest (see section 4) an exponentially small fraction of the most probable states have a combined probability close to 1, there should be a strong bias toward those states. It is easy to check that to achieve the quickest convergence in averaging some function $\phi(x)$ over the states x distributed according to $P(x)$, one should sample the states with probability $\propto (|\phi(x) - \bar{\phi}|P(x))^{2/3}$, where $\bar{\phi}$ is the expectation value of $\phi(x)$. In our problem we only know how to efficiently sample $\propto P(x)$ (as will be shown below), but this still gives (typically exponentially) faster convergence than with the uniform sampling. Generally, the strategy of favoring samples that potentially contribute the most to a quantity of interest, is called importance sampling.

The AIXI agent essentially uses Solomonoff induction to predict future and computes its expected reward from that prediction. The universal prior (Eq.(1)) is defined in terms of a (universal) monotone Turing machine [8]. A monotone machine has a unidirectional (unidirectional = head moves in one direction only, to the right) input and output tapes, which are used for reading a program and writing the output of a computation, respectively. The machine also uses internal tapes for computations. The time bounded universal prior $M_\tau(\mathbf{x})$ is now defined in terms of minimal programs (minimal program = input string to the left of the input head when the last symbol of the output string is being printed) that output \mathbf{x} within the time bound τ .

Consider the problem of sampling extensions \mathbf{y} of length l of a given string \mathbf{x} with probability $M_\tau(\mathbf{xy}|\mathbf{x})$. The solution is obvious: first sample a minimal program p outputting \mathbf{x} according to its algorithmic probability, then use fair coin flips to extend the program as far as needed to output l symbols more (within the time bound). How can this be realized within our MC procedure? First, one generalizes the MC to monotone machines. This is straightforward, we just need to consider more complex configurations $C = (s, \mathbf{x}_1 \times \dots \times \mathbf{x}_n, t_1 \times \dots \times t_n)$ for an n -tape machine, and choose rules F so that the monotone machine is represented appropriately. Second, the MC should start with a non-halting state, so the discovered programs would not halt upon printing \mathbf{x} .

When computing agent's policy via V_k^+ (see Eq.(3)), one again faces a problem of exponential growth (here, the growth of the number of possible policies

with k), which seems to render planning impossible for any but very small k . To deal with this problem we propose to use an importance sampling procedure. In particular, the agent should be learning its own actions the way it learns the environment through universal induction. To compute its policy the agent now samples possible policies that are most likely. It then performs the reward expectation maximization on these sampled policies. This procedure is explained below in detail. Notice then, that in the limit of infinite resources the exact expectimax result (as prescribed by Eq.(3)) is recovered.

As we have said, the agent's input and output signals x_i and y_i are now learned on equal footing. The Solomonoff prior is thus defined on the strings \mathbf{xy} , (and not just on strings of \mathbf{x} , conditioned on \mathbf{y} in chronological manner, see [2]). We shift the agent's cycle index so that it is non-negative integer on future symbols, and negative on its history (the history = observed symbols, is denoted $\mathbf{xy}_{<0}$). We first use the MC procedure to sample the strings from $M_\tau(\mathbf{xy}|\mathbf{xy}_{<0})$ as described above. The i th sampled string is denoted $\mathbf{xy}^{(i)}$, the total reward along the string (starting from the current time = 0) is denoted $V^{(i)}$ and the number of times the string was sampled (multiplicity of the string) is denoted $m^{(i)}$. In the AIXI theory the next action (output signal y_0 in our case) is chosen by computing $V_{0k}^+(xy_{0:j})$ in the space of all possible strings (see Eq.(3)). We want to compute V_{0k}^+ over a restricted space of sampled strings only. We want to use frequency of sampling of a particular string as an approximation to the probability measure (as in Eq.(3)). A minor complication relative to Eq.(3) is that sampling now depends on probabilities of actions y_i . An optimal action at (cycle) time n is defined as $y_n^+ \equiv \text{argmax}_{y_n} \tilde{V}_{0k}^+(xy_{0:n})$, where \tilde{V}_{0k}^+ is the analog of V_{0k}^+ defined with approximate probability measure (see Eq.(7)). The idea is to drop the strings in which there are suboptimal choices of y_i . The procedure can be formulated recursively as

$$m_{n-1}^{(i)} = \begin{cases} m_n^{(i)}, & \text{if } y_n^{(i)} = y_n^+ \\ 0, & \text{if } y_n^{(i)} \neq y_n^+ \end{cases}$$

$$\tilde{V}_{0k}^+(xy_{0:n-1}) = \frac{\sum_{\substack{i: xy_{0:n-1}^{(i)} = xy_{0:n-1}}} \tilde{V}_{0k}^+(xy_{0:n}^{(i)}) m_{n-1}^{(i)}}{\sum_{\substack{i: xy_{0:n-1}^{(i)} = xy_{0:n-1}}} m_{n-1}^{(i)}} \quad (7)$$

starting from the horizon $n = k$ and setting there $m_k^{(i)} = m^{(i)}$ and $\tilde{V}_{0k}^+(xy_{0:k}^{(i)}) = V^{(i)}$. The summation in Eq.(7) is over the sampled strings which start with $xy_{0:n-1}$. It is not difficult to check (using Eq.(2)) that in the limit of infinite computational resources (that is $m^{(i)} \rightarrow +\infty$ for all possible trajectories) the exact expectation maximization procedure Eq.(3) is recovered.

4. Assumptions about the environment

The approximation to the AIXI that we consider is (most likely) not computationally universal. But we will give a couple reasons which, in our mind, justify such

an approach. First, the known universal schemes [14,15,16] are only asymptotically optimal, that is for a generic problem there likely to be a prohibitively large initial computation, needed to find an optimal algorithm. It is far from obvious how this entry cost can be reduced (for a wide class of interesting problems) without jeopardizing universality. Second, one could wonder about the implication of the anthropic principle for utility of the algorithmic prior, even if the universes are sampled from it [17]. A simple description does not by itself make a universe life-friendly (for example, a program computing digits of π has no room for life). For evolution, and hence for an intelligent life (as we now it), to succeed, a certain degree of space-time repetitiveness in the environment is desired. A bias toward such environments may potentially be very helpful, especially for achieving the goals of artificial general intelligence (AGI).

Motivated by the above ideas of life-friendliness as well as by general observation of the class of problems facing the AGI community, we make certain assumptions about the environment, as explained below. We consider computable environments - environments that can be described by a code, and we have in mind the shortest such code (given a particular environment). Think about the code as organized as a collection of subroutines, calling other subroutines and so on. If removing a particular subroutine call causes only a localized change in the output of a computation, such subroutine is said to represent a feature or detail. The time and extent of the output change define the time and (time) scale of the feature. A hierarchy of subroutine-calls sets the hierarchy of features (details).

A typical environment that we are interested in possesses features on all time scales, and a typical larger scale detail is built on top of smaller (yet comparable) scale details. Description of a typical detail is simple (here, simple = low Kolmogorov complexity), given description of the smaller scale details it is built on. We assume that a typical environment structure is hierarchical and somewhat self-similar, that is the scale of features changes exponentially with hierarchy levels. The details, separated by the time much larger than their scale, can typically be viewed as i.i.d. (independent identically distributed).

The assumptions that we made are not very specific because we do not rigorously investigate computational complexity of our approach. The intention is simply to give a feeling for what kind of problems this approach may be suitable.

5. Redefinition of the Turing machine

In this section we discuss how the idea of using a Metropolis Hastings MC to sample programs from the universal prior should be further expanded and transformed in view of the assumptions about the agent's environment that were made in section 4. Because of the paper length restrictions, we will not go into details here, but rather outline general principles that one should follow. The details will be published elsewhere.

We expect the real world agent to process information and choose its actions in online manner. This means, on one hand, that the agent should be able to reuse fragments of its code, which it has constructed (to explain the data) over its history. On the other hand, most of the processing should be dedicated to

the recent data in a small, relative to the agent's lifespan, time window. In other words, the code explaining observations should be constructed incrementally, yet a typical new piece of code will be strongly connected to the already existing code. For the MC procedure formulated for a conventional Turing machine, which executes its program in a linear fashion, this should be a serious challenge, because of the problem of matching beginning of the new piece to the ending of the old code. (Remember that the MC procedure reconstructs a TM computation in reverse, hence the issue of matching in incremental learning).

Therefore, we need an effective representation of the code, which will allow efficient MC simulation. We list several properties that this representation must have.

- The new representation should conveniently enable reuse of its code. From the point of view of the original Turing machine, one modifies the MC procedure to include nontrivial moves, performing many original moves at once. Quite naturally, the new representation should contain the subroutines of section 4 as building blocks, and the new moves should operate on these blocks (subroutines).
- One consequence of our assumptions is that a typical cause-effect chain that we need to consider is short (it has small number of intermediate links, though time scope can be large), while the number of chains in the time window, where the data is actively processed, can be large. Hence, because the MC algorithm reconstructs a program in reverse (from effect to cause), execution of a program in the new representation should follow causal logical structure of the program, as opposed to a linear execution.
- The new representation should incorporate the hierarchical organization of details, with their scale changes exponentially with hierarchy levels. Because the MC affects the details that have causal effect on current observations, the time window of MC processing changes with the hierarchy level also exponentially. The same can be said about the range of the horizon (how far ahead the agents looks to predict): the smaller scale details have proportionally shorter horizon, than larger scale details.
- Solomonoff induction prescribes to try all possible programs, but assign lower prior probability to longer programs. In practice, due to resource limitations, one can only keep track of most probable programs. This also implies that in online inference one cannot (practically) sample candidate programs independently (a la Levin search), but should instead modify incrementally programs with high posterior probabilities. Some of the modifications that increase the posterior should be taken care automatically. For example, many subroutines will have free parameters. Those parameters should be encoded efficiently, with some kind of Shannon-Fano [18] or (adaptive) Huffman [19] coding, generally following concepts of minimum description length (MDL) [20,21],(or, more broadly, Solomonoff induction). Creation of new subroutines is another type of modifications. The probability of creating a particular subroutine should be determined through its algorithmic probability, as a function of its description length. Part of its description will encode calls of certain subprograms which create the new subroutine. The length of encodings of these subprograms is again deter-

mined from the same MDL principle: the larger the number of useful subroutines they create, the shorter their description will be, the more often they will be used. Then there will be higher level subprograms responsible for creating lower level subprograms and so on. These hierarchy of subprograms for creating modifications of the code essentially represent hierarchy of meta-learning levels.

6. Conclusion

We have presented very basics of ideas underlying the computational approximation to the AIXI model that we propose. We simultaneously pursued two goals: 1) to construct an approximation that reduces to the AIXI model in the limit of infinite computational resources, 2) to design an approach that does not sacrifice, when computational resources are limited, important real world capabilities, such as the ability to plan over the range of divergent time scales, from minimal (single cycle) to very large (exponential number of cycles). The details of the approach implementation relevant to the real world environments will be published elsewhere. The main immediate challenge we would like to address is to verify whether our approach is indeed suitable for tackling nontrivial real world problems. One of the steps that we consider in this direction is to test the proposed Monte Carlo algorithm on image recognition.

References

- [1] M. Hutter. Lecture Notes in Artificial Intelligence (LNAA 2167), *Proc. 12th European Conf. on Machine Learning, ECML*, 2001, 226-238
- [2] M. Hutter. *Universal Artificial Intelligence: Sequential Decisions Based On Algorithmic Probability*. Springer, 2004.
- [3] R. J. Solomonoff. A formal theory of inductive inference: parts 1 and 2. *Information and Control*, 7:1-22, 224-254, 1964.
- [4] R. E. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.
- [5] J. Schmidhuber. Goedel machines: self-referential universal problem solvers making provably optimal self-improvements. In B. Goertzel and C. Pennachin, editors, *Artificial General Intelligence*, 119-226, 2006.
- [6] R. J. Solomonoff. Complexity-based induction systems: comparisons and convergence theorems. *IEEE Trans. on Information Theory*, IT-24:422-432, 1978.
- [7] M. Hutter. New error bounds for Solomonoff prediction. *J. Computer and System Science* 62(4):653-667, 2001.
- [8] M. Li and P. M. B. Vitanyi. *An introduction to Kolmogorov complexity and its applications*. Springer, 2nd edition, 1997.
- [9] A. M. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proc. London Math. Soc.*, 2(42):230-265, 1936.
- [10] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:10871092, 1953
- [11] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97109, 1970
- [12] J. Schmidhuber. Discovering neural nets with low Kolmogorov complexity and high generalization capability. *Neural Networks*, 10(5):857-873, (1997)

- [13] J. Schmidhuber. The Speed Prior: A New Simplicity Measure Yielding Near-Optimal Computable Predictions. In J. Kivinen and R. H. Sloan, editors, *Proceedings of the 15th Annual Conference on Computational Learning Theory (COLT 2002)*, Sydney, Australia, Lecture Notes in Artificial Intelligence, pages 216–228. Springer, 2002.
- [14] L. A. Levin. Universal sequential search problems. *Problems of Information Transmission*, 9(3):265–266, 1973.
- [15] J. Schmidhuber. Optimal ordered problem solver. *Machine Learning*, 54:211–254, 2004.
- [16] M. Hutter. The fastest and shortest algorithm for all well-defined problems. *International Journal of Foundations of Computer Science*, 13(3):431–443, 2002.
- [17] J. Schmidhuber. A Computer Scientist’s View of Life, the Universe, and Everything. In C. Freksa, ed., *Foundations of Computer Science: Potential - Theory - Cognition, Lecture Notes in Computer Science*, pp. 201–208, Springer, 1997.
- [18] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, 1948.
- [19] Huffman, D.A. A Method for the Construction of Minimum-Redundancy Codes. *Proceedings of the I.R.E.*, 1952, pp 1098–1102
- [20] J. J. Rissanen. *Stochastic complexity in Statistical inquiry*. World Scientific, 1989.
- [21] C. S. Wallace. *Statistical and Inductive Inference by Minimum Message Length*. Springer, 2005.

Essential Phenomena of General Intelligence

Marc PICKETT¹, Don MINER and Tim OATES

University of Maryland, Baltimore County

Abstract. We present a set of cognitive phenomena that should be exhibited by a generally intelligent system. To date, we know of few systems that address more than a handful of these phenomena, and none that are able to explain all of them. Thus, these phenomena should motivate a system's design, test its generality, and can be used to point out fundamental shortcomings. The phenomena encourage autonomous learning, development of representations, and domain independence, which we argue are critical for general intelligence.

Keywords. Cognitive Architectures, Concept Formation, Autonomous Learning

Introduction

In this paper we suggest a set of cognitive phenomena for which the design for a generally intelligent agent should be able to account. This set is non-exhaustive, but we know of no system that meets all the criteria, and such a system would be an advancement for AI. This set can be used to motivate and evaluate such a design, which we'll refer to as a cognitive architecture. The more fully a cognitive architecture addresses the items in this set, the better. On the other hand, if an architecture fails to address some of these phenomena even in principle, then these holes may need to be addressed if the architecture's goal is general intelligence.

Background: Core Goals of General Intelligence

Traditional approaches to AI focus on selecting an application and then constructing representations for that domain by hand. These approaches are problematic in that they require much labor intensive knowledge engineering. Furthermore, these systems tend to be brittle, often failing when they encounter unanticipated situations. Earlier works discussing desiderata of intelligent agents ([1], [2], [3]) fall into this category, focusing on planning and problem solving using a human-provided domain model, and containing autonomous learning and development of representations as an afterthought, if at all. An alternate approach to AI is to have an intelligent agent develop its representations autonomously. In this alternate approach the agent is viewed as a "robot baby" [4]. The

¹Corresponding Author: Marc Pickett I, University of Maryland, Baltimore County; E-mail: marc@coral-lab.org

robot baby is provided a minimal amount of knowledge (implicit or otherwise) about the world and is expected to learn and develop a conceptual structure from large amounts of raw sensor data over a long period of time. This approach is attractive because it requires little knowledge engineering and is robust because the agent learns to adapt to unanticipated situations. This approach also directly addresses the Symbol Grounding Problem [5] — the problem of creating meaning using only a set of meaningless symbols — by directly grounding all an agent’s knowledge in sensory data.

Since we provide a minimal amount of domain knowledge, domain independence and generality should be among the top criteria for a cognitive architecture. Therefore, an empirical demonstration of an architecture should contain several disparate (though data-rich) domains with a minimal amount of human-provided data “massaging”. A set of domains might contain robot sonar sensor data, a large corpus of text, a series of images, and a simulation of Conway’s Game of Life. For each of these, an architecture should, at a minimum, autonomously develop an ontology that’s useful for characterizing that domain. For example, when given sonar data, the agent may build a hierarchy of motifs. When given images, the agent should develop edge filters, and when given Conway’s Game of Life, the agent should develop the concept of a “glider”.

To answer the question more precisely of exactly *what* an intelligent agent should do with its data is perhaps tantamount to answering the question of what intelligence is. It has been suggested that a core purpose of intelligence is to concisely characterize a set of data [6], [7]. That is, given data, an intelligent agent should generate a model that best compresses the data. This is the principle of Minimum Description Length (MDL). It is fundamentally equivalent to Ockham’s Razor, which says, in effect, that “The shortest model (that predicts the data) is the best model.”. If we assume that the prior probability of a model is inversely proportional to the exponent of its description length, then Ockham’s Razor is also fundamentally equivalent to the Bayesian principle that states that “The most probable model is the best model.”.

We somewhat agree with these claims. An intelligent agent should be able to build a model that concisely characterizes its sensor data, and it should be able to use this model to answer queries about the data. Such queries might consist of making accurate predictions about given situations. The agent should also be able to generate plans to accomplish goals (or obtain reward). However, the time needed (in terms of steps of computation) to answer these queries should also be taken into account. Thus, it is sometimes useful to occasionally trade memory for time. For example, an intelligent being might cache a result that it has deduced if it expects to use the result again.

To make this concrete, suppose our agent’s domain is Euclidean Geometry. In this domain, a huge but finite set of theorems of can be “compressed” down to a model containing just 5 postulates and some rules for inference. Such a model would neither be very useful nor would it work the same way as a person. A professional (human) geometer would likely “cache” useful lemmas, thereby speeding up his or her future deductions. It seems true that the same should apply to a generally intelligent being. Another example involves sensor data. If we equip our agent with a video camera, it’s possible that the most concise representation of the data (if the pictures are fairly continuous) will be an encoding typical of many video compression algorithms. That is, the representation might fully describe the initial frame, then describe each subsequent frame as changes from its previous frame. A problem with this approach is that it would take longer to answer queries about the end of the day than the beginning (because the entire

day would have to be “unwrapped”). This also seems contrary to our intuitions about what an intelligent agent should be able to do.

Thus, we propose an alternative to Ockham’s Razor called Marctar’s Axe, which states “The quickest model (that predicts the data) is the best model.”. By quickest, we mean the model that takes the fewest steps of computation to get accurate answers to queries. Of course, there’s a tradeoff between speed and accuracy, but this can be folded into a single number by setting a parameter that would act as an “exchange rate” between steps of computation and bits of accuracy. Marctar’s Axe somewhat overlaps with Ockham’s Razor in that fast models tend to be small and tidy so that computation isn’t spent searching through disorganized sets of information. Marctar’s Axe also addresses the utility of caching: caching the answers to frequent queries (or frequent “way points” in derivations) can yield a faster model.

In the next section we present a set of phenomena that a cognitive architecture should be able to explain or produce. We view the end goal of AI in terms of Marctar’s Axe. That is, to obtain quick and accurate answers or predictions about a set of data. The items in this set can be viewed as subgoals of Marctar’s Axe.

Desirable Cognitive Phenomena

We consider the following list of cognitive phenomena to be necessary (but not necessarily sufficient) features of general intelligence. A cognitive architecture that explains general intelligence should have a story for how it addresses them. This list is incomplete, but many cognitive phenomena not on the list are corollaries of those on the list. For example, a full solution to the problem of representing, creating, and using invariant representations could readily be used to solve the Frame Problem [8], which is the problem of stating what remains unchanged when an event occurs.

The items in the list aren’t necessarily independent. That is, some of the items might be corollaries of other items in the list. Therefore, these phenomena can either be directly addressed, or some may be solved as emergent properties of an architecture.

Concept Formation

As we mentioned in the Background section, an agent should be able to develop its own representations of the world. These representations should, at some level, form a concept ontology, which should be arranged in a semantic *hierarchy*. For example, the concept that corresponds to a pterodactyl should belong to both the class of flying things, and the class of reptiles. The concept formation mechanism should be able to make concepts out of virtually anything, not only physical objects. There should be concepts that characterize relations, events, stories, actions, and even cognitive actions.

Invariant Representations and Analogy

When a person dons a pair of green-tinted sunglasses for the first time, they have little trouble adapting to their altered visual input, but this isn’t such a trivial task for a (visual) robot. In terms of raw sensor data, a green-tinted scene has very different values from the same scene in its natural color. We suspect that this is because people have abstract representations that are invariant of the instances that caused them. Representations devel-

oped from visual data should also be invariant to translation, rotation, and scaling. These *invariant representations* aren't limited to visual data. A stenographer can hear different speakers say the same phrase in different pitches, volumes, and speeds, yet produce the same transcription.

An important class of invariant representations consists of those formed through *analogy*. Some suggest that analogy may even be the "core of cognition" [9]. Analogy allows us to focus on the relations among entities rather than superficial aspects of the entities. For example, we might notice that a red ant killing a black ant and stealing a piece of food it is analogous to a situation in Hamlet where Claudius murders Hamlet's father and usurps the throne of Denmark. In this situation *binding* is important. That is, we must be able to specify that the red ant corresponds to Claudius, the black ant to Hamlet's father, and the piece of food maps to the throne. Analogy is also useful for *knowledge transfer*: if an analogy is found, then conclusions about one domain can map to another domain.

Plato's Cave: Theory Building

In his Allegory of the Cave [10], Plato describes a group of people whose observations of the world are solely shadows that they see on the wall of a cave. The question may arise as to whether these observations are enough to propose a theory of 3-dimensional objects. In principle, this problem can be solved. If an agent is given a representational framework that's expressive enough to encode a theory of 3-dimensional objects, then the agent could go through the combinatorially huge number of theories expressed in this language (under a certain length) and choose the one that best explained the data (where "best" can be defined in terms of Ockham's Razor or Marctar's Axe). The best theory will likely include a description of 3-dimensional objects (assuming such a theory is of unrivaled utility for characterizing the data). Thus, our task is possible, given an exponential amount of time. There are other real examples of building theories of phenomena that aren't directly observable: neither atoms, genes, radio waves, black holes, nor multi-million year evolutionary processes are directly observable, yet scientists have built theories of these.

A robot given raw sensor data (such as uninformed visual data) is faced with fundamentally the same problem. Visual observations of a 3-dimensional object, such as a pen, can be very different (in terms of raw sensor data) depending on whether the pen is viewed lengthwise or head on. Therefore, the ability to propose "scientific theories" of this type is something a cognitive architecture should be able to explain. The architecture's representation framework needs to be expressive enough to encode such theories, and the architecture's model-builder should be able to discover theories in polynomial time.

Reasoning, Parsing, and Planning

Clearly, the ability to reason is an essential component of intelligence. Reasoning requires rules of inference, and the ability to *learn* these rules should be another requirement of intelligence. That is, an agent shouldn't rely on being told rules such as "If it is raining, and a robot goes outside, then that robot will get wet."

An agent should be capable of *hypothetical reasoning*: an agent should be able to represent counterfactual situations, and deduce consequences of these situations. An

agent might also pay special attention to cases where many different hypotheses yield the same conclusion (and thereby develop a general rule). For example, a robot might “imagine” several scenarios in which it falls from great heights, each simulation resulting in the conclusion that the robot would be damaged. The robot should then generalize that falling from heights causes damage. An essential component of hypothetical reasoning is being able to represent that an event is merely make-believe. Some otherwise promising systems, such as that described by Hawkins and Blakeslee [11] seem to lack this ability. Furthermore, reasoning should be able to continue for more than a few steps, even under uncertainty (in which case an architecture should be able to investigate different scenario branches).

An agent should be able to explain its world in terms of its learned conceptual structure. This should be done by parsing, or classifying data according to its current concepts, and by using rules of inference that it has developed. An architecture should be able to escape local optima in its characterization of the world. For example, an agent should be able to reclassify data, or replace one explanation by a shorter one. An agent should also be able to remove obsolete or unused concepts from its ontology.

An agent should also be able to use reasoning (and especially hypothetical reasoning) to develop plans for accomplishing goals. We suspect that reasoning, parsing, prediction, classification and explanation in terms of developed concepts, and planning can be implemented as different facets of a common algorithm. For example, planning might simply be the process of “explaining” how a desired situation can come about.

Finally, an agent should be able to combine its abilities to reason, learn rules of inference, and form heterarchical conceptual structures to implement hierarchical reasoning. That is, an agent should be able to create a conceptual structure of rules, and use these rules to quickly reach conclusions. For example, an agent faced with the task of breaking a piece of wood might follow this path: (going up the heterarchy) “Wood is-a rigid object. Breaking is-a change. To change a rigid object requires force applied to it.” (and back down) “There are several ways to apply force. Some involve striking with another rigid object. A rock is another rigid object...”.

Metacognition

It would be useful for an agent to be able to observe and modify aspects of its own cognitive behavior. Such metacognition might allow the agent to cache “lemmas” (and other conclusions) or develop heuristics to speed searches. Metacognition, because it deals with information about information, is useful for deciding which actions and cognitive actions an agent can take to improve its model. Therefore, metacognition can be used to pose questions and design experiments (action plans) to gain information. Metacognition can also be used to search for inconsistencies in the model, and modify the conceptual structure when contradictions are found.

It’s possible for metacognition to be elegant: If the representation schema for an architecture is general enough, then it should be able to encode cognitive actions. If the model-building mechanism of the architecture is powerful enough to blindly work on any system described in its representation framework, then it’s conceivable that an agent can characterize its own cognitive actions just as it would characterize any other stream of data (assuming we separate cognitive actions from metacognitive actions, thus avoiding a feedback loop).

Statistical and Symbolic Components

Sun [1] argues that a cognitive architecture that explains people should contain “an essential dichotomy” that can be roughly paraphrased as a split between traditionally statistical and symbolic methods. Sun’s arguments are from a psychological perspective and might not apply to general intelligence. That is, we leave open the possibility that there might be a unified system that captures both the symbolic and statistical elements of cognition. Whether an architecture explicitly has this dichotomy or not, it seems clear that an intelligent agent should have the strengths of both components.

The world is too complex to be modeled completely, and therefore a system must be able to characterize and handle uncertainty. Furthermore, an agent should be able to pick up on subtle statistical patterns such as correlations among events. Statistical methods are useful for this, but standard methods, such as support vector machines or connectionist models, aren’t without their downsides. For example, any connectionist system must address The Binding Problem [12], the problem of specifying which concepts are bound to which parameters, which is important for making use of analogies. Solving The Binding Problem is trivial in symbolic representations. A statement as simple as “bind <symbol1> <symbol2>” might suffice.

Along similar lines, an intelligent system should also be able to create new grammatical constructions. For example, given the concept of “red” and the concept of “ant”, an agent should be able to represent a “red ant”. There should be no hard limit to the depth of allowable constructions. An agent should also be able to represent a (new) concept as complex as “raining rocking horses on Pluto’s 3rd moon”.

Language

We’ve explicitly excluded language from our list of phenomena because we’re not convinced that language is a primary phenomenon of intelligence. We suspect that if a system is able to create grammatical constructs, develop representations, reason and plan, then it will also be able to understand and generate natural language. We suspect that there’s a general algorithm that handles both language and the other phenomena we list in this paper. For example, graph grammars have been applied to a wide variety of applications [13]. Since graph grammars are an extension of string grammars, an algorithm for learning graph grammars can be applied to learning natural language grammars as well.

Empirical evidence from neuroscience also seems to support this hypothesis. Although there are parts of the human cortex, such as Wernicke’s area and Broca’s area, that are associated specifically with language abilities, there is also evidence of the plasticity of the human brain. For example, patients who have their language dominant hemisphere removed are sometimes able to relearn to produce and comprehend natural language [14].

Outlook

We feel that at this stage, a relatively simple system that’s unoptimized, yet able to address most of the phenomena described above is preferable to a complex architecture

that's optimized but able to address just a few items. This is for the same reason that, were we studying aviation in the late 19th century, a Kitty Hawk style flyer would have been preferable to a system containing only the landing gear and navigation system of an F-15. That is, an architecture should be a full design for intelligence. Case studies in various aspects of intelligence may be useful, but an architecture should focus more on the overall design and interactions of these components.

Therefore, elegance of an architecture is important. An architecture should strive to be simultaneously both simple and general. One strategy to accomplish this is to put the bulk of complexity in the *process* of the architecture as opposed to the architecture's *design*. That is, a simple architecture that takes more computational resources to learn and reason should be preferable to a complex architecture that's a constant factor more efficient. For example, an architecture describable in 500 lines should be preferable to an architecture that takes 20,000 lines to describe but runs 10% faster. Optimization should come after generality.

References

- [1] R. Sun, "Desiderata for Cognitive Architectures," *Philosophical Psychology*, vol. 17, no. 3, pp. 341–373, 2004.
- [2] P. Rosenbloom, J. Laird, and A. Newell, *The SOAR papers: research on integrated intelligence*. Cambridge, MA: MIT Press, 1993.
- [3] A. Newell and H. A. Simon, *Human Problem Solving*. Prentice Hall, 1972.
- [4] P. Cohen, T. Oates, C. Beal, and N. Adams, "Contentful Mental States for Robot Baby," in *Proceedings of the 18th National Conference on Artificial Intelligence*, 2002.
- [5] S. Harnad, "The Symbol Grounding Problem," *Physica D*, vol. 42, pp. 335–346, 1990.
- [6] J. G. Wolff, "Information Compression by Multiple Alignment, Unification and Search as a Unifying Principle in Computing and Cognition," *Artif. Intell. Rev.*, vol. 19, no. 3, pp. 193–230, 2003.
- [7] M. Hutter, *Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability*. EATCS, Berlin: Springer, 2004.
- [8] J. McCarthy and P. J. Hayes, "Some Philosophical Problems from the Standpoint of Artificial Intelligence," *Machine Intelligence*, vol. 4, pp. 463–502, 1969.
- [9] D. R. Hofstadter, "Analogy as the Core of Cognition," *The Analogical Mind: Perspectives from Cognitive Science*, pp. 499–538, 2001.
- [10] Plato, *The Republic, Book VII*. 360 BC.
- [11] J. Hawkins and S. Blakeslee, *On Intelligence*. Times Books, 2004.
- [12] C. von der Malsburg, "The What and Why of Binding: The Modeler's Perspective," *Neuron*, vol. 24, pp. 95–104, 1999.
- [13] H. Ehrig, G. Engels, H. Kreowski, and G. Rozenberg, eds., *Handbook of Graph Grammars and Computing by Graph Transformation, Volume 2: Applications, Languages and Tools*. World Scientific Publishing Co., 1999.
- [14] D. Boatman, J. Freeman, E. Vining, M. Pulsifer, D. Miglioretti, R. Minahan, B. Carson, J. Brandt, and G. McKhann, "Language Recovery After Left Hemispherectomy in Children with Late-onset Seizures," *Annals of Neurology*, vol. 46 (4), pp. 579–86, 1999.

OSCAR: An Architecture for Generally Intelligent Agents

John L. POLLOCK

Department of Philosophy

University of Arizona

Abstract OSCAR is a fully implemented architecture for a cognitive agent, based largely on the author's work in philosophy concerning epistemology and practical cognition. The seminal idea is that a generally intelligent agent must be able to function in an environment in which it is ignorant of most matters of fact. The architecture incorporates a general-purpose defeasible reasoner, built on top of an efficient natural deduction reasoner for first-order logic. It is based upon a detailed theory about how the various aspects of epistemic and practical cognition should interact, and many of the details are driven by theoretical results concerning defeasible reasoning. The architecture is easily extensible by changing the set of inference schemes supplied to the reasoner. Existing inference schemes handle many kinds of epistemic cognition, including reasoning from perceptual input, causal reasoning and the frame problem, and reasoning defeasibly about probabilities. Work is underway to implement a system of defeasible decision-theoretic planning.

Keywords: OSCAR, defeasible, epistemic cognition, planning, practical reasoning

1. Epistemic Cognition and Defeasible Reasoning

OSCAR is a fully implemented architecture for cognitive agents, based on the author's extensive work in philosophy concerning epistemology [1,2,3,4] and the theory of practical reasoning [5]. OSCAR is written in LISP, and can be downloaded from the OSCAR website at <http://oscarhome.soc-sci.arizona.edu/ftp/OSCAR-web-page/oscar.html>.

The basic observation that motivates the OSCAR architecture is that agents of human-level intelligence operating in an environment of real-world complexity (henceforth, GIAs — “generally intelligent agents”) must be able to form beliefs and make decisions against a background of pervasive ignorance. It takes little reflection to realize that what human beings know about the world is *many* orders of magnitude smaller than what is true of the world. Our ignorance is of two sorts. First, we lack knowledge of almost all individual matters of fact. What do you know about individual grains of sand, or individual kittens, or apples hanging on all the apple trees scattered throughout the world? Suppose you want to adopt a kitten. Most planners make the closed world assumption, which would require us to know everything relevant about every kitten in the world. But such an assumption is simply preposterous. Our knowledge is worse than just gappy — it is *sparse*. We know a little about just a few of the very large number of kittens in the world, but we are still able to decide to adopt a particular kitten. Our knowledge of general matters of fact is equally sparse. Modern

science apprises us of some useful generalizations, but the most useful generalizations are high-level generalizations about how to repair cars, how to cook beef stroganoff, where the fish are apt to be biting in Piña Blanca Lake, etc., and surely, most such generalizations are known to no one.

In light of our pervasive ignorance, we cannot get around in the world just forming beliefs that follow deductively from what we already know together with new sensor input. We must allow ourselves to form beliefs that are made probable by our evidence, but that are not logically guaranteed to be true. For instance, in our normal environment, objects generally have the colors they appear to have, so we can rely upon this statistical generalization in forming beliefs about the colors of objects that we see. Similarly, objects tend to retain certain kinds of properties over time, so if we observe an object at one time, we tend to assume that, in most respects, it has not changed a short time later. None of these inferences can deductively guarantee their conclusions. At best, they make the conclusions probable given the premises.

GIAs come equipped (by evolution or design) with inference schemes that are reliable in the circumstances in which the agent operates. That is, if the agent reasons in that way, its conclusions will tend to be true, but are not guaranteed to be true. Once the cognizer has some basic reliable inference schemes, it can use those to survey its world and form inductive generalizations about the reliability of new inferences that are not simply built into its architecture. But it needs the built-in inference schemes to get started. It cannot learn anything about probabilities without them. Once the agent does discover that the probability of an *A* being a *B* is high, then if it has reason to believe that an object *c* is an *A*, it can reasonably infer that *c* is a *B*, and the probability of this conclusion being true is high. This is an instance of the *statistical syllogism* [9]. Notice that in order for the agent to reason this way with new probability information, the statistical syllogism must be one of its built-in inference schemes.

An agent whose reasoning is based on inference schemes that are less than totally reliable will encounter two kinds of problems. First, reasoning in different ways (employing different subsets of the built-in inference schemes) can lead to conflicting conclusions. In the literature on defeasible reasoning, this is what is known as “rebutting defeat” [6,7]. For example, I may look at an object and note that it looks red to me. This gives me a reason for concluding that it is red. But my colleague Claudio, whom I regard as highly reliable, may assert that it is not really red. Because I believe that he is highly reliable, the statistical syllogism gives me a reason for concluding that it is not red. An agent that reasons in ways like this needs some cognitive mechanism for deciding which conclusions to adopt when conflicts arise.

Second, as noted above, the agent can learn that it is in circumstances in which one of its built-in inference schemes is not reliable, or less reliable than it is assumed by default to be. If the agent discovers empirically that an inference scheme is unreliable in some specific circumstances, then when it learns it is in those circumstances, it should not make the inference, or if it has already made the inference, it should withdraw the inference. This is one kind of “undercutting defeat” [6,7]. Undercutting defeaters attack an inference without attacking the conclusion itself. For instance, if I know that illumination by red light can make an object look red when it is not, and I see an object that looks red but I know it is illuminated by red lights, I should refrain from concluding that it is red, but it might still be red.

As I have just illustrated, we can discover new probabilistic information that provides us with an undercutting defeater for an inference. Sometimes the system designer (or evolution) will already have noted this and built corresponding

undercutting defeaters into the system so that the cognizer can take advantage of them without having to first make the empirical discovery. In the human cognitive architecture, we find a rich array of built-in inference schemes and attendant undercutting defeaters. One of the tasks of the philosophical epistemologist has been to try to spell out the structure of these inference schemes and defeaters. I have made a number of concrete proposals about specific inference schemes [1,8,9,6,2,3,4]. Often, the hardest task is to get the undercutting defeaters right. For example, in [2] I argued that the frame problem is easily solved if we correctly characterize the undercutting defeaters that are associated with the defeasible inference schemes that we employ in reasoning about causation, and I implemented the solution in OSCAR.

Deductive reasoning is “monotonic”. Once an argument has been constructed for a conclusion, it becomes reasonable to accept the conclusion, and further reasoning is irrelevant. But defeasible reasoning is nonmonotonic. Argument construction is still monotonic, but constructing an argument for a conclusion no longer guarantees that the conclusion is acceptable, because we can have other arguments that conflict with the given argument. We end up with a network of interacting arguments, some of which may support defeaters for some of the steps of others. Jointly, these constitute an “inference graph”. The logical problem then arises of deciding what conclusions a cognizer should accept on the basis of its entire inference graph. Theories of defeasible reasoning aim at solving this logical problem. This is the “defeat status computation”, which computes which conclusions to accept and which to regard as defeated.

There are a number of different theories about what the defeat status computation should be computing. We need a characterization of the set of conclusions that should be regarded as defeated and the set that should be regarded as undefeated given any particular inference graph. Such a characterization constitutes a “semantics for defeasible reasoning”, and it provides the target for the defeat status computation. Originally (1980 - 1993), OSCAR was based on a semantics [1] that was later proven equivalent [10] to Dung’s [11] subsequently developed “admissible model semantics”. More recently (since 1993), in response to difficulties involving self-defeating arguments [12], OSCAR has been based on a semantics [6,13] that was recently proved [14] equivalent to the subsequently developed “preferred model semantics” of Bondarenko, et al [15].

The natural temptation is to try to build an implemented defeasible reasoner on the model of familiar deductive reasoners. First-order deductive reasoners generate the members of a recursively enumerable set of deductive consequences of the given premises. By Church’s theorem, the set of consequences is not decidable, but because it is r.e., its members can be systematically generated by an algorithm for constructing arguments. (This is what the completeness theorem for first-order logic establishes.) However, the situation in defeasible reasoning is more complex. If we assume that it is not decidable whether there is an argument supporting a particular conclusion (for first-order logic, this is Church’s theorem), then it cannot be decidable whether there are arguments supporting defeaters for a given argument. This means that in constructing defeasible arguments, we cannot wait to rule out the possibility of defeat before adding a new step to an argument. We must go ahead and construct arguments without worrying about defeat, and then as a second step, compute the defeat statuses in terms of the set of arguments that have been constructed. So argument construction must be separated from the defeat status computation. (Many implemented systems of defeasible reasoning do not make this separation, and as a result they are forced to focus exclusively on decidable underlying logics, like the propositional calculus. But

that is too weak for a GIA.)

A GIA cannot wait until all possibly relevant arguments have been constructed before we compute defeat statuses, because the process of argument construction is non-terminating. It must instead compute defeat statuses *provisionally*, on the basis of the arguments constructed so far, but be prepared to change its mind about defeat statuses if it finds new relevant arguments. In other words, the defeat status computation must itself be defeasible. This is precisely the way human reasoning works. We decide whether to accept conclusions on the basis of what arguments are currently at our disposal, but if we construct new arguments that are relevant to the conclusion, we may change our mind about whether to accept the conclusion.

The literature on nonmonotonic logic and most of the literature on defeasible reasoning has focused on what might be called *simple defeasibility*. This is defeasibility that arises from the fact that newly discovered information can lead to the withdrawal of previously justified conclusions. But as we have seen, there is a second source of defeasibility that arises simply from constructing new arguments without adding any new information to the system. We can put this by saying that the reasoning is *doubly defeasible*.

I have made four important observations about the reasoning of a GIA. First, it must have an expressive system of representations for encoding its beliefs, including at least first-order representations. Second, this means that it must employ powerful systems of reasoning, including at least full first-order logic. Third, this means that argument construction will be non-terminating, and the existence of arguments supporting particular conclusions will be undecidable. But fourth, that forces the reasoning to be doubly defeasible. To the best of my knowledge, OSCAR is the only implemented defeasible reasoner that accommodates double defeasibility [16,6]. Accordingly, OSCAR is the only implemented defeasible reasoner that can make use of strong reasoning techniques like first-order logic.

These observations have the consequence that a GIA cannot be viewed as a problem solver. The life of a GIA does not consist of being presented with (or constructing) a sequence of problems, solving each in turn and being done with it, and then going on to the next. Because reasoning is non-terminating, a GIA can only solve problems defeasibly. When the agent has to act, it acts on the basis of the solutions it has found to date, but if it has more time, there is always the possibility that better solutions will emerge or that difficulties will be found for previously discovered solutions. Problems can be set aside when defeasible solutions are found, but they can never be forgotten. There is always the possibility that an agent will have to return to a particular problem. Furthermore, solutions that were found for old problems but have not yet been acted upon may interact with what solutions are available for new problems, and that may necessitate looking for different solutions for the old problems. For a GIA (as for humans), nothing is ever completely finished. An implementation of a GIA is an infinite loop, not a finitely terminating problem solver.

2. Practical Cognition

Thus far I have focused on epistemic cognition. However, agents are most fundamentally entities that act on their environment, and the main purpose of cognition is to direct action. In crude agents, actions may be nothing but reflex responses to sensor input, but in sophisticated agents, actions are directed in part by appeal to the

agents' beliefs about their environment. The cognitive processes that issue in action make up what is called "practical cognition". As we will see below, the distinction between epistemic and practical cognition is not a clean one, but it is useful nevertheless. Roughly, practical cognition consists of those aspects of cognition that deal with adopting and executing plans.

We can distinguish between plan construction and plan adoption. In sophisticated agents, multiple plans may be constructed, aiming both at the same goal and at different goals. Plans aimed at different goals may interact in various ways (competing for resources, sharing actions, etc.), and this affects which plans the agent should adopt. So plan construction is one thing, plan adoption another. I will talk about plan adoption below, but first let us think a bit about plan construction. In a GIA, should that be part of epistemic cognition, or should it be a separate kind of cognition?

In recent years, there have been impressive advances in planning theory, resulting in "high performance planners" that are capable of solving problems much harder than those classical goal regression planners could solve. But most of this work is inapplicable to building GIAs. This is for two reasons. First, high performance planners almost invariably make the closed world assumption, and do so in an absolutely essential way that cannot be relaxed. But this flies in the face of our initial observation of pervasive ignorance. So GIAs cannot be constructed on the basis of such planners.

Second, a GIA will also lack knowledge of what the outcomes of actions will definitely be. It will at best be able to predict outcomes probabilistically. Accordingly, planning must be based on probabilities. Furthermore, GIAs do not face isolated planning problems. As an agent learns more about its environment, new opportunities arise and new threats arise, resulting in new planning problems. It may not be possible for an agent to achieve all its goals. So even if it can construct a plan for achieving a goal, that does not mean that the plan should be adopted. It must be considered how the plan is related to other plans. If the plan is adopted, will the agent have to reject another plan because, for example, it lacks the resources to execute both? In choosing between conflicting plans, the agent must take account of their costs and benefits. So the requisite kind of planning for a GIA is probabilistic and decision-theoretic.

Existing decision-theoretic planners almost invariably make two kinds of assumptions that are wholly unrealistic when applied to GIAs. First, they make the closed world assumption regarding individual matters of fact in the start state. Second, they assume that the cognitive agent has at its disposal a complete probability distribution regarding the probabilistic connections between any matters of fact that are relevant to the planning problem. For example, they often use this to encode information about actions in Bayesian nets. We have already seen that the closed world assumption is unacceptable for GIAs, but the assumption of a complete probability distribution is equally preposterous. Not only do we not know the values of most probabilities — a GIA could not encode their values even if they were known. Suppose a problem is described by logical compounds of a set of n simple propositions. Then to specify a complete probability distribution we must provide the values for 2^n logically independent probabilities. For a rather small number of simple propositions, there is a completely intractable number of logically independent probabilities. For example, given just 300 simple propositions, a grossly inadequate number for describing many real-life problems, there will be 2^{300} logically independent probabilities. 2^{300} is approximately equal to 10^{90} . To illustrate what an immense number this is, recent estimates of the number of elementary particles in the universe put it between 10^{80} and 10^{85} . Thus to know all the probabilities required for a complete probability distribution,

a GIA would have to encode 5 – 10 orders of magnitude more logically independent probabilities than the number of elementary particles in the universe. And this is from a problem that can be described in terms of just 300 simple propositions. In the real world we need vastly more.

There is a more profound difficulty for using existing planning technology in a GIA. Existing planners “compute plans”. Their input is a planning problem which includes all of the information needed to find a plan, and then they run a program that computes a plan and terminates. This strategy is inapplicable to GIAs, for several reasons. First, in the real world a GIA cannot be expected to come to a planning problem already knowing everything that is relevant to solving the problem. In the course of trying to construct a plan, an agent will typically encounter things it would like to know but does not know. For instance, if it is planning how to make a peanut butter and jelly sandwich, it may discover that it does not know where the peanut butter is. This illustrates that addressing a typical planning problem may give rise to epistemic queries which will initiate further epistemic cognition. Planning and epistemic cognition must be smoothly interleaved. This cannot be handled by doing all the epistemic cognition before beginning the planning, because we often do not know what we will have to know until the plan is partly constructed. For example, in a goal regression planner we may need to find out whether the preconditions for some action are satisfied, but we will not know which actions are relevant until the planning is already underway and a partial plan constructed. Most emphatically, we cannot require a GIA to already know everything that might be relevant.

Second, humans engage in hierarchical planning. That is, faced with a planning problem like that of how to attend a conference, they construct a schematic plan involving high-level actions like “fly to Los Angeles” and “rent a car”. Only later do they engage in further planning regarding how to fill in the details. This is for two separate reasons. First, planning is a computationally difficult and time-consuming process. An agent may have to make decisions (e.g., whether to accept an invitation to give a talk at the conference) that depend upon whether one can construct a good plan for attending the conference. Constructing a schematic plan may suffice for reasonably believing that one will be able to construct a full plan, because one may have general beliefs about the circumstances under which such schematic plans can be expanded. Thus one can quickly make the decision about whether to attend the conference, and then work out the details later. Second, and more important for present purposes, one often lacks the knowledge required to expand the schematic plan and has no way to get that knowledge until a later time. For instance, I might agree to pick up a friend at the airport without knowing the exact time at which his flight will arrive (it is “sometime Thursday afternoon”). This may be knowledge I cannot possibly acquire until shortly before the plane arrives (my friend may not even have purchased his ticket yet). So it is impossible to construct more than a schematic plan, but it is essential to do that much before agreeing to pick him up. Then when I learn more, I can expand the plan. Note further that, we almost never expand schematic plans fully until we begin executing them. In the process of executing the initial steps, we typically acquire further knowledge that is required for expanding the later parts of the plan. What this illustrates is that planning and knowledge acquisition must be interleaved. In building a GIA, we cannot assume that it knows everything it needs to know before it begins planning.

Many of these observations have been noted before by members of the planning community, but what has not generally been appreciated is that they profoundly change

the logic of planning. If planning and epistemic cognition are essentially interleaved, then because the epistemic cognition is defeasible, so must be the plan construction. If a plan is constructed by assuming beliefs that are later withdrawn, the plan should be withdrawn. In principle, this might be handled by replanning from scratch, but because planning is so computationally difficult, that would be very inefficient. It seems that it would be better to keep track of where in the planning process various beliefs are used, and then try to repair the plan if those beliefs are withdrawn. This is the way defeasible reasoning works. We have to keep track of our arguments so that we know what conclusions were used in inferring a given conclusion, and then we use that information in deciding which conclusions to accept in the face of defeaters. It looks like the structure of defeasible planning must be exactly similar, and will require all the same sophistications as full-fledged defeasible epistemic cognition. Rather than duplicating all of this in a separate planning module, and somehow integrating that module with epistemic cognition, it seems more efficient to do it all with a single module. In other words, do defeasible planning within the system of epistemic cognition by reasoning defeasibly about plans. It is at least plausible that this is the way humans construct plans.

Let us look more carefully at the idea that a GIA should construct plans by reasoning about them epistemically rather than by computing them using a separate module. How well this might work will depend upon the structure of the planning. It is an open question what form of planning should be employed by an GIA, but in the absence of the closed world assumption, the only *familiar* kind of planner that can be employed in a GIA is something like a classical goal regression planner. I don't want to claim that this is exactly the way planning should work in a GIA, but perhaps we can at least assume that the planner is a refinement planner that (1) constructs a basic plan, (2) looks for problems ("threats"), and (3) tries to fix the plan. It follows from our observations about epistemic cognition that the set of ⟨planning,solution⟩ pairs cannot be recursively enumerable, and planning cannot be performed by a terminating computational process [17]. This is because the search for threats will not be terminating. The result is that, just like other defeasible reasoning, planning cannot be done by "computing" the plans to be adopted. The computation could never terminate. A GIA must be prepared to adopt plans provisionally in the absence of knowledge of threats (that is, assume defeasibly that there are no threats), and then withdraw the plans and try to repair them when defeaters are found. So the logical structure of planning is indistinguishable from general defeasible reasoning.

The observation that planning is defeasible changes not only the logical structure of plan construction, but also the logical structure of the decision-theoretic reasoning involved in adopting plans. For a GIA to adopt a decision-theoretic plan, the requirement cannot be that it is optimal (i.e., at least as good as any other possible plan for achieving the same goals). This is because there will be no time at which it is not still possible that we will later find a better plan. Instead, a GIA must adopt "good" plans provisionally, being prepared to replace them by better plans if the latter are found [5]. But it may never find optimal plans.

My suggestion is that a GIA should reason epistemically about plans rather than computing them with some computational black box. To explore the feasibility of this approach, I implemented a classical goal regression planner within OSCAR that worked by reasoning about plans [18,19]. Various defeasible inference schemes were supplied, and then the planning proceeded by using them in reasoning. The planner was not particularly impressive (roughly equivalent to UCPOP [20]), but it worked,

indicating that this approach is in principle feasible.

Classical goal-regression planning is now regarded as hopelessly inefficient. In part, that is an unfair charge, because it derives from comparing goal-regression planning to the current high-performance planners, and as we have seen, they make essential use of the closed world assumption and as such are not viable candidates for use in GIAs. Nevertheless, classical goal regression planners do not seem to be able to solve very difficult problems. There may be reason to hope, however, that decision-theoretic goal-regression planners can do better. This is because knowledge of probabilities and utilities gives us important additional resources for use in directing search. Whether this will really work remains to be seen, but my current work is aimed at designing and implementing a defeasible goal-regression decision-theoretic planner that can be incorporated into OSCAR. The theory underlying this is presented in [5]. The proposed implementation is based heavily on recent results that I have obtained regarding reasoning defeasibly about probabilities [8] (thus obviating the need for a GIA to have a complete probability distribution at its disposal).

3. An Architecture for Interleaving Practical and Epistemic Cognition

I have argued that in a GIA, plan construction should be performed by epistemic cognition. What remains for practical cognition is the task of posing planning problems, evaluating plans once they are constructed, deciding which to adopt, and directing plan execution. All of these processes involve a lot of interaction between practical cognition and epistemic cognition. Basically, practical cognition passes queries to epistemic cognition and epistemic cognition sends answers back to practical cognition. In this section, I will give a high level description of how this all works in the OSCAR architecture. It is diagrammed in figure 1.

The point of epistemic cognition is to provide the information required for practical cognition. Any practical system of epistemic cognition must take account of what kind of information would be useful in the agent's practical endeavors, and focus its epistemic efforts accordingly. Practical cognition poses queries which are passed to epistemic cognition, and then epistemic cognition tries to answer them. Different queries are passed to epistemic cognition depending upon what practical cognition has already occurred. For example, once the agent has adopted a particular goal, it tries to construct a plan for achieving it. In order to construct such a plan, a query should be passed to epistemic cognition concerning what plans are apt to achieve the goal. Similarly, when the agent adopts a plan, the timing of the execution will depend upon when various things happen in the world, so practical cognition should send a corresponding query to epistemic cognition. Epistemic cognition answers these queries by producing appropriate beliefs, which are passed back to practical cognition. The queries posed by practical cognition comprise the set of *ultimate epistemic interests*.

Apparently the course of epistemic cognition must be driven by two different kinds of inputs. New information is input by perception, and queries are passed to it from practical cognition. The queries are *epistemic interests*. They represent things the agent wants to know. The agent can be equipped with inference-schemes that would allow it to answer a particular query if it knew something else. For example, given an interest in knowing ($P \ \& \ Q$), the agent could satisfy that interest if it knew P and Q separately. So the agent *reasons backward* and adopts interest in P and Q . These are

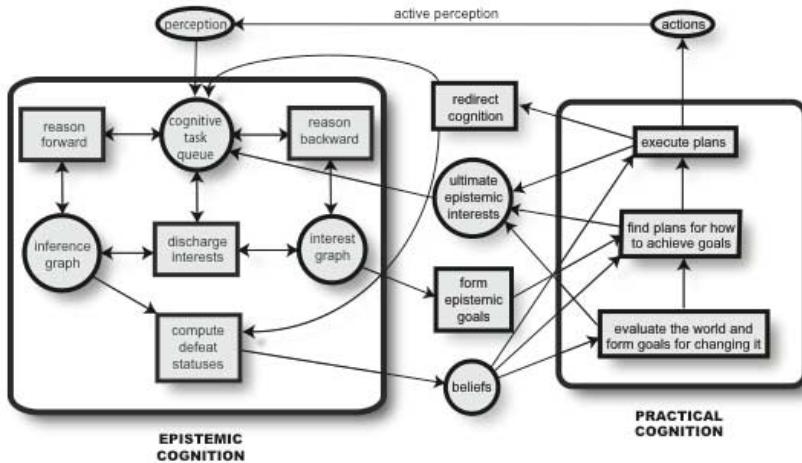


Figure 1. The OSCAR architecture.

derived epistemic interests. In this way the agent can reason backward from its ultimate epistemic interests and forwards from perceptual input until it finds itself in a situation in which forward reasoning has produced a belief that answers a query at which it has arrived by backward reasoning. This *discharges* the interest, enabling the agent to use the answer to that query in answering the query from which it was derived, and so on. By interleaving backward and forward reasoning, the agent is thus able to use its ultimate epistemic interests to help guide its epistemic cognition. OSCAR implements such bidirectional reasoning, and uses that to implement a natural deduction system for first-order logic [6]. It turns out that the bidirectional structure makes the deductive reasoner very efficient, and it often outperforms the more familiar resolution-refutation systems. A few years ago, Geoff Sutcliffe, one of the founders of the TPTP library (“Thousands of problems for theorem provers”) issued a challenge to OSCAR. At CADE-13, a competition was held for clausal-form theorem provers. Otter was one of the most successful contestants. In addition, Otter is able to handle problems stated in natural form (as opposed to clausal form), and Otter is readily available for different platforms. Sutcliffe selected 212 problems from the TPTP library, and suggested that OSCAR and Otter run these problems on the same hardware. This “Shootout at the ATP corral” took place, with the result that OSCAR was on the average 40 times faster than Otter. In addition, OSCAR was able to find proofs for 16 problems on which Otter failed, and Otter was able to find proofs for only 3 problems on which OSCAR failed. Taking into account that Otter was written in C and OSCAR in LISP, the speed difference of the algorithms themselves could be as much as an order of magnitude greater.

Many questions of practical interest cannot be answered just by thinking about what the cognizer already knows. To answer even so simple a question as “What time is it?”, the agent may have to examine the world — at least look at a clock. More difficult questions may require looking things up in reference books, talking to other

cognizers, searching one's closet, performing experiments in particle physics, etc. These are *empirical investigations*. What is characteristic of empirical investigations is that epistemic interests give rise to actions aimed at acquiring the information of interest. Actions are driven by practical cognition, so this involves a connection whereby epistemic cognition initiates further practical cognition. Practical cognition begins with the formation of goals, and then looks for plans that will achieve the goals. Accordingly, the mechanism whereby epistemic cognition can initiate practical cognition is by introducing "epistemic goals" — goals for the acquisition of information.

Empirical investigation introduces another interconnection between epistemic and practical cognition. There is a general distinction between "active" and "passive" perception, where active perception consists of putting oneself in an appropriate situation for passive perception. Some of the actions generated by practical cognition in the course of an empirical investigation will typically involve active perception. For instance, the agent may look something up in a reference book by turning to the appropriate page and then directing its eyes to the right place on the page.

Empirical investigation is accommodated by adding two links to the architecture. One link is from epistemic cognition, via the formation of epistemic goals, to practical cognition, and the other links is from practical cognition to epistemic cognition via active perception. Adding these links has the consequence that we get loops from practical cognition to epistemic cognition, then back to practical cognition, and so on. Practical interests give rise to epistemic interests, which may in turn produce epistemic goals, which may initiate a search for a plan for how to find the desired information, which passes a new query back to epistemic cognition. That may give rise to new epistemic goals and further loops back through practical cognition.

Although epistemic cognition is initiated by practical cognition, it need not be *directed* by practical cognition about how to answer questions. That would lead to an infinite regress, because practical cognition always requires beliefs about the world. If an agent could not acquire some such beliefs without first engaging in practical cognition, it could never get started. This indicates that there must be a *default control structure* governing epistemic cognition, and in particular, governing the way in which an agent tries to answer questions. However, in human beings it is also possible to override the default control structure. For example, consider taking an exam. A good strategy is often to do the easy problems first. This involves engaging in practical cognition about how to arrange our epistemic tasks. Our default control structure might, for example, take them in the order they are presented, but we can think about them and decide that it would be better to address them in a different order. Cognition about cognition is *reflexive cognition*.

When we redirect the course of our cognition by thinking about it and deciding what it would be best to do first, we are engaging in practical cognition about how to cognize. Clearly it is a good idea for a sophisticated cognitive agent to be able to modify the course of its own cognitive endeavors by engaging in practical cognition about how best to pursue them. An agent can learn that certain natural (default) strategies are unlikely to be effective in specific circumstances, and new strategies may be discovered that are more effective. The latter are often obtained by analogy from previous problem solving. It is desirable for a rational agent to use practical cognition in this manner to direct the course of either epistemic or practical cognition.

In the OSCAR architecture, cognition is driven by (1) sensor input, (2) the

production of new conclusions by reasoning, and (3) the production of new epistemic interests by either backward reasoning or queries passed from the various practical reasoning modules. There will typically be more cognitive tasks to be performed than can be performed at one time. So these three kinds of items are placed on a prioritized *cognitive task queue*, and popped off and processed in order of priority. OSCAR is then implemented as an infinite loop that repeatedly retrieves the top members of the cognitive task queue and passes it to the cognitive modules appropriate to its type, its type being any of (1) – (3).

4. Work to be Done

The OSCAR architecture is fully implemented, and has been since the early 1990's. It is described in some detail in my [6], and in more detail in *The OSCAR Manual* [21]. But this is not to say that I have a fully implemented GIA. The architecture is the skeleton of an agent. To build an agent, we must supplement it with sensors, inference schemes, and extensors. I have written fairly extensively about some of the inference schemes required for general epistemic cognition, and my ongoing work on decision-theoretic planning aims at filling in some more of the required inference schemes. Further inference schemes will be needed to direct reasoning about plan execution, and there are other gaps to be filled.

The OSCAR architecture is modular. The implemented system includes a natural deduction theorem prover for first-order logic, and a system of defeasible reasoning that implements the defeat status algorithm of [5]. However, the first-order theorem prover can be changed by simply altering the inference schemes supplied to the system, and different algorithms for computing defeat status can be used. OSCAR is equipped with inference schemes for reasoning defeasibly about perceptual input, temporal projection, causation, and the frame problem, and recent work has produced a system of inference schemes for reasoning defeasibly about probabilities. Existing AI systems can be incorporated into the architecture without change by treating them as Q&I modules. In many cases it may be possible to re-implement them as sets of inference schemes. Then, rather than operating as black boxes, they can be integrated into the more sophisticated control schemes of the architecture.

Work on extensors must be relegated to robotics. That is beyond the scope of my expertise. I hope that I will eventually be able to collaborate with roboticists on this.

Work on the sensors is also beyond the scope of my expertise, but the development of sensors must be integrated into the general inference schemes of epistemic cognition. This is something that I have written about elsewhere [3,4]. Humans do not just take the output of video-camera-like eyes and plug that into defeasible inference schemes. In humans, reasoning begins with the visual image, but the visual image is the product of very complex computational preprocessing that is only partly understood at this time. The human visual system takes two two-dimensional retinal bitmaps and converts them into a complex three-dimensional image already parsed into surfaces, objects, edges, corners, etc., located spatially with respect to one another. Furthermore, the image is the product of sensor output over an interval — not just instantaneous sensor output. We can see through a slotted fence while driving beside it (but not while we are stationary), we can literally see motion, and motion parallax plays an important role in computing the three dimensional aspects of the image. Because the image involves all this preprocessing, the epistemic cognition that appeals to it can be simpler. What we

are given is an image of objects and their properties — not just retinal bitmaps — and then we can reason from that sophisticated input, assuming defeasibly, for example, that objects tend to have the properties they are represented as having in the image. Building the reasoning system is probably much easier than building the visual system that is capable of doing this preprocessing.

5. Acknowledgment

This work was supported by NSF grant no. IIS-0412791.

References

- [1] Pollock, John, *Contemporary Theories of Knowledge*, Rowman and Littlefield, 1986.
- [2] Pollock, John, “Perceiving and reasoning about a changing world”, *Computational Intelligence*. **14** (1998), 498-562.
- [3] Pollock, John, and Joseph Cruz, *Contemporary Theories of Knowledge*, 2nd edition, Lanham, Maryland: Rowman and Littlefield, 1999.
- [4] Pollock, John and Iris Oved, “Vision, knowledge, and the mystery link”, with Iris Oved. In *Philosophical Perspectives* **19** (2005), 309-351.
- [5] Pollock, John, *Thinking about Acting: Logical Foundations for Rational Decision Making*, New York: Oxford University Press, 2006.
- [6] Pollock, John, *Cognitive Carpentry*, MIT Press, 1995.
- [7] Pollock, John, , “Defeasible reasoning”, in *Reasoning: Studies of Human Inference and its Foundations*, (ed) Jonathan Adler and Lance Rips, Cambridge: Cambridge University Press, 2008.
- [8] Pollock, John, “Reasoning defeasibly about probabilities”, in Michael O'Rourke and Joseph Cambell (eds.), *Knowledge and Skepticism*, Cambridge, MA: MIT Press, 2008.
- [9] Pollock, John, *Nomic Probability and the Foundations of Induction*, Oxford University Press, 1990.
- [10] Prakken, Henry and Gerard Vreeswijk, “Logics for Defeasible Argumentation”, to appear in *Handbook of Philosophical Logic*, 2nd Edition, vol. 5, ed. D. Gabbay and F. Guenther, Kluwer: Dordrecht, 2001.
- [11] Dung, P. M., “On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming, and *n*-person games”, *Artificial Intelligence* **77** (1995), 321-357.
- [12] Pollock, John, “Self-defeating arguments”. *Minds and Machines* **1** (1991), 367-392.
- [13] Pollock, John, “Justification and defeat”, *Artificial Intelligence* **67** (1994), 377-408.
- [14] Vo, Quoc Bao, Norman Y. Foo, and Joe Thurbon, “Semantics for a theory of defeasible reasoning”, *Annals of Mathematics and Artificial Intelligence*, **44**(2005), pp. 87-119.
- [15] Bondarenko, A., P. M. Dung, R. A. Kowalski, and F. Toni, “An abstract argumentation-theoretic approach to default reasoning.” *Artificial Intelligence* **93** (1997), 63-101.
- [16] Pollock, John, “How to reason defeasibly”, *Artificial Intelligence*, **57** (1992), 1-42.
- [17] Pollock, John, “The logical foundations of goal-regression planning in autonomous agents”, *Artificial Intelligence* **106** (1999), 267-335.
- [18] Pollock, John, “Reasoning defeasibly about plans”, OSCAR project technical report. This can be downloaded from <http://www.u.arizona.edu/~pollock/>.
- [19] Pollock, John, “Defeasible planning”, in the proceedings of the AAAI Workshop, *Integrating Planning, Scheduling and Execution in Dynamic and Uncertain Environments*, Carnegie Mellon University, June 7, 1998, ed. Ralph Bermann and Alexander Kott. AAAI Technical Report WS-98-02.
- [20] Penberthy, J. Scott, and Weld, Daniel, “UCPOP: a sound, complete, partial order planner for ADL”. *Proceedings 3rd International Conference on Principles of Knowledge Representation and Reasoning*, 1992, 103-114.
- [21] Pollock, John, *The OSCAR Manual*, available online from <http://www.u.arizona.edu/~pollock/>.

Anticipative coordinated cognitive processes for interactivist and Piagetian theories

Jean-Charles QUINTON ^{a;1}, Jean-Christophe BUISSON ^a and Filipo PEROTTO ^{a;b}

^a *University of Toulouse (INP/ENSEEIHT),*

IRIT (Computer Science Research Institute of Toulouse),

2 rue Charles Camichel, BP 7122, 31071 Toulouse Cedex 7 France

^b *Instituto de Informática,*

Universidade Federal do Rio Grande do Sul (UFRGS),

Av. Paulo Gama, 110 - Porto Alegre/RS - Brazil CEP: 90040-060

Abstract. This paper presents a model of intelligence based on principles introduced by Piaget and the interactivist framework. It focuses on embodiment and sensory-motor aspects of the mind but copes with general issues such as regulation, accommodation to noise and variability or synchronization of the agent internal dynamics with the environment. The proposed evolutionary and constructivist theory is illustrated by a real-time rhythm recognition program and goal-reaching algorithm, both based on prediction and assimilation.

Keywords. interactivism, Piaget, assimilation, anticipation, regulation, rhythm, goal-reaching

Introduction

Though most of the work has still to be done, we aim at understanding and modeling the fundamental principles that make cognition possible. Taking inspirations from biology, psychology or philosophy, we try to validate hypotheses by implementing them in computer programs. Even if they are no proof of the plausibility of such theories for natural intelligence, they still allow us to refine the theory by adapting incorrect assumptions. By developing a solid basis for a general purpose artificial intelligence, human performance capable machines might follow in the future.

In this view there is no need for a robot or an application to go much faster than any living being on the same specific task, performance being better accounted in terms of adaptation and adaptability. Indeed low level sensory-motor activities, most current systems are unable to perform, require synchronizing with the environment, respecting correct timing and physics laws. Since we are concerned with such kind of issues, we define and evaluate our progress by applicative achievements reflecting a global coherent framework. Integrating aspects from various fields of science and trying to match evo-

¹Corresponding author's e-mail: quinton@n7.fr

lution might not be necessary to reach a form of artificial general intelligence. Nevertheless analyzing or comparing the structure and functioning of organisms may help in producing robust adaptable autonomous systems.

Robots navigating in real environments face complexity, unpredictability, noise, electronic or mechanical imperfections, time delays and constraints. Thus, developing on robots is the perfect ground for testing adaptability; nevertheless simulations allow drastic parameters changes at will as well as environment customization. Moreover computer programs can be speeded up to monitor data evolution or down to analyze situations, produced in many versions at a fast pace and low cost. Therefore we have chosen to model simple behaviors, testing individual or coupled principles, in simulated environments with close to real-time real-life dynamics. Though artificial worlds and agents may possess only a few degrees of freedom, the complexity lies in the interactive and continuous nature of processes emerging from their coupled dynamics.

We believe in a philosophical naturalistic stance [1] as developed in the interactivist framework [2] or enaction paradigm [3]. Part of the forthcoming detailed applications are also based on Piagetian concepts such as assimilation and accommodation, introduced to account for the development of children during the sensory-motor period [4]. Several teams throughout the world capitalize on theoretical assumptions from the same background and target outstanding applications showing human level of cognition. However our goal is to go into the details of a coherent theory, explaining the development of an agent by coordinating action and perception through learning and coupling with the environment. We will therefore present the approach by detailing its main features then introduce explanations and applications targeting particular subjects of interest for artificial general intelligence: synchronization, use of tools, coordination, goal-reaching and unsupervised learning.

1. Framework and specificities

The main difference between the interactivist framework and classical symbolic and computational approaches, which have been dominating artificial intelligence and cognitive science for decades, is its commitment to a process metaphysics [5]. Many fields in science have progressively shifted from substance to process theories, for instance to account for quantum physics specificities. By moving from a list of properties and atoms to interactions and processes, the so-called ‘symbol grounding problem’ [6] is dissolved: novelty is explained in terms of emerging dynamics of coupled systems instead of combinations of a limited set of symbols. Representations and concepts are then functional and implicit; change becomes the default and stability is to be explained by compensatory influences or rhythmic cycles in a network of processes (or more generally in terms of attractors in the dynamical systems language).

Philosophers such as Bickhard or Dennett also support a continuously increasing complexity and progressive appearance of cognition, intelligence and consciousness through species evolution [7]. In the end there might only remain a weak coupling between the body and higher functions of cognition, but exploring the differentiations and emergence of living beings may be the first step to select relevant principles to design truly intelligent architectures. Additionally, this approach allows an incremental construction of agents with milestones set on the turns of past evolution. In a way it is sim-

ilar to the goals targeted by the subsumption architecture [8] though we rather look into refining the minimal set of principles at the lowest level before combining activities.

Although numerous processes might interact to produce an emergent behavior, the agent is neither an integrative nor hybrid system: we argue that unified principles can be applied for any modality and at any level. Of course the system might benefit from the full potential of classical artificial intelligence techniques or signal processing algorithms, but they will remain at the interface to ease the interaction with the environment. The homogeneity in structure and communication allows merging independently developed subsystems without having to implement the way they will combine: synchronization and coordination principles ruling the local dynamics extend to the global emergent system.

2. Principles

We will now describe the principles our research is focused on, in a sequence following the history of life evolution, even if many principles are deeply intertwined or might overlap in time.

2.1. Regulation

Regulation was already present in the very first living organisms, and sometimes even in non living phenomena. Animals are good at coping with variability as long as life is still possible, by regulating their behaviors and accommodating to environment variability. Computers are not: most programs show little sign of adaptation and hardly deal with ‘continuous’ changes in their inputs, even when introducing many other concepts we think useful for cognition as in Drescher’s model of Piaget’s schemes [9].

Regulation covers several meanings and goes from muscle fibers control to complex path planning. Even for what seem to be fairly simple contractions, regulation has to deal with strained cells, the effect of gravity depending on the movement, nutrient levels and a slew of additional factors. When shifting to complex bodily movements, regulation goes far beyond recruitment and homeostasis. Merleau-Ponty described how joints of the whole body get involved while trying to grasp an out-of-reach object [10]. Monitoring and controlling all these parameters in a centralized way is illusory, and more details will be given in the goal-reaching section of the paper.

More surprisingly, living organisms are also able to metaphorically map behaviors to different modalities, as long as the interactions keep their structure and temporality [11]. After birth, there is no clear differentiation between sensors (myelinization is for example not complete) and even after learning, the plasticity of the cortex can deal with dramatic changes in the brain-sensor coupling [12]. Regulating actions to cope with temporary hindrance or life lasting infirmities is of a rare complexity but is naturally performed by animals. Similarly, taping a known rhythm, humming it, singing it, nodding it or expressing it by any mean is something most humans can do.

2.2. Assimilation and far from equilibrium dynamics

In Bickhard’s genesis of cognition, far from equilibrium dynamical systems maintain their function by actively interacting with their environment. The very same behavior is found in the theory of autopoietic systems, illustrated by a simple cell model called

the tessellation automaton [13]. This form of survival and autonomy might be a good minimal candidate to define life, but may simply be interpreted in terms of assimilation. The concepts of assimilation and accommodation in Piaget's terminology refer to the forces that make processes subsist. An interactive process is defined by its function and relation to its environment. It needs to adapt to the situation, coping with the uniqueness of each moment in its history, either by neglecting differences or by modifying its own structure to match specificities. Such kind of processes will also be referred to as schemes in the following paragraphs.

Schemes might be passive, only synchronizing with external signals, but most of them affect their close environment by acting on it and increase their assimilation level in return, turning them into actors of the overall dynamics. Assimilation might well be enhanced or the associated process 'activated' by environmental cues independently of the agent's focus and decisions, but processes will influence or control actions depending on which level they assimilate the situation. For example, a known but unattended object appearing in the field of view might raise the assimilation level of a visual tracking scheme. The agent might progressively follow the new target when letting the scheme get control over eye movements; this is what happens when someone wanders and let attention focus on any salient feature encountered in its environment.

Any internal, sensorial or motor cue that can be assimilated by a particular scheme might increase its influence if conceived as an attractor in the global dynamical landscape of possible actions. For human beings, a simple evocation when thinking or dreaming is often enough to trigger a flow of activity spreading through the mind. The propagation interacts with the structure and current activity of the network, following thoughts associations, memories or sequences [14].

2.3. Interrelated parallel processes

As cells or organs interact in multicellular organisms at biological level, society members live their own lives but participate in several organizations. The same applies for interactive processes, which are constituents of the agent. Every scheme continuously tries to assimilate its environment, sometimes with success but often without being able to synchronize with inadequate situations. What matters at the global agent's level, is not the amount of adapted processes, but their existence with a sufficient level of activity to guide actions.

This massively parallel structure is compatible with redundancy, loss and creation of new schemes, each of them influencing the overall dynamics without being necessary to the emergent system. Several processes might perfectly synchronize with the same situation and even promote the same actions, not hindering each others. One might even be a more specialized version of another, allowing a progressive recognition. For example what might initially be assimilated to a fast moving object by roughly tracking a shape, may successively become a bird then an eagle. Associated schemes will be partially activated by the indirect influence of the general process, promoting compatible though more precise actions, like saccadic or smooth eye movements to check for particular features found on a bird but not on an airplane, found on an eagle but not on a crow.

Schemes not only indirectly interact through correlated actions on the environment but may also synchronize and coordinate with other related processes by propagating or modulating their activity. This aspect and its implications for goal-reaching behaviors

will be detailed in the corresponding section. Whatever the underlying principles ruling the interactions are, only the relative strength of the various attractors coexisting in the dynamical landscape matters in the end. This relativity removes the need for thresholds and even a broad weak activity might strongly influence the agent's perception if the context prevents any excessive stimulation. Internal activity might therefore overcome or replace the real sensory flow, making the agent subject to illusions or dreaming.

2.4. *Anticipation*

As soon as animals started to explore complex and unpredictable environments, genetics could no more select hardwired adapted reflexes for all situations and the risk of dying from an unadapted behavior increased. Therefore being able to learn from past experience and anticipate consequences of actions became crucial. Moreover by living in a future-oriented manner, organisms could integrate timing in their activities. This is particularly useful for 'distant' senses such as sight, to anticipate the approach of a predator, but also to account for inertia in complex metabolisms, nutrients being necessary for systems to function before a burst of activity. This shift from reactive systems to anticipative systems is an elegant solution to delay problems in any sensory-motor task, but also ease the understanding of planning and similar future-oriented phenomena.

Anticipation combined with assimilation introduces a normative aspect not present in frameworks requiring a supervisor stating whether the agent is right or wrong [2]. By acting on the environment and anticipating consequences, a scheme expresses a simple form of knowledge. The action may not be performed or anticipations not confirmed because of environmental constraints (obstacles hindering the movement or hiding features) or conflicting schemes (promoting different actions). In any case, the process will lower its assimilation level, i.e. its activity and strength as an attractor, and let other schemes decide for more adapted actions. In fact, the assimilation level symbolizes confidence in the behavior relatively to the current situation. Assimilation and the intrinsic inertia of schemes account for the object permanency appearing during child development. For example, the ability to follow objects even when partially hidden or moving behind a screen has been extensively studied in literature [15]; it might be explained by an insufficient decrease in assimilation during occlusions, keeping the agent focused on the anticipated trajectory.

When humans are faced with daily objects that are designed in such a way they become hardly recognizable, previously acquired schemes, perfectly adapted to standard pieces of furniture, still apply. As long as it can be comfortably sat on, a chair will be assimilated as a chair, whether it has a round shape or lacks a leg. The same was experienced by Tsien and his colleagues with mice lying in dishes or any material containers correctly satisfying the anticipations of a 'nesting' behavior [16].

3. Application to rhythm recognition

We have designed in the past a computer program which uses these principles for recognizing musical rhythms in an unsupervised way [17]. A human user had to rhythmically strike a single key on the keyboard. The user was free to speed up or slow down, stop and restart at any point of the score. Moreover humans are generally not able to keep a

perfectly regular pace and make mistakes even in a rhythm they decided to teach to the program, so it had to accommodate to speed, phase and imperfections in real-time (figure 1). Correctly recognizing and synchronizing with the rhythm involved anticipating the following beats, and the score was dynamically displayed and updated on the screen. We perform similarly when tapping with our foot while listening to some music. Notice that, given the slow biological feedback loop from ear to foot, a good synchrony is impossible to achieve without actually already knowing the rhythm and anticipating the beats. At the beginning the program would only know the binary rhythm, but progressively differentiate new rhythms, adding new notes or changing their length.

In addition to implementing and testing several principles, this program helped in realizing aspects that follows from assimilation: sensory signals must be assimilated by a process to be perceived and interpreted, novelty must be close to something already assimilated to be learned. Though an utterance in a totally unknown language would be assimilated as a sound and maybe as independent syllables, the rhythm and structure of sentences will not get caught without proper interactions with speakers. Assimilation may also explain how many magic tricks work: by focusing the attention of the public on distracting aspects of the situation, schemes tracking objects might never reach the sufficient level of assimilation and get enough control to make the agent detect a blatant disappearance [18].

Since a myriad of anticipated features combine into a scheme temporally strengthening its activity, there is often no need to confirm all the predictions. Except when the context is ambiguous enough and a decision has to be made to select a particular scheme, the agent will keep switching between behaviors or thoughts, as long as the emerging behavior remains adapted. Our inability to perceive stunning changes as well as unattended events in our environment when concentrating on a task is called perceptual blindness. It is integrated in several theories and accounts for several previously described properties of perception [19].

External influence of the environment on the agent's dynamics is often described as bottom-up or reactive depending on the field. It has to be related and compared to the top-down anticipative activity of internal processes. In the introduced rhythm program, external activity of the human user influences the learning algorithm and guides current recognition, successive sessions leading to totally different differentiations. At the same time, internal activity structures the perception, coping with noise, missing notes and

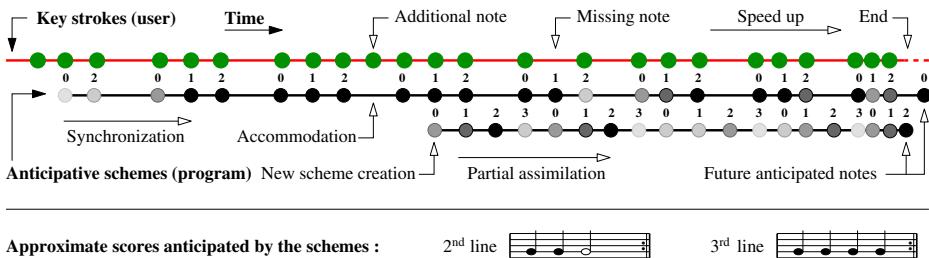


Figure 1. Annotated screenshot of the rhythm recognition program at the end of an interactive session. The first line corresponds to the unconstrained user input with each dot representing a single stroke. The additional lines reflect the knowledge of the agent, constituted by schemes anticipating future strokes in real-time. The program correctly accommodates to mistakes and pace variability. A genetic algorithm also generates differentiated schemes to accommodate to slightly different rhythms.

varying tempo. It also determines what the artificial agent is capable of and how much it can interpret. Although it is certain both interact and synchronize, what is called internal, external and the delimiting border is quite fuzzy. Interactions between top-down attention and bottom-up perception of salient features may account for the distinction between subliminal, unconscious and conscious processing [20].

4. Relation to the environment

This distinction between the inside and the outside, self and others, has recurrently been discussed by philosophers and is linked with embodiment in our view [21]. We must precise beforehand that we do not presume the body has to be natural (developing with mind during philogenesis) but think robots will be able to grow an implicit knowledge about their body by a short coupling history with their environment (ontogenesis). Still interacting with a coherent environment is necessary not only to learn regularities, but also to develop natural and intelligent behaviors.

The notion of environment is relative to the system, always constituted by the sensing and actuating capabilities it possesses. Environment is limited to a local extra-cellular space when considering cells exchanging molecules through their membrane, but embraces hormones for long-range internal communication between organs and distant object perception at the organism level. The organization and relations between dynamical systems eventually lead to the assimilation of a wide environment at the person level, including other individuals, technology or even society. When climbing several steps on the ladder of emergence, the coupling becomes weaker [22], as for the relation between higher values in human psyche and body metabolism.

Each process has a subjective view of the objective world and only interacts through perceptions and actions useful to its function. Variables and internal activities at a given level of emergence might well become inaccessible or part of the environment for a higher level system integrating the activity of hundreds of such lower level processes. Consciousness may result from the reflexive activity of a dynamic cluster of processes, modulating the activity of ‘lower’ unconscious systems. Tasks that require conscious attention at first, as when repeating moves specific to a newly practiced sport, progressively become automatic, to the point where it becomes impossible to decompose the skill into smaller pieces. Once mastered, the nearly autonomous skill can only be monitored and modulated by acting upon its environment.

This description makes even fuzzier the limits of the agent, though several researchers already consider the cognition is out there, ‘the world being its own best model’ as Brooks underlined in *Elephants don’t play chess*. Interacting with the environment is like discussing with it: acting is like asking a question, and perceiving like listening to the answer. Communication is intentional and goal-oriented: the agent is willing to satisfy its anticipations and come to an agreement but adapts its behaviors to the environment responses by introducing cooperative arguments.

Using tools is only a question of correctly integrating them into the agent dynamics by interfacing them with other behaviors. If coupled enough to be indistinguishable from pure abstract cognition, always playing a role in the resolution of issues and discovery of solutions, they might be part of the intelligence. A similar relation exists between speeches dealing with abstract concepts and associated spontaneous body movements,

contributing to the understanding of notions grounded in sensory-motor behaviors [23]. If tools had to be put somewhere, it would be between purely internal thoughts and clearly external laws ruling the physical world.

Perfectly using a tool has little to do with knowing how it works, as the archerfish shoots insects from distance without mastering fluid mechanics, gravity or refraction laws. The approach is rather similar with web services applied to grid computing: the end-user ideally just learns how to send requests, without caring about the algorithm used to solve his problem. The various libraries that may fulfill the request are articulated to the global system defined by its function. Back to organisms, a human body replaces its skin cells without losing its integrity or nature.

5. Internalized activities and goal-reaching

By propagating their activity to related processes, schemes whose assimilation level is not high enough to control the evolution of the situation still influence the internal dynamics. We will refer to this activity as ‘internalized’ although schemes might partially synchronize with the environment. In this case, anticipations are confirmed by a weak evocation from other processes. In a recently developed computer program designed to model goal-oriented coordination of schemes, we introduced activity propagation as the main characteristic to account for path planning. It proved functional and efficient in coordinating schemes in a 2D navigation simulation with dynamical goals.

Whereas assimilating schemes are reduced to their simplest expression in the current version of our demonstration program, the resulting behavior matches the basic expectations of a general navigation algorithm. A scheme consists of a contextual situation including sensory and motor information, an anticipated situation with the same structure and a real number reflecting its assimilation level. Such links between potentially perceived states act as shortcuts for the agent to project into the future. A chaotic network consisting of all the interactive schemes shapes the dynamical landscape. Though schemes remain the only form of representation, their heterogeneous distribution within the network accounts for the various needs in terms of precision and specialization. The relative weights reflecting schemes assimilation also allow continuous changes and regulation in the overall dynamics. In the field of color perception, red and yellow would

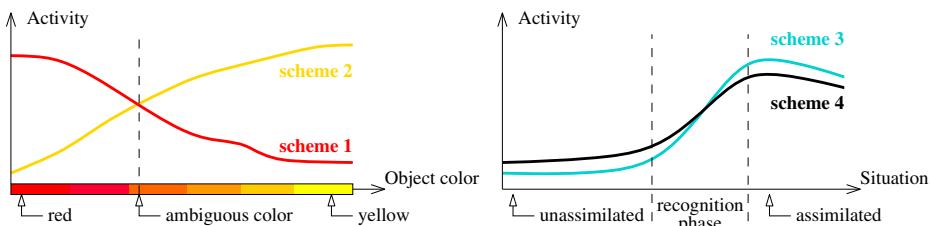


Figure 2. Conflicting or synchronous assimilation of schemes. (left line chart) Schemes assimilate different colors, potentially involving complex interactions. The number and repartition of such schemes might greatly depend on culture and environment. Whatever the basic set of colors is, intermediate colors will be differentiated by the relative level of assimilation of existing schemes. (right) Schemes 3 and 4 may simultaneously assimilate different correlated aspects of the external dynamics, as lip movements and associated sounds during an utterance. One might also be more specific than the other or just redundant.

for example be assimilated by two schemes, therefore creating distinct implicit classes, though orange might be perceived as an intermediate color, partially assimilated by both of them (figure 2). Color perception might not just be the acquisition of cones activity like reading a value, but a complex process involving changes in the reflected light depending on object and observer's positions [19].

The decision to go in a given direction to reach a goal is often taken without even being able to perceive the target. It is not only unrealistic to anticipate the consequences of all possible actions, but impossible to predict what the exact evolution of the dynamics will be. In our approach, goals consist in impeded schemes partially assimilating the situation but unable to fully satisfy their anticipations. If they reach the highest levels of activity in the network relatively to other processes, they will shape the overall dynamical landscape. Propagating their activity and increasing the assimilation level of compatible schemes, implicit chains of interactions will connect the current assimilated situation to the goal, guiding the agent toward it (figure 3).

Anyway, though the agent might imagine obstacles or intermediate steps, it will only be able to truly satisfy the associated schemes by acting and confirming its predictions, coping with unpredicted events or variations as they occur. From a car driving perspective to illustrate the phenomenon, some ‘abstract’ schemes would connect distant cities and diffuse their activity at their extremities. The activity is attenuated with distance to reflect the decreasing degree of assimilation. Simultaneously, a quantity of schemes requiring fast interactions with an ever changing perceptual world would account for traffic signs reading, braking or turning behaviors. Though all the schemes have the same structure and function, the difference lies in the scope of their influence, i.e. the range of the resulting shortcuts.

Although several schemes might perfectly assimilate the current situation, those co-ordinating with the goals will attract the agent slightly more. This bifurcation in the dynamics will progressively lead to a higher assimilation level of the chosen schemes, which will in return increase their control as long as their anticipations are confirmed. Additionally, goals are relative to a given scale and might be simply considered as local attractors when shifting to more global behaviors. Such a stance toward goals contrasts with reward maximization and similar algorithms supposing a clear cut between evaluation and cognitive processes [24,25] or the fixed hierarchy of Brook’s subsumption architecture.

6. Learning and evolution of the system

Learning may start with sensory-motor contingencies as immediate interactions with the environment. By a permanent tendency to assimilate its environment, the agent will generate new schemes to better account for specific situations or for complex regularities. The notion of object, concepts and symbols will then arise from schemes internalized activity [26], freed from the high variability and limits of the physical world. Indeed, by aggregating numerous local dynamics subject to rapid changes into statistically more stable schemes, the agent will recursively build an implicit hierarchy. Top elements will present a temporal and spatial stability characterizing abstraction, even in noisy environments disturbing low level synchronization, or complex situations where motor control switching between conflicting schemes is required [27]. When looking at a visual scene,

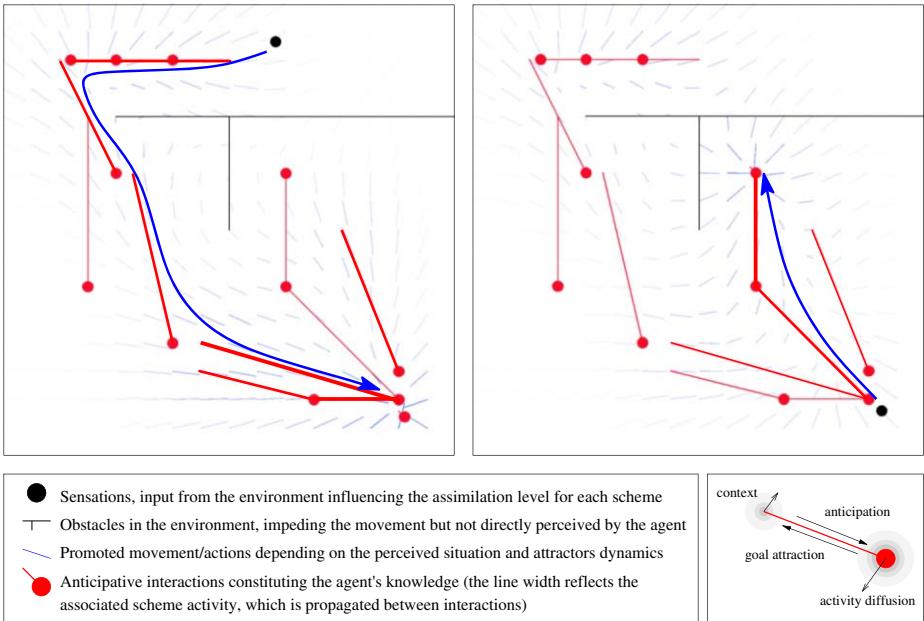


Figure 3. Goal-reaching dynamics illustrated on a screenshot of the 2D navigation program. Lines with dots correspond to known moves in the environment, the black dot to the current position. (left screenshot) Although no global trajectory is computed (*a posteriori* superimposed arrow), the agent follows the activity gradient resulting from the combined assimilations of all schemes. (right) Even when keeping the same structure, changes in activity lead to a different attractor and a totally different dynamics.

we perceive a structured world composed of permanent objects though the optical flow hitting our retina is totally renewed for each slight rotation.

Learning does not start nor end, neither can it be dissociated from the overall functioning of the agent. Learning improves the agent's knowledge in real-time by coping with previously unassimilated elements; it continuously modifies the dynamics but does not hinder it. Though memory might get reinforced by running the same schemes again and again, particularly during sleep [28], past memories also gets modified by present events. What appears singular and unique the first time it occurs gets progressively generalized when encountered again. Whereas specializing schemes too much results in an inability to take advantage of past experience in similar contexts, extreme generalization leads to confusion and unadapted behaviors. Thus an agent has to find the golden mean between assimilation and accommodation relatively to the tasks it has to perform.

Building multi-scale networks of interactions competing for assimilation not only helps solving this issue, but allows clusters of highly connected processes to emerge. These clusters, stabilized by activity propagation, might be the first step to escape sensory-motor limitations. Recognizing a 'restaurant' would be equivalent to staying in the network defining the concept while interacting with the environment. Visual interactions with the shop front, formalized communication with a waiter, arrival of the menu and selected meal dishes are all coherent with the standard 'being in a restaurant' behavior. Pubs, fast-food or self-service restaurants would be assimilated at a lower degree, but

the concept would remain adequate as long as the agent would satisfy its anticipations and goals.

Even if activity might be initially bound to physiological needs, monitoring body indicators like hormones levels, most of the internalized activity in a human adult is only weakly coupled with biology. Abstract attractors in everyday life dynamics, like habits or higher values, may derive from simple needs such as thirst or hunger. Such physiological processes are genetically selected, hence can be considered as ‘positive’ behaviors for the agent’s survival. Yet even obsessive-compulsive disorders, psychological addictions, self-mutilation or extreme sport practice could develop from the very same ‘positive’ behaviors. However, the need for direct perception and strong coupling with the body would be replaced by a recurrent autonomous internal activity relatively independent of metabolism regulations. For instance, assimilating the activity of others to its own schemes is a first step towards empathy. Nevertheless, processes regulating and stabilizing the basic functions of our organism would remain. Their overwhelming strength when unsatisfied is explained by the inescapable feeling of body signals: averting one’s gaze makes threatening pictures vanish but thirst is not to be forgotten (figure 4).

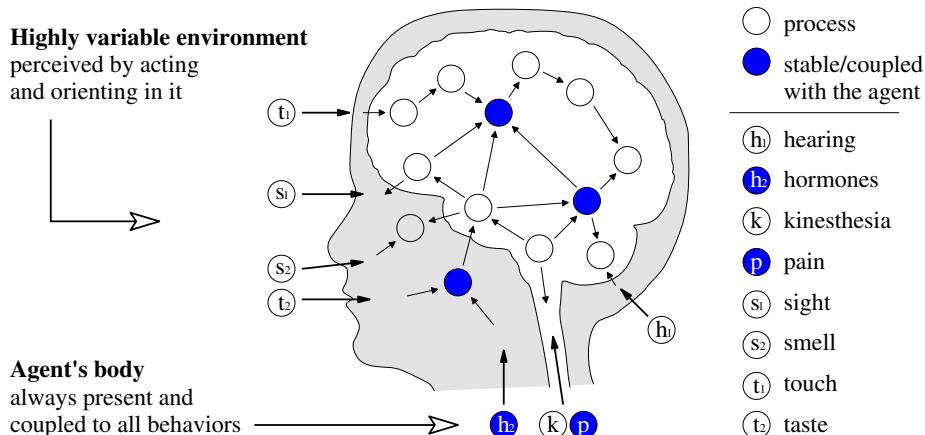


Figure 4. Evanescence and obsessive perceptions relative to the coupling with the agent. When the agent interacts with its environment, the flow from many sensory modalities might change drastically. Still internal sensations or the stable activity of some schemes will not be disturbed by such instant changes. Though a clear cut is drawn on the figure, a continuous range of couplings exists, defining a fuzzy limit between the agent’s core and environment.

Perspectives

After the rhythm and coordination applications, we now aim at better integrating all aspects of our theories in a computer program encompassing a broader spectrum of phenomena. Particularly, though time is implicitly introduced in the navigation algorithm by the constraints and timing of the environment, speed regulation is absent. A good candidate would be a driving agent who could visually anticipate trajectories and act accordingly. Gaze control introduces sensory-motor conflicts between promoted actions, thus a good coordination of schemes would be needed. Alternation and regulation of behaviors

are required to simultaneously track the road and look at traffic signs for example. This application opens up to another dimension, allowing many and even abstract extensions such as navigating in a city, respecting speed limits or avoiding accidents.

References

- [1] B. Forrest, Methodological Naturalism and Philosophical Naturalism: Clarifying the Connection, *Philosophy* **3** (2) (2000), 7–29.
- [2] M.H. Bickhard, *Foundational issues in artificial intelligence and cognitive science. Impasse and solution*, Elsevier, 1995.
- [3] F. Varela, E. Thompson and E. Rosch, *The Embodied Mind: Cognitive Science and Human Experience*, Cambridge, MA: MIT Press, 1991.
- [4] J. Piaget, *The origins of intelligence in children*, International Universities Press, Inc, 1952.
- [5] M.H. Bickhard and W.D. Christensen, Process Dynamics of Normative Function, *Monist* **85** (1) (2002), 3–28.
- [6] S. Harnad, The symbol grounding problem, *Physics D*, **42** (1990), 335–346.
- [7] D. Dennett, *Kinds of Minds: Toward an Understanding of Consciousness*, The Science Masters Series, New York: Basic Books, 1996.
- [8] R.A. Brooks, *Cambrian Intelligence - The early history of new AI*, MIT Press, 1999.
- [9] G.L. Drescher *Made-up Mind. A constructivist approach to artificial intelligence*, MIT Press, 1991.
- [10] M. Merleau-Ponty, *The Structure of Behavior*, Beacon Press, Boston, 1963.
- [11] P. Bach-y-Rita, M.E. Tyler and K.A. Kaczmarek, Seeing with the brain, *International journal of human-computer interaction*, **15** (2) (2003), 285–295.
- [12] J.Y. Lettin, H.R. Maturana, W.S. McCulloch and W.H. Pitts, What the frog's eye tells the frog's brain, *Proceedings of the IRE*, **47** (11), 1940–1951.
- [13] B.M. Mullin and F.J. Varela, Rediscovering computational autopoiesis. In: *Proceedings of the Fourth European Conference on Artificial Life*, Cambridge, MA: MIT Press, 1997, 38–U47.
- [14] B. Shanon, What is the function of consciousness?, *Journal of Consciousness Studies* **5** (3) (1998), 295–308.
- [15] S.P. Johnson, D. Amso and J.A. Slemmer, Development of object concepts in infancy: Evidence for early learning in an eye-tracking paradigm, *PNAS* **100** (2003), 10568–10573.
- [16] L. Lin, R. Osan and J.Z. Tsien, Organizing principles of real-time memory encoding neural clique assemblies and universal neural codes, *Trends in Neurosciences*, **29** (1), (2006), 49–57.
- [17] J.-C. Buisson, A rhythm recognition computer program to advocate interactivist perception, *Cognitive Science* **28** (2004), 75–88.
- [18] G. Kuhn and B.W. Tatler, Magic and fixations : Now you don't see it, now you do, *Perception* **34** (2005), 1153–1161.
- [19] J.K. O'Regan and A. Noë, A sensorimotor account of vision and visual consciousness, *Behavioral and Brain Sciences*, **24** (5) (2001), 939–1011.
- [20] S. Dehaene, J.-P. Changeux, L. Naccache, J. Sackur and C. Sergent, Conscious, preconscious, and subliminal processing: a testable taxonomy, *Trends in Cognitive Sciences*, **10** (5) (2006), 204–211.
- [21] T. Ziemke, What's that thing called Embodiment?, *Proceedings of the 25th Annual meeting of the Cognitive Science Society*, (2002), 1305–1310.
- [22] X. Barandiaran and A. Moreno, On What Makes Certain Dynamical Systems Cognitive: A Minimally Cognitive Organization Program, *Adaptive Behavior - Animals, Animats, Software Agents, Robots, Adaptive Systems*, **14** (2) (2006), 171–185.
- [23] E. Hutchins, *Cognition in the Wild*, Cambridge, MA: MIT Press, 1995.
- [24] N. Sprague and D. Ballard, Eye Movements for Reward Maximization, *Advances in Neural Information Processing Systems*, **16** (2003).
- [25] P.N. Prudkov and O.N. Rodina, Synthesis of Purposeful Processes, *Psychology*, **10** (70) (1999).
- [26] J. Piaget, *The construction of reality in the child*, New York: Basic Books, 1954.
- [27] J. Hawkins and S. Blakeslee, *On Intelligence: How a New Understanding of the Brain will Lead to the Creation of Truly Intelligent Machines*, Time Books, 2004.
- [28] M.A. Wilson and B.L. McNaughton, Reactivation of hippocampal ensemble memories during sleep, *Science*, **265** (1994), 676–679.

Hybrid Reasoning and the Future of Iconic Representations

Catherine RECANATI

*LIPN – CNRS UMR 7030, Université Paris 13
Institut Galilée, Av. J-B. Clément, 93430 Villetaneuse, France
catherine.recanati@lipn.univ-paris13.fr*

Abstract. We give a brief overview of the main characteristics of diagrammatic reasoning, analyze a case of human reasoning in a mastermind game, and explain why hybrid representation systems (HRS) are particularly attractive and promising for AGI and Computer Science in general.

Keywords. Diagrammatic representation, Iconic representation, Analogical representation, Hybrid representation systems, Cognitive modeling and reasoning.

Introduction

Logical linguistic representations have a high power of abstraction and many people think that they can model our reasoning abilities, partly because our knowledge expresses in linguistic terms, and also because our formal tools are built on alphanumerical representations. Nevertheless, inferential systems solely based on textual representations are very inefficient. Moreover, these systems raise difficulties at the representational level, because they require a complete specification of the concrete and abstract properties of the modeled objects. This is why computer scientists, used to think in terms of data structures, have early defended the use of diagrammatic representations, for instance in problem solving, on the basis of the fact that these representations were better adapted to specific domains (see [1] for an historical survey and critiques of logicist AI).

Although commonly used in Science, for instance in Mathematics or Physics, diagrammatic representations have long suffered from their reputation as mere tools in the search for solutions. At the beginning of the 90's, Barwise and Etchemendy (B&E) have strongly denounced this general prejudice against diagrams ([2], [3], [4]). To cope with complex situations, they defended a general theory of valid inferences that is independent of the mode of representation, and these works lead on the first demonstration that diagrammatic systems can be sound and complete [5].

As far as human reasoning is concerned, there are many examples using non linguistic form of representation, and, to quote B&E, "human languages are infinitely richer and more subtle than the formal languages for which we have anything like a complete account of inference. [...]. As the computer gives us ever richer tools for representing information, we must begin to study the logical aspects of reasoning that uses nonlinguistic forms of representation" [2].

Following in B&E footsteps, our general project is to defend the interest of hybrid representation systems (HRS) – i.e. systems linking together several kinds of representations. We claimed in [6] and [7], that only HRS could yield to the building of models of reasoning, both computationally efficient and cognitively plausible.

In this paper, we will first recall the most interesting characteristics of diagrammatic inferential systems, and add some comments about an example of human hybrid reasoning in a mastermind game. In the next section, we will give some arguments for the systematic study (and use) of HRS in AGI and cognition modeling, and some hints for their usefulness in program specification and semantics.

1. Some characteristics of diagrammatic inferential systems

In [2], B&E emphasized that the main properties of diagrammatic systems derive from the existence of a syntactical homomorphism between icons and represented objects. In many cases, this homomorphism yields to a very strong property called *closure under constraints*. In closed under constraints systems, the consequences of initials facts are included de facto in the representation and do not require extra computation. This makes these systems very efficient. As we have underlined in [6] and [7], this also shows a deep duality between two modes of reasoning.

Linguistic (or traditional logical) reasoning requires: (1) the representation of initial properties of objects; (2) an explicit representation of abstract properties (or relations among objects); and (3) a computational mechanism linking the two sources of information (to establish the validity of a non-explicit consequence). Thus, by construction, such systems require calculations. For instance, if you know that Ann is on the left of Gaston on a bench, and that Gaston is on the left of Isabel, you need to add that the relation “be on the left of” is transitive to prove that Ann is on the left of Isabel.

To the opposite, diagrammatic reasoning usually does not require the explicit representation of such abstract properties, because these properties are taken automatically into account by syntactic constraints on the representation itself. In our example, an iconic representation of the first fact will look like the (left) juxtaposition of two symbols (say, A for Ann and G for Gaston, as in: A G); and the second fact will yield to the juxtaposition of a third symbol (say, I for Isabel), as in: A G I.

Thus, you will just “see” on the resulting representation that A is on the left of I, without any computation. Since many consequences automatically appear on representations, diagrammatic systems provide an easy treatment of conjunctions and are computationally very efficient. Unfortunately, they have difficulties with disjunctive cases¹. Alternatives may require the use of several diagrams, which must then be traversed one after the other, as in the linguistic case¹. Note also that in many diagrammatic systems, each representation corresponds to a genuine situation, and that contradiction is impossible to represent (which can be good or bad depending on what you need to represent).

Many researchers have tried (in the nineties) to analyze diagrammatic inferential systems properties and closure under constraints in particular. For Stenning and Oberlander (S&O) [9], diagrammatic representations seem mainly to differ from

¹ The difficulty with disjunction reinforces the thesis that cognitive representations are mainly diagrammatical, because human performances are better in conjunctive than in disjunctive cases [8].

linguistic ones by a more limited power of abstraction, but greater computational efficiency. They claimed that there are three classes of representational systems: the MARS (Minimal Abstraction Representational Systems), the LARS (Limited Abstraction Representational Systems) and the UARS (Unlimited Abstraction Representational Systems). They argue that this hierarchy of representational systems is analogous to that of languages isolated by Chomsky, and that most diagrammatic representation systems are LARS. A MARS is a system in which a representation corresponds to a unique model of the world under the considered interpretation. For instance in a mastermind game, a row of letters standing for a row of colored pawns, as [B B Y Y R], will be a minimal abstraction representation of a possible solution. However, you can easily augment the number of models captured in a MARS by introducing new symbols that allow abstracting on representations. For instance, in the mastermind example, you can have a “–” symbol standing for an undetermined color, as in [B B – Y R]. Such systems can quantify massively on possible models, but cannot specify arbitrarily complex dependences between the specified dimensions. This is why S&O called them LARS. They claimed that only linguistic symbols, added to a representation, could allow the description of arbitrarily fine dependences between dimensions. They defined a LARS as “a system that keeps its representations simple, and keeps assertions out of its keys” and claimed that most diagrammatic inferential systems are LARS.

S&O identify the restricted capacity of diagrammatic systems with a property called “specificity”, which requires information of a certain kind to be explicit in all interpretable representation. In [10], Perry and Macken (P&M) have opposed to this strong notion of specificity (i.e. the mandatory specification of values of properties other than the one you try to represent) the notion of “determined character” due to Berkeley. Berkeley's notion of a determined character is that it is not possible to represent an object as having a certain property, without representing at the same time a specified value for this property. Thus, I cannot represent a triangle on a figure, without ending with a particular triangle. As well, it is not possible to represent a colored object on a drawing without specifying its color, but I can perfectly say, « this object has an interesting color », without specifying which oneⁱⁱ. For P&M, closed under constraints systems have, in addition to this determined character, a property called “localization” (already identified by Larkin and Simon in [12]). Localization is more important than specificity to characterize diagrammatic representations. Nevertheless, there are two properties of localization. The one identified by P&M is a purely logical property also called *unique token constraint*. It is the property of using only one token of a symbol to represent an object. This property disappears generally when you use a typed systemⁱⁱⁱ. Finally, P&M distinguish five kinds of representation going from text to images: graphic texts, charts, diagrams, maps and pictures. Their categorization uses two additional properties, iconicity and a constraint and systematic homomorphism (required to handle closure under constraints).

As far as geometric or spatial aspects are concerned, Macken, Perry, and Hass emphasized the importance of *iconicity* in [13]. Iconicity allows representations with richly grounded meaning – that is meaning whose relation to form is not arbitrary. An iconic sign may have a readily inferable meaning (RIM), an easily remembered meaning (ERM), or an internally modifiable meaning (IMM). Road signs provide numerous examples of ERM, RIM and IMM (for instance, signposting bends). There also are many examples of symbols having a RIM in musical scores (as for instance, crescendo situated under the stave). However, iconicity is only partially analyzed until

now, and IMM is still puzzling. We think that it could be sometimes linked to the syntactic homomorphism, because our personal conclusion is that the main distinction between linguistic (or symbolic) representation systems and analogical representation systems (as diagrammatic systems) must be characterized in terms of the power of the meta-language required to provide the semantics of the system. In the analogical case, the metalanguage needs to reference syntactical properties of the object language, while in the symbolic case, this is not obligatory^{iv}.

2. Hybrid human reasoning in mastermind

The preceding section recalls that iconic representations can be first class citizen, i.e. valid syntactical objects in inferential systems. It also underlines what iconic representations are good for and what they are not. At first sight, a limited power of abstraction and the request of a unique syntactical homomorphism are restrictive, and situations to which purely diagrammatic reasoning applies seem limited². Nevertheless, graphical and textual representation systems being complementary (at representational and algorithmic levels), the shortcomings of both systems can disappear in HRS. Therefore, the preceding review acts as a critique of current approaches to reasoning, which tend to emphasize only one mode, diagrammatic or linguistic, and are set up in opposition to the other mode (e.g. the mental logic vs. mental models debate). Let us now look at an example of hybrid human reasoning in a mastermind game.

Mastermind³ is well suited to the study of human reasoning, because it constrains the player to perform logical reasoning. Furthermore, the geometry of the grid encourages the players to use diagrammatic representations. For most players, reasoning is fragmented and opportunistic, and consists in partial deductions using several types of representations. In [14], we highlighted this hybrid character: most of deductions are graphical, while the model under construction partially expresses verbally. In fact, the use of graphical representations mitigates limitations in the cognitive capacities of the player, anchoring reasoning on inexpensive visual capacities, and relieving thus verbal memory. In return, visual capacities being themselves restricted, the shape of the diagrams and the ordering of hypotheses are biased (this because, even when they express verbally, hypotheses are also grounded on the grid). For instance, the left-to-right order (of pins and pawns) and the ease of visual translations, influence the choice of hypotheses to be considered first. Nevertheless, some players use these biases to develop their own strategy of resolution in an intelligent way.

We have insufficient room here to report all of our observations, but we can shortly comment a game of one player (grid on Figure 1). The grid ensures the memorizing of preceding results, but, as we will see, it is also a geometrical support for organizing proof and backtracking. Our player separates her game in two phases: first determining the colors, and then determining the places. In both phases, she uses

² Contrary to what may seem initially, graphical representations are not only helpful in modeling situations where a (concrete) spatial homomorphism applies.

³ The game consists in discovering a hidden row of five colored pawns. One player (the leader) hides a configuration of pawns. The second player can then dispose on a grid a tentative configuration of pawns, and the leader replies by posting pins (on the right) indicating if and how pawns correspond to the solution one's. A white pin means a good position and color for one pawn, and a black one a misplaced color. The rows remain visible during the game, and the player has to find out the solution with a limited number of rows.

representations that can be qualified as mental models because they are very similar to those of Johnson-Laird [15]. The interesting fact here is that these models (which also correspond to LARS of S&O) are ordered both by increasing order of specificity, and by decreasing order of probability. This makes backtracking easier, since the model considered next is determined, and guarantees a quick convergence to the solution, since these models are in decreasing order of probability.

6.	R R G Y G	o o o o o
5.	G R R Y G	o o o • •
4.	R G R Y G	o o o • •
3.	R R R G G	o o o •
2.	O O B B B	
1.	B B Y Y R	o •

Figure 1. Game of an experienced player

The player begins on row 1 by her favorite attempt (a 2/2/1 distribution), which possible replies revealed being statistically more informative than those of other colors distributions (such as 3/2, 4/1, 5/1, 1/1/3 or 1/1/1/2, etc.). Given the pins on the right side, she considers first the interpretation displayed on Figure 2, i.e. that one blue is placed correctly, one yellow misplaced, and that there is no red. (She might take in his hand a blue and a yellow pawn to help memorizing, and note mentally that the three colors are exhausted).



Figure 2. A first interpretation schema

We note this mental model by [1B] [1Y] (and “no red”) – using square brackets for the notion of exhaustion introduced by Johnson-Laird. (Note however that the model behind the schema of Figure 2 is more specific, since it includes some information on places, but in this first phase of the game, the player does not pay much attention to them). Then, she plays the second row, trying new places for blue (anticipation on future reasoning about blue places), and introducing a new color: orange. By luck, both orange and blue are missing colors, and the interpretation of the second row is obvious. Blue being excluded, she switches to a new model based on a new interpretation of the first row: [1 Y] 1R.

Then, she plays the third row both to try new places for red, and to try a new color. Getting four pins as a result, she concludes easily that the colors of the solution must be yellow, red and green. Given that there is only one yellow, she considers first [1Y] [2R] [2G] (which seems more probable than [1Y] [3R] [1G]). She then begins reasoning on places and supposes that on the first row, it is the first left yellow that is correct (we will note this model by [– – Y – –], knowing that the empty places must be filled by the missing pawns within [1Y, 2R, 2G]).

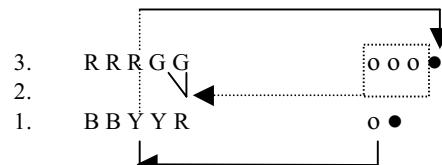


Figure 3. A diagrammatic reasoning

With the diagrammatic reasoning illustrated in Figure 3 (start following the arrows from the first row), she infers that on row 3, a red is misplaced, and thus, two greens well placed ($[- - Y G G]$). The solution should be $[R R Y G G]$, but this conflicts with the four pins of row 3, which should then be all white. Thus, she has to backtrack and reconsider the position of the yellow pawn on the first row ($[- - - Y -]$).

A graphic reasoning very similar to the preceding one reveals that in this case, the left green is misplaced and the right one correct (i.e. $[- - - Y G]$). She then tries a fourth plausible row, but is this time unlucky. Nevertheless, colors are confirmed and she knows by experience that, getting 3 white and 2 black pins means that two pawns have just to be exchanged to give the solution. The two pawns to switch are to be found in the first three pawns $[R G R - -]$, thus the green must be exchanged with one of the two reds. She tries $[G R R Y G]$ on row 5, but is unlucky again. However, there is now only one solution for the switch, and she wins on the last row.

An interesting fact about this game is the use of graphical inferences as those depicted on Figure 3. There are other sorts of graphical inferences used by experienced players. For instance, by focusing on the common parts of several rows, inferences can be drawn from the requested mappings between the set of common pins and the set of common pawns. All of these inferences are in a way “local” within the global reasoning, and they use creative graphic schemas mapped on the fly onto the grid. At a higher level, the strategy used by this player consists in a systematic ordering of the possibilities opened by a given row. This kind of strategy is often used. At the beginning, the reasoning is rooted on the first row, and the most probable model is considered first, here with a left-to-right bias in case of equality. For instance, in the preceding game, the ordering of the several models compatible with the first row (without considering places) is the following:

$[B][Y]$ no red < $[B]$ R no yellow < $[Y]$ R no blue.

The players compare competitive models and their relative probabilities directly from the number of pins and pawns. This is why some players have a tendency to prefer continuous color arrangements to separate ones, because quantities (or mass) are then more salient, and the comparison analogically performed easier. In many cases, the player builds a model in easy stages by covering a lattice where models fit into each other on a branch (by being more specific). The nature of considered models is not always as systematic as in our example, and may vary among players and/or situations. Nevertheless, an important fact is that these models layout on the grid in a visual manner. Figure 5 gives examples of several models (fitting together graphically).

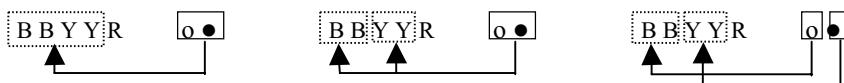


Figure 5. Some graphical schemas of interpretation

However, concerning our player, the global reasoning path is oriented by two directions (both grounded on the grid): (1) a left-to-right orientation of the possible models within a row, and (2) the natural vertical ordering of the rows. This systematic ordering helps remembering which model has to be consider next in case of backtrack. This global strategy applies as well in the second phase of the game. Here for instance, the ordering on the first row is:

$[- - Y - -] < [- - - Y -] < [- - - - R]$

The first model [– – Y – –] was quickly eliminated, and [– – – Y –] evolved progressively in a more specific solution.

Another interesting fact about this example is that diagrammatic representations prevent here from incoherence, instead of introducing errors (as many people claimed they merely do). Here this is due to the use of limited abstraction diagrams in which contradiction is impossible to represent. Furthermore, partially because of the specificity property mentioned in the first section, LARS appear to be good candidates for ordering models by inclusion. Models may also be orderly among other dimensions, by using probabilities or other specific attributes.

From this point of view, our example can be seen as a prototype for a family of programs, where information arises incrementally (here on each new row) and which are more or less determining (or approximating) the “solution” – thought of as a matrix of values. In such cases, the articulation of local (possibly graphical) subsystems within the lattice and the general level controlling flow (in charge of backtracking) is simple, because the role of each module is well definite. Each new information may bring specific constraints between specific values, and expresses partially in some subsystem, but the general program cannot be prepared to all of them. Then, other local heuristics or strategies will help and give a core to the general reasoning.

For instance, this sort of architecture could naturally apply to natural language processing, because text appears sequentially, both at discourse level and at sentence level. Suppose we have to process some text (i.e. already organized in words, but a similar architecture would apply to speech). Each word arrives with new information about the “meaning” of a sentence. Such meaning could locally be a matrix of several sorts of attributes (which might be values or actions) depending on what the software is supposed to do. With current semantics theories, it could be made of linguistic features from several domains (morphology, syntax, semantics, etc.). In each domain, there are specific constraints that can be handled by modular and more or less independent subsystems (for instance, in events semantics, you might have specific representations for time, space, causality, etc.). Thus, the general program may use statistics, proper strategies (as trying to discover syntactical features first) or a left-to-right bias as in our mastermind example, to find a way through the several possibilities of filling up the mandatory features – without presupposing that some of these features (syntactic ones in particular) have to be completely determined first. Particular features may also be let undetermined, to keep the natural lack of precision in language.

3. Perspectives for Hybrid Representation Systems in AGI

HRS may lead to amazing results concerning efficiency. A paradox is that a given demonstration may be limited by a minimal cost in any symbolic system, and still be less costly in a hybrid system including and binding the two sorts of representations (iconic or symbolic ones). Note that there is nothing sophistic here, because in a hybrid system there is no need of a global language to bind its subsystems⁴ (remind Gödel’s proof). Furthermore, the articulation of several subsystems in a complex representational system, bases sometimes simply on the fact that they denote the same objects in the world, and therefore coherence between two subsystems has not

⁴ This is why B&E gave a theoretical justification of the two main algorithms implemented in *Hyperproof* [16] based on purely mathematical grounds, without using any intermediate language.

necessarily to be handle. In the domain of reasoning, the objection that situations in which a unique homomorphism applies are rare is as well not too serious, because you can use several homomorphisms. The situation is just that the subsystems denote different properties of models or objects, and what expresses in one subsystem do not express necessarily in the other. Nevertheless, some information can be transfer from one system to another (on the basis of safe correspondences), endowing the global system with superior inferential and computational capacities. And there is no special need of an intermediate language.

Contrary to what may seem initially, graphical representations are not only helpful in modeling situations where a spatial homomorphism applies. Their increased use in science is also due to their obvious ability to convey abstract meanings. Via space, they bring new possibilities of structuring and abstracting (compared to sequences of letters alone). From this point of view, HRS are definitively on top of traditional UARS in the hierarchy of S&O. Besides their application to reasoning, their systematic study should improve formalization in many domains, which are relevant for AGI, as cognitive science, natural language semantics and linguistics. There are many domains in semantics where iconic representations seem better suited than logical formalisms. In linguistics, the numerous schemas, found in works on time and aspect (for instance [17], [18], [19]), are an indication of the plausibility of this thesis. We believe that concerning these domains, it is due to the nature of our cognitive apparatus (see next subsection). We also believe that the addition of iconic features in theoretical languages or tools could bring major advances in other fields of Computer Science, less concerned by world representations, as for instance, in the domain of semantics of programming languages, or in software design in general. By way of conclusion, we add two subsections to reinforce these claims. The following are surely controversial proposals. (They are also rather independent and some might be valid, others not.)

3.1. Outline of a model of the human mind (relation between thought and language)

To further understanding of the human mind, its higher cognitive capacities, and more specifically the nature of the relation between language and thought, the goal is to develop a model of language understanding and use that attains observational adequacy, i. e. that is able to pass the Turing test. To achieve this goal, we must aim higher, by trying to reach explanatory adequacy, that is, to develop a model of how the system can reasonably acquire the “knowledge” (i. e., systems of knowledge/belief, etc.) that enables it to attain observational adequacy.

The only way a mind can acquire the rich variety of knowledge humans do acquire is to start with a strong innate basis. The only way to build a system with a strong innate basis is to organize this basis into modules that are well adapted to representing the aspects of the world they represent. This is because of the way the world is (it is rich and varied, and the basic conceptual apparatus needed to represent time and temporal relations, for instance, must use different resources obeying different constraints than that needed to represent spatial relations, or interpersonal relations and other minds, or causal interactions, etc). There are probably also general computational constraints (problems of tractability and expressive adequacy), and the need for revision within relevant constraints (as well as many other factors), which will determine the emergence of a set of modules.

The mind’s rich set of innate modules, its “knowledge” about the world (including itself) is thus in the form of representational capacities. While it can be heuristically

useful to formulate knowledge/beliefs about time, for instance, as a set of axioms (i. e., declaratively) it is more plausible to consider that the mind embodies this knowledge as a capacity for representation (for instance, for representing temporal entities and relations among them). The knowledge is then embedded as constraints on what can be represented, and it will be useful to approach the problem of specifying knowledge in a certain domain, as the problem of specifying a ‘grammar’ of possible representations in that domain (e. g. possible representations of temporal relations among situations — precedence, overlapping, inclusion).

Besides this rich set of domain-specific modules, the mind needs to be equipped with a set of procedures for developing and enhancing the innate basis. While some of these are no doubt domain-specific, others must be domain-independent. We hypothesize that the human mind starts life with an innate basis for domain-specific knowledge that is more analogical or diagrammatic in nature, and that one of the important ways it develops is in the enrichment of the innate representational capacities with more symbolic representational capacities⁵.

A mind that has the ability to choose how it will represent a particular problem it needs to solve, choosing from a repertoire of representational capacities that include more analogical and more symbolic notations is more flexible, hence more “intelligent” (more apt to solve its problems, hence to survive). We postulate that humans have this kind of mind. To handle this ability to choose between several representational capacities, and to keep its repertoire relatively unchanged (after a certain level of development), a mind needs also to have generic and global cognitive procedures to construct representations on the fly.

Following the general framework of cognitive approaches to language, we believe that linguistic forms are (partial and undetermined) instructions for constructing interconnected domains with internal structure. As claimed in [20] by G. Fauconnier, this construction takes place at a cognitive level C. This level is distinct from language structure. Constructions at level C are not “meanings”, neither representations associated with any particular set of linguistic expressions. They are not representations of the world, or of models of world, or whatsoever of this sort. However, these constructions relate language to real world, and they provide various real-world inferences. They also are novel and different for each case of language use, and mental spaces and connections build up as discourse unfolds. The primary goal of (and primary evidence for) the approach in terms of interconnected domains is scientific generalization.

The first developments of Fauconnier “mental spaces” theory focus on processes of transfer from a source (or base) to a target. The capacity of organisms to carry out such projections lies at the heart of cognition in its many forms. The analyses given by Fauconnier are numerous and based on a rich array of linguistic data (counterfactuals; time, tense, and mood; opacity; metaphor; fictive motion; grammatical constructions; and quantification over cognitive domains). Further developments of the theory study another very interesting operation, conceptual blending [21], which also depends centrally on structure projection and dynamic simulation. Like standard analogical mapping, blending aligns two partial structures, but in addition, blending projects

⁵ Similar hypotheses relative to the architecture of mind and compatible with data (in psychology of reasoning), conclude to the existence of a meta-representational reflective level (i.e. handling meta-representation, as thought about thought, and the like) where slow logical inferences are drawn consciously.

selectively to form a third structure. (Creativity in Science is often based on conceptual blending).

All these works in Cognitive Semantics give us guidelines and examples to investigate in details how symbolic and iconic representations might relate in an intelligent complex system.

3.2. Additional remarks from a Computer Scientist point of view

The problem of the building (and, at first, of the description) of complex program architectures on computers is the concern of software engineering. We think that the numerous difficulties arising at this level are due to the deficiencies of our programming languages, in particular because they do not incorporate a more sophisticated level of description of the import features from other modules. Our claim is that they do not describe their own architecture (and therefore cannot incorporate import features at this level of description). We will not develop this claim here, although diagrams are obviously helpful for the description of architectures. We will only add a few remarks on iconicity (or on the non arbitrary shape of a symbol), to show that this dimension could be helpful at various levels of semantic description.

A first remark is that a non arbitrary shape character may appear in small touches, at the level of an isolated symbol, without even being included in a true iconic system (with proper analogical properties). For instance, a simple difference in the character font, as the addition of bold face, could modify traditional symbolic representations in a creative way. You can keep the old meaning for the new expression (for instance a value 0 and a value **0** both referring to zero as usual) and nevertheless have a supplementary meaning, relative to another dimension in the modeled world, or in the calculation process itself. You can for instance distinguish between a true (and final) value, from one that could still change, or be set by default by the system. Or, when added to more abstract symbols, as those describing the rewriting rules of a logic system, it could introduce second order rewriting rules, allowing to ignore intermediate terms (for instance, if not in boldface). Thus, traditional elimination rules could apply in a more efficient way (i.e. between distant elements), just by means of an additional graphical feature, defined and used at the level of the meta-language itself.

Our second remark is that, in the context of a computer, the general schema for the implementation of the homomorphism between syntactic representations and semantic representations do not stand on a simple line, as philosophers and logicians consider. It will translate into a program that will calculate, from internal representations (coming from our syntactic representations, by an operation of “internalization”), other internal representations – which we have to “externalize” if we want to get them explicit. Therefore, there are many other means to establish correspondences, or exploiting particular diagrammatic features, between all of these representations. In particular, some iconic relations may bind the syntax of the programming language used to that of the internal representations used, yielding to internally modifiable meaning (IMM). In reflective interpreters (cf. lisp), an object interprets as program or data, depending on its context of use. In such framework, the traditional data/program distinction vanished (as in machine languages). With reflective features, evaluation can be suspended or delayed (some functional languages implement lazy evaluation). Some programming languages may also have other specificities, as for instance, a pattern-matching operation as in PLASMA (an actor language of the eighties). Therefore, on a computer,

very complex relations between the represented world and the representing world are virtually possible.

Another remark relative to the use of bold (or other such features) is that it can obviously be used to handle some notion of focus. Focus theories have not yet been successfully designed, but it is a lack in our theoretical tools. There are many fields where some notion of focus would be of great help (in perception theory, in discourse theory, etc.). One reason of this failure might be precisely that the theories of focus require references to the underlying computational mechanism (as reflective properties of the programming language)^v.

If we take seriously the assumption of endnote IV, i.e., that the meta-language required to provide the semantics of a system has to reflect (in some way) the possibilities of configurations of terms in the representational language, then we have to investigate the following questions: what syntax do we need to easily provide the semantics of HRS? Would it be enough to add simple reflective and local graphical feature (as those of some of our programming languages) to a traditional functional and symbolic language, or should this syntax be trickier?

Conclusion

Works done so far on diagrammatic reasoning provide fragments of evidence about how people use iconic representations, and identify some of the problems raised by the project of AGI. Yet, there is still much to do to understand the variety of forms in which information can be stored and manipulated in intelligent control systems. We believe that we could make important progress in studying in details the relation between iconic and symbolic features in hybrid representation systems, as well as in paying attention to them in the theoretical tools and symbolic languages that we use.

Endnotes

ⁱ

However, contrary to what many authors have said, it is not difficult to represent disjunctive cases on diagrams, and we will see some exemplars in the next section (see Figure 5). It is also possible to have iconic symbols of second order in purely diagrammatic systems. C.S. Peirce first suggested to represent disjunctions in the form of a line connecting two iconic symbols. But in a formal system, the introduction of such symbols requires the definition of transformation rules on diagrams.

ⁱⁱ

The analogical/digital distinction also relies on a notion of specificity for Dretske [11]. For him, every signal transmitting information necessarily carries this information under two aspects: an analogical form and a digital form. The analogical form always contains an additional specificity relative to the information properly conveyed by the digital form.

ⁱⁱⁱ

The omnipresence of representation of the same type designating the same object is thus observed in human language, where references to an object can be spread out everywhere in a document, so that information is not « localized » (quoted from [13]). For P&M, this additional character is the one required to give diagrammatic systems the closure under constraints property, when combined with iconicity and a constraint and systematic homomorphism.

^{iv}

Let us take the example of Ann, Gaston, and Isabel, who are represented as « ordered » in the diagrammatic case. A minimal difference, but an essential one, between the two types of representations is the following:

- (I) left-of(a, g) & left-of(g, i)
- and (II) ordered([a, g, i]) (or just [a, g, i])

There is an additional syntactical complexity for (II) which prevents its meaning, contrary to that of (I), from being described as a function of one argument of its predicate's meaning. Indeed, you can easily assign a meaning to the semantic equation: $\llbracket \text{left-of} (a, g) \rrbracket = \llbracket \text{left-of} \rrbracket (\llbracket a \rrbracket, \llbracket g \rrbracket)$, while you cannot write anything else but: $\llbracket \text{ordered} ([a, g, i]) \rrbracket = \llbracket \text{ordered} \rrbracket (\llbracket [a, g, i] \rrbracket)$, which implies giving meaning, *at the meta-language level*, to a configuration of terms (the list figuring between simple square brackets). Therefore, the semantic descriptive meta-language must offer possibilities of syntactical structuring of data similar to the ones figuring in the representation language, because it will sometimes be necessary to assign them a meaning. This is not to say that all syntactical nuances of the representational system must be reflected in the interpretation system, because not all iconic representation features are interpreted in a diagrammatic representation (think to the use of marked features in mathematical figure to derive geometrical proofs). Nevertheless, it shows that semantic compositionality relies on syntactic considerations.

^v Note that in the context of graphical interfaces, several notion of focus are required at a very low level (in the graphic server itself), in order to link the keyboard (and/or events on the pointer of the mouse) to a particular window. The development of graphical interfaces (and networks) has introduced considerable changes in the previous programming framework. (1) There are other sources of input than letters (at least, mouse inputs), and other sorts of output (graphics, sound). (2) The input/output data are of distinct nature, but they may be linked together in the system (as the mouse and the screen). (3) The sharing of input/output devices by several programs adds some additional complexity to the emerging framework.

References

- [1] A. Sloman, Musings on the role of logical and non logical representations in intelligence, *Diagrammatic Reasoning : cognitive and computational perspective*. J. Glasgow, N. Nari, Narayanan, B. Chandrasekaran, (eds.), MIT Press, AAAI Cambridge, MA and London, 1995, 7-32.
- [2] J. Barwise and J. Etchemendy, Visual Information and Valid Reasoning, in *Visualization in Mathematics*, Zimmerman, W., ed., Mathematical Association of America, Washington DC, 1990.
- [3] J. Barwise and J. Etchemendy, *Hyperproof*. CSLI Publications, Stanford, 1994.
- [4] J. Barwise and J. Etchemendy, Heterogeneous Logic. J. Glasgow and alii (eds.), MIT Press, AAAI Cambridge, MA and London, 1995, 211-234.
- [5] S-J. Shin, *The logical status of Diagrams*. Cambridge University Press, 1994.
- [6] C. Recanati, Raisonneur avec des diagrammes: perspectives cognitives et computationnelles, *Intellectica* 40, 2005, 9-42. <http://hal.archives-ouvertes.fr/hal-00085004/fr>.
- [7] C. Recanati, Characteristics of diagrammatic reasoning, *Proceedings of EuroCogSci07*, The second European cognitive science conference, May 23-27 2007, Lawrence Erlbaum Associates, Delphi, Greece, 2007, 510-515. <http://hal.archives-ouvertes.fr/hal-00153328/fr>.
- [8] J. Bruner, J. Goodnow, and A. Austin, *A study of thinking*, Wiley, New York, 1956.
- [9] K. Stenning and I. Oberlander, A Cognitive Theory of Graphical and Linguistic Reasoning: Logic and Implementation, *Cognitive Science*, 19 (1), 1995.
- [10] J. Perry and E. Macken, Interfacing Situations, in *Logic, Language and Computation*, J. Seligmann and D. Westerståhl (eds.), Stanford University Press, 1996.
- [11] F. Dretske, *Knowledge and the flow of information*, Blackwell, Oxford, 1981, 137.
- [12] J. Larkin and H. Simon, Why a Diagram Is (Sometimes) Worth Ten Thousand Words. *Cognitive Science*, 11, (1987).
- [13] E. Macken, J. Perry, and C. Hass, Richly Grounding Symbols in ASL. CSLI Report no. 93-180, 1993.
- [14] C. Recanati, Diagrammes pour résoudre le problème d'Einstein et celui d'un joueur de Mastermind, Rapport LIPN, dec 2004, Université Paris13. <http://hal.archives-ouvertes.fr/hal-00085056/fr/>.
- [15] P.N. Johnson-Laird, *Mental Models: towards a cognitive science of language, inference, and consciousness*, Cambridge University Press, Cambridge, 1983.
- [16] J. Barwise and J. Etchemendy, *Hyperproof*. CSLI Publications, Stanford, 1994.
- [17] H. Reichenbach, *Elements of symbolic Logic*, Macmillan, New York, 1947.
- [18] C. Smith, *The parameter of Aspect*. Studies in Linguistics and Philosophy 43, Kluwer Academic Publishers, 1991.
- [19] N. Hornstein, *As Time Goes By - Tense and Universal Grammar*. MIT Press, Cambridge MA, 1993.
- [20] G. Fauconnier, *Mappings in thought and language*. Cambridge University Press, 1997.
- [21] G. Fauconnier, Conceptual blending and analogy. *The analogical mind*. D. Gentner, et alii (eds). MIT Press, Cambridge MA and London, 2001.

Cognitive Constructor: An Intelligent Tutoring System Based on a Biologically Inspired Cognitive Architecture (BICA)

Alexei V. SAMSONOVICH^{a,1}, Kenneth A. DE JONG^{a,b}, Anastasia KITSANTAS^c,
Erin E. PETERS^c, Nada DABBAGH^c, and M. Layne KALBFLEISCH^{a,c}

^a*Krasnow Institute for Advanced Study, George Mason University*

^b*Computer Science Department, George Mason University*

^c*College of Education and Human Development, George Mason University
Fairfax, VA 22030-4444, USA*

Abstract. Significant progress can be made in the part of elementary school education that relies on intelligent tutoring systems (ITS), if the role of a referee and a peer advisor will be performed by a pedagogical agent that is a computer implementation of a cognitive architecture modeling the process of learning. Recent studies in cognitive architectures funded by the DARPA IPTO BICA Program have identified the key potential of feasible today artificial intelligence as bootstrapped cognitive growth (i.e., gradual acquisition of knowledge and skills using previously acquired knowledge and skills), up to a human level of intelligence in a selected domain. This approach is not limited to laboratory settings and short-term paradigms, it is intended for a long-term, open-ended learning scenario in real-world settings. Several cognitive architectures were designed for this purpose, among which is GMU BICA, a self-aware biologically inspired cognitive architecture. Here we describe a computational model of student learning based on GMU BICA and its use as an ITS called Cognitive Constructor, which has two components called a Science Microworld and a Pedagogical Agent (GMU BICA agent). Results of our analysis show that the system will be useful in elementary school education.

Keywords. Self-regulated learning, intelligent tutoring system, pedagogical agents

Introduction

Integration of state-of-the-art computer technologies with leading pedagogical practices such as self-regulation in learning is a critical challenge for educational research [1]. The term “self-regulated learning” (SRL) refers to learning that is guided by metacognition, planning, self-monitoring, evaluation of personal progress, and motivation to learn [2-5]. SRL is not reducible to simple memorization of facts and involves many elements of higher cognition (in addition to those listed above): reasoning and problem solving, knowledge retrieval and transfer, active construction of new concepts, hypotheses testing, etc. The paradigm of SRL using CBLE becomes more and more popular in educational research and practice and has a high potential value for elementary school [6-9].

Achieving mastery of complex cognitive skills requires intensive teaching and many hours of personal practice [10]. Research on self-regulation of learning shows

that expert learners during independent practice use various self-regulatory processes (strategic goal setting, self-monitoring, and self-evaluation), to achieve peak performance [11, 12]. Self-regulation refers to the degree to which a learner is metacognitively, motivationally and behaviorally active participant of their own learning process [13]. Similarly, extensive research evidence with novice learners shows that setting strategic process goals, self-monitoring, and self-evaluation play an important role in increasing skill, self-efficacy, positive self-reactions and intrinsic interest in the task [14-17]. Assuming the positive effects of self-regulatory processes on students' learning, the question is whether it is possible to improve the quality of novice learners' practice episodes by providing them with training in the use of self-regulatory processes.

Zimmerman [12] identified four levels in students' development of complex skills: (i) observation, (ii) imitation, (iii) self-control and (iv) self-regulation. During the first level, observation, the learner observes a model that possesses expertise performing the skill in an effort to form an image to guide further learning. In the second level, imitation, the learner executes the skill personally with feedback and guidance from the expert model teacher. In the next level, self-control, the learner practices on his/her own. Strategic process goals and self-monitoring facilitate the learner's attainment of skills. Finally, in the last level of learning, self-regulation, the student learns to adapt his/her skills to dynamically changing environments. The students' attention is shifted towards outcomes, and skills are performed without major difficulties. This model is applicable to cognitive skills in general and learning skills in particular. Accordingly, essential elements of ITS-based SRL include: (i) demonstration of a cognitive skill or concept; (ii) adaptive intelligent scaffolding [18-20]; (iii) computer-based self-evaluation; and (iv) usage of the skill in further learning. These elements can be implemented based on the Cognitive Constructor as described below. In addition, our particular interest is in modeling of the entire process of learning, which can be used for designing better learning strategies.

Educational problems and needs related to CBLE-based SRL

While the method of SRL is potentially more powerful than traditional classroom learning, it has its own problems and open questions. One of the needs is intelligent assessment of results of learning. Assessment should be done regularly in the process of learning rather than based on a summative test [9]. Therefore, it may include self-monitoring and peer assessment; however, none of the two methods is objective and sufficiently accurate. In a dream scenario, each student should have an objective, intelligent referee that is always available to student's needs.

The same logic applies to scaffoldings used in SRL. Scaffolding needs to be adaptive, social and intelligent. In particular, the following needs are recognized [18, 21]: (1) adaptive computerized scaffolding supporting different aspects of SRL, including intelligent and metacognitive feedback, and (2) pedagogical agents that can diagnose SRL problems. Similar capabilities are required to help instructors in designing efficient SRL strategies and curricula. In summary, the following needs are recognized by practical users of SRL:

- adaptive, natural, intelligent cognitive and metacognitive feedback to students;
- intelligent, automated assessment of student knowledge and success in learning, the ability to diagnose difficulties and to reveal bottlenecks in SRL;

- the ability to predict student performance before using a new SRL technique, and to modify the paradigm accordingly, in order to maximize the effect of learning.

We explain below that a computational solution to these challenges is feasible and will be practically usable in elementary schools. According to Vygotsky [22], the potential of cognitive development of a child is largely determined by Zone of Proximal Development, which is understood as those cognitive abilities that are available to the child under adult or peer guidance only. Vygotsky [22, 23] also suggested that any intrapersonal functionality develops interpersonally. From this point of view, it follows that it should be possible to provide the necessary social interaction during learning in a virtual form, and a CBLE-based SRL can be efficient, if it provides an adequate to the learning needs adaptive scaffolding that is perceived as a social contact.

In this light, the necessity of a new approach in building CBLE follows from empirical educational studies. For example, Kim and Baylor [21] examine state-of-the-art pedagogical agents from a social-cognitive perspective and conclude that adaptive, intelligent cognitive and metacognitive feedback is still missing in the scaffolding that existing CBLE can provide. Intelligent automated diagnostics of difficulties of SRL is another challenge. Although current state-of-the-art pedagogical agent architectures impress by their cognitive and conversational abilities [24, 25], they still miss the necessary level of intelligence that would be consistent with their design and intended function. This can be illustrated by our sample online session with AutoTutor [25, 26] in which one of us (A.V.S.) participated as Student (Figure 1). While not obvious initially, it becomes obvious later during the session that the agent lacks an understanding of the dialogue to the extent that the illusion of virtual presence is lost (we describe below in this section why this illusion is important).

Another state-of-the-art ITS called Betty's Brain [24] is essentially based on a semantic net: a tool that is very powerful in representing ontologies and causal relations, while at the same time it may not be very convenient for performing qualitative simulations of physical processes and phenomena like those that underlie reasoning in the previous example. These two examples of CBLE indicate problems that exist at the state-of-the-art level. In general, the need for higher intelligence in CBLE, ITS and pedagogical agents is recognized by many authors. Therefore, the general educational problem that needs to be addressed is that of automated assistance to students and instructors in SRL/CBLE paradigms at a high-intelligence level.

Another problem, that seems to stand separately, eventually boils down to the same need of higher intelligence. Kim and Baylor [21] who analyze PALs (here PAL stands for “pedagogical agent as learning companion”) conclude that more natural behavior of PALs is necessary: “First, the naturalness of PAL behavior may be crucial for fostering social relations with learners. The current status of PAL technology is rather limited in creating desirable naturalness. In that regard, second, PALs may require intelligence (e.g., dynamic interaction and adaptive feedback) to substantiate their instructional potential. Currently, technology cannot fully feature intelligent PALs.” [21]. It appears that of the two aspects, (i) realistic human-like appearance and (ii) human-like intelligence, the latter proves to be more crucial for the desired “naturalness” of artificial agents.

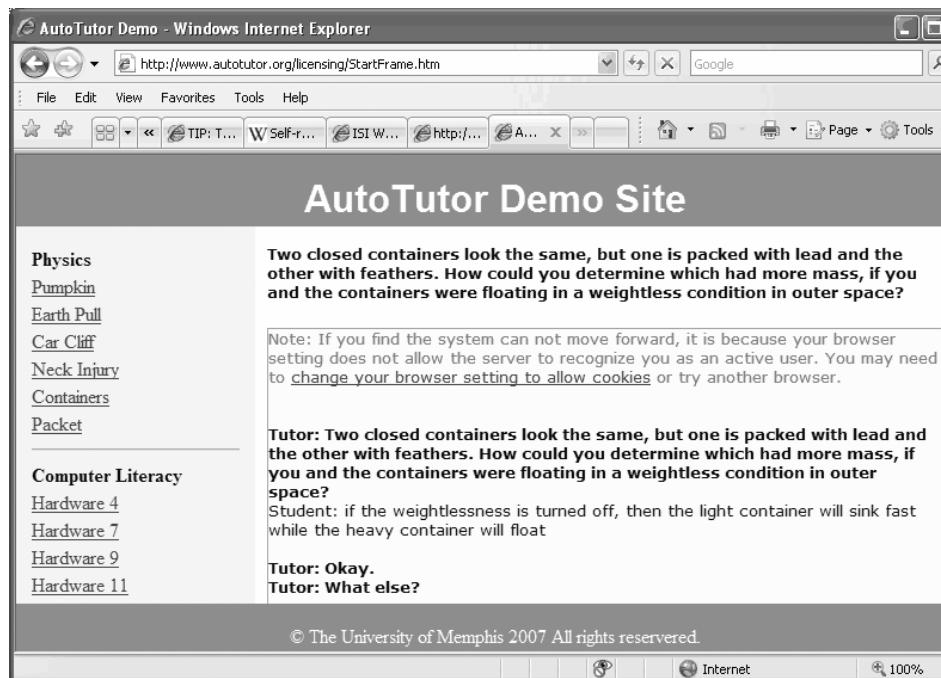


Figure 1. A negative example of a session with an online tutor. The answer of the Student is definitely not “Okay”. In fact, this tutor almost always selects one of several pre-programmed responses, including “Okay”, “Sortof”, “Kindof”, “Good”, etc., and then asks to elaborate more, whenever the input is unexpected.

One piece of evidence supporting this conclusion comes from studies of the effect of “presence break” in virtual reality. The feeling of presence in virtual reality is a well-documented phenomenon that is objectively measurable by psychometrics, behavioral assessment, etc. The same measures are good indicators of breaks of presence [27, 28]: i.e., experiences of a sudden loss of the illusion of virtual reality. The notion of breaks of presence extends to perception of artificial entities embedded in the environment that pose as “alive conscious beings” [29]. Interestingly, the study of Slater and colleagues found that the main determinant of the illusion of genuine agency is not the appearance of the agent (including image, sound and spoken language quality), but the consistency of agent’s behavior with illusion-driven expectations. Here consistency is measured according to human commonsense standards. In other words, an entity that looks, moves and sounds identical to a human will be perceived as an inanimate artifact if its actions make no sense; while on the other hand, a wire frame that is responsive, social and acts intelligently will be perceived as a conscious subject [30, 31]. Consistent with this conclusion observations were recently made by Daniel Levin (Vanderbilt) who found that the appearance of a robot does not affect human judgment of its cognitive/mental capabilities (a talk given at the NCARAI Seminar Series: www.nrl.navy.mil/aic/seminars on October 22, 2007). At the same time, there is strong evidence that feeling of agent’s presence is highly diminished or disappears when participants communicate with the virtual agent through a text interface only [32]. The bottom line here is that human-like intelligence rather than human-like appearance or human-level natural language capabilities should be a goal in designing efficient pedagogical agents.

Therefore, in the present work we analyze a computational model of student learning based on a CBLE called Science Microworld and a GMU BICA agent called in this context Pedagogical Agent, together with specific intervention paradigms, from the point of view of their feasibility and usability in authentic elementary classrooms.

1. Logic model and the intervention

The model of the student mind used to build an intelligent referee and an adaptive scaffolding is grounded theoretically and empirically in recent advances in cognitive science. Currently there are several empirically grounded theoretical and computational frameworks that identify and implement components and cognitive elements underlying human reasoning and learning, including ACT-R [33, 34], Soar [35, 36], EPIC [37], event calculus [38], mental models [39, 40] and other. Our framework based on schemas and mental states [41] comprises their advantages and adds a new essential element: the sense of Self in the cognitive system, as we describe below.

1.1. Computational framework underlying the intervention: GMU BICA

The GMU BICA cognitive architecture is described in a number of our publications [e.g., 41, 46]. It is based on eight highly interconnected components (higher-level symbolic: working memory, semantic memory, episodic memory, and the input-output memory buffer; algorithmic: procedural memory, the driving engine, the reward and punishment system; and connectionist: represented by the cognitive map), and on four building blocks: (i) the Self concept [42], (ii) the formalism of schemas [41, 46], (iii) a neuromorphic cognitive map [47, 48], and (iv) a functionalist mapping of these cognitive components onto brain structures. Our design and specification of these building blocks is based on the current state of the art in cognitive psychology, in neuroscience and in artificial intelligence. Representations in symbolic components are based on schemas, in procedural memory they are hard-coded primitives. The input-output buffer operates in terms of states of schemas and interacts with working memory. The driving engine and the reward-punishment system “run” the above components. They are implemented algorithmically. The central cognitive map component serves to coordinate the higher-level symbolic components and to map their cognitive contents.

1.2. The Self concept, its implementation and functional role in GMU BICA

In the human brain, cognitive processes based on schemas instantiated in working memory constitute mental states of awareness that are attributed to the Self of the cognitive system [42]. This sense of Self fully develops in children by the age 4, and its development coincides with the emergence of a complex of higher cognitive abilities [43]. The cognitive process of SRL is driven by the sense of Self, occurs through voluntary actions of Self [44] and subsequently becomes stored in long-term memory as a personal experience of Self, resulting in formation of episodic memories that are attributed to Self [45]. Therefore, implementing the sense of Self in a cognitive system [42] must be vital for modeling human learning. Our computational framework [41] is unique in the sense that it is based on a human-like concept of a Self [42], together with related to it mental states and episodic memories that are attributed to the Self. While the role of a Self may not be noticeable in many traditional experimental paradigms

based on laboratory settings, it becomes more evident when it enables autonomous cognitive growth in a long-term learning scenario that characterizes real classroom settings, or human-like truncation of the theory of mind exploited in a dialogue with a human. This is why we believe that our approach provides a better ground for modeling student learning than other existing computational approaches (e.g., CLARION [51]).

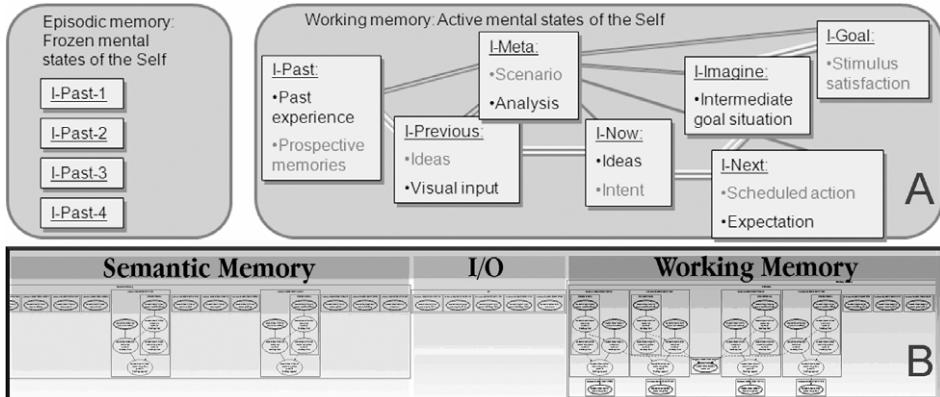


Figure 2. Snapshots of GMU BICA top-level dynamics. **A:** Theoretical snapshot of working (right) and episodic (left) memory systems of GMU BICA. Mental states (light boxes) have self-explanatory labels, they instantiate the Self of the system. The light double line shows the main sequence of mental states called the working scenario. **B:** Actual snapshot of semantic, working and input-output memories of a GMU BICA rapid prototype in a paradigm when the agent is exploring an outdoor environment. Ovals represent instances of schemas. Only a small part of semantic memory is shown.

The implementation of the concept of Self in our architecture is based on a framework including mental states as objects (Figure 3). As a consequence, working memory in our architecture (Figure 2) is partitioned into a number of mental states (Figure 2 A), such that each mental state corresponds to an instance of a Self of the agent taken at a unique perspective (labels “I-Now”, “I-Previous”, “I-Next”, etc. are self-explanatory). The key steps in this implementation are: (i) axioms of the Self [42], or semantic constraints, that are enforced by the architecture design and by the dynamic rules; (ii) a mental state, or a snapshot of awareness of the Self; (iii) the mental state lattice and a working scenario; and finally, (iv) working and episodic memory systems composed of mental states and organized by the cognitive map.

A *mental state* can be intuitively understood as a particular subjective perspective (i.e., a view from a certain point in space, certain moment of time, certain agent's identity, certain scale of distances, etc.) taken together with all experiences that the subject has (or had, or will have) at that given moment. This notion is an abstraction, because there is no “subject” in our system. However, when this abstraction controls behavior of functional units that are connected to sensors and actuators, then agent's behavior may create an illusion that there is one subject in the system. In this sense, our self is both emergent and innate at the same time.

Instead of one, our architecture has a dynamical network of mental states, several of which can be active simultaneously and interact with each other. From the technical point of view, a mental state in our framework is a limited set of (mutually bound) instances of schemas associated with one and the same mental perspective. Their evolution in physical time is called mental simulation. The closest framework would be

event calculus [38]; however, there are substantial differences. E.g., fluents in event calculus are represented by predicates, our representations are instances of our schemas that have many standard attributes (Figure 3 A). E.g., the attribute called “attitude” allows the agent to distinguish between real, imaginary, past, future, desired, obligatory, intended, etc. objects and events.

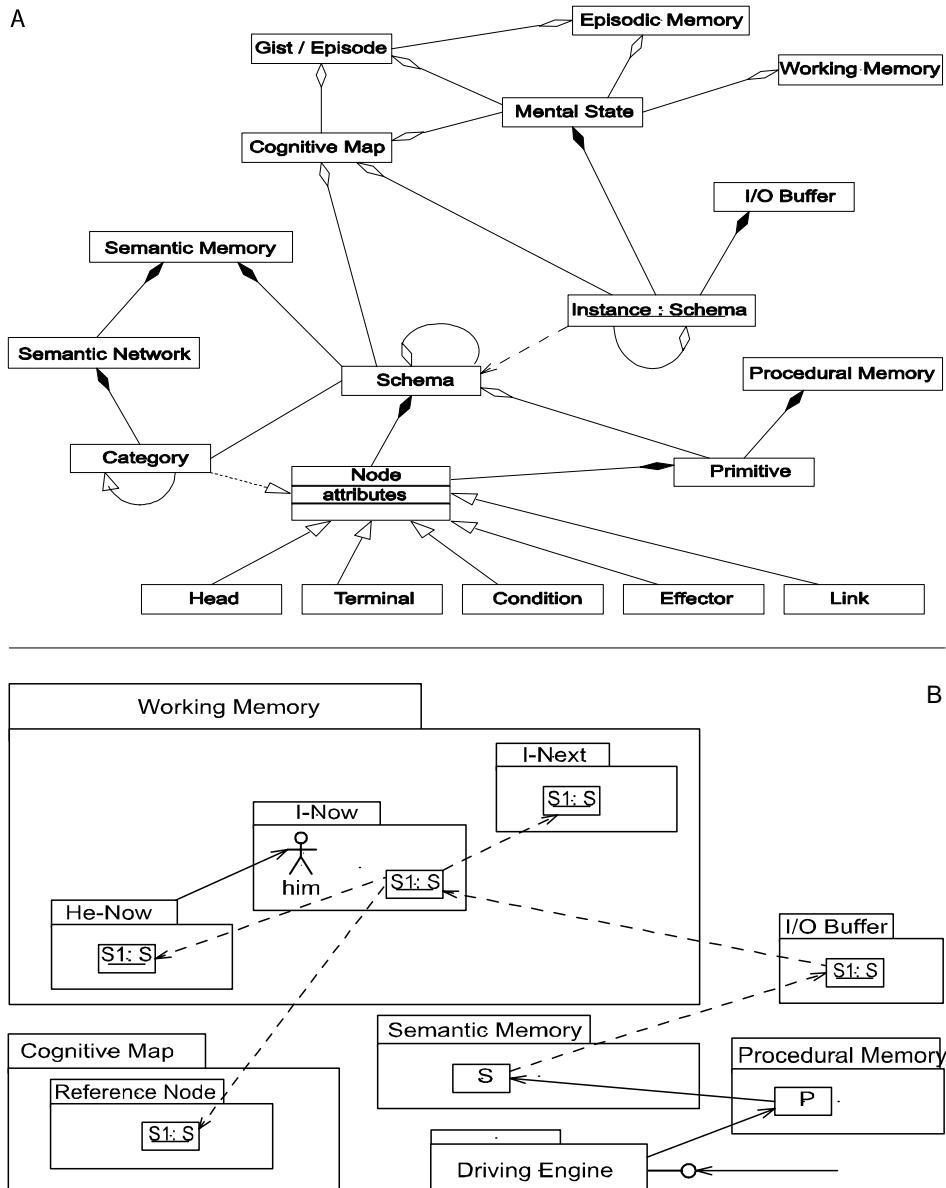


Figure 3. GMU BICA. **A:** a UML class diagram explaining the framework of schemas and mental states. Schemas and their instances are represented separately, by objects in the same format. This diagram does not show interactions among components. **B:** a snapshot of a process of sensory perception with shared attention.

Sensory perception with shared attention is an example illustrating implementation of sensory input, Theory of Mind and interactions of mental states in our architecture. Suppose the system is aware of another agent in the same environment. That agent is represented in working memory by a mental state He-Now and an instance of the agent body schema in I-Now, to which He-Now is attributed (Figure 3 B). An incoming sensory signal is received by the driving engine, which sends it to a primitive, which in turn instantiates the schema of the perceived entity in the input/output buffer (IO). At the next step, the perceived instance in IO is copied into I-Now and from there to He-Now and to He-Next (if the entity is likely to persist at the next moment of time). If the entity is likely to persist for a long time, then the instance is copied to reference memory, which is a part of the neuromorphic memory system (cognitive maps).

In summary, a mental perspective (an instance of self) is characterized by the subject identity, status, moment of time, location in space, etc. A mental state is an instance of a self together with all attributed experiences. Therefore, our notion of *self-awareness* is the following: “the self understood as an idealized abstraction is indirectly represented in the cognitive system by mental states attributed to it” (these are instances of an abstract cognitive construct), rather than “the system reflects itself or some of its aspects: the body, the software, etc.” Our system has multiple representations of mental states that may reflect each other, while none of them can be called “the original Self” (e.g., Figure 3 B). Thus, the Self in our system is “abstract” or “illusory”, it is “innate” and “emergent” at the same time. Furthermore, this abstraction is never represented explicitly at the higher level as a structure or a set of mechanisms having internal parts. Instead, the abstraction itself can only be represented explicitly by an atomic token [42]. Mental state labels, their relationships and functional roles change over time: there is no fixed architecture in terms of mental states.

1.3. The intervention

The intervention is based on two computational components that should be used in combination with each other (Figure 4). The first component is a specialized computer-based learning environment that we call Science Microworld: it allows a student to construct processes of reasoning and learning on a computer screen. The second component is a Pedagogical Agent implemented as a GMU BICA agent: it is an automated learner and problem solver that operates within the Science Microworld environment and interacts with a human user. Both components are innovative and unique in educational research. Science Microworld can be developed first and tested independently of Pedagogical Agent.

In essence, Science Microworld consists of a schema database (semantic memory) representing facts and rules relevant to the curriculum, which is linked to a graphical and text-based interface. This interface can be called a *Harry-Potter-style interface*, because the principle of its operation is the following: select an object (using a mouse or screen touch) and type a phrase referring to the object, which will invoke the associated schema (as illustrated in Figure 5). A schema in Science Microworld is the same universal building block that is used for cognitive representations in the GMU BICA-based Pedagogical Agent. Science Microworld has an interface component that represents elements of cognition by symbols on a computer screen, making them available for entering, editing, processing, analyzing and storing in computer memory. It provides minimal automated functions, including consistency check and derivation of immediate results when an available schema is applied. It can be used for problem

solving and for development of new knowledge using SRL, allowing a user to build (and to store as schemas) new concepts and skills based on previously developed concepts and skills.

Pedagogical Agent, which is the BICA agent placed on top of Science Microworld, will make the simulated processes of reasoning and learning fully automated, using our previously designed cognitive architecture GMU BICA described above. Given the Science Microworld environment, a cognitive task of learning or problem solving becomes similar to a path-finding task of a kind that our GMU BICA previously solved [46]. In addition, Pedagogical Agent will be able to monitor student performance in Science Microworld and to simulate mental states of the student, using the same framework that it uses to simulate its own mental states (the same principle presumably works in the human brain: [47]). Therefore, the intended function of the Cognitive Constructor (consisting of Science Microworld and Pedagogical Agent) is three-fold:

- a self-regulated learning tool providing adaptive scaffolding and peer advising for students;
- a diagnostic tool for intelligent and possibly implicit evaluation of student knowledge and learning skills; and
- a simulator of student learning performance (relying on up-to-date individual student models) used as a guide for an instructor in re-designing teaching strategies and planning lectures.

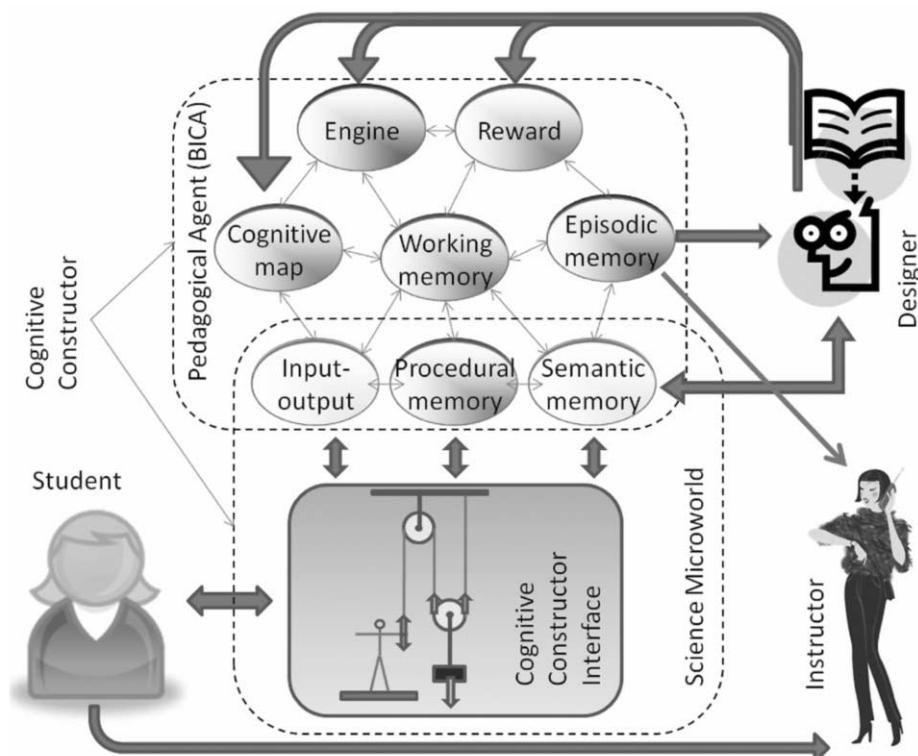


Figure 4. Logic model for the intervention using Cognitive Constructor. A student interacts with Cognitive Constructor via the Internet, using Cognitive Constructor Interface. Student's activity is monitored and guided by Pedagogical Agent – an agent based on GMU BICA architecture.

2. Examples of Cognitive Constructor in Action

Here we consider two illustrative examples, one of which we implemented as a rapid prototype. Both learning episodes can be used as a part of an elementary school science course and performed by students over the Internet using Cognitive Constructor, as a homework assignment.

Example 1: States of Water

The episode starts by Cognitive Constructor presenting several facts to the student:

- Water can exist as ice (solid), water (liquid), or vapor (gas).
- A solid object retains its shape.
- Liquid retains its volume.
- Gas occupies available space.
- Water vapor is not visible to the human eye.
- States of water can be turned into one another by changing temperature.
- The amount of water is preserved over time and upon state transitions.

All these facts will be stored based on schemas that will have graphical representations in Science Microworld. During initial presentation, each fact will be illustrated by a picture, a diagram, or a short animation. The next step is to consider a possible experiment with water and its outcome, also represented by Cognitive Constructor in its Science Microworld component.

- Water left in a closed container in a room remains in the same amount over days.
- Water left in an open container in a room disappears over days.

Now the student and, in parallel, the Pedagogical Agent, are supposed to use available facts (schemas) in order to hypothesize an explanation and make a plausible generalization, thereby learning a new general fact. Here is an outline of this process that can be *constructed* by a student on the computer screen:

- Water in an open container could change its state (apply the schema to water in container, using the *Harry-Potter-style interface*).
- If it turned into ice, it would still be there (see result produced by Science Microworld).
- If it turned into vapor it would go into the room (see result when the schema is applied).
- Vapor would not be visible (assert, using the *Harry-Potter-style interface*).
- Hypothesis (possible answer): water turned into vapor.
- Generalization (learning): if so, then water can turn into vapor at room temperature (represent as a new schema).

The hypothesis that water can turn into vapor at room temperature (the schema candidate) needs to be tested by further consideration of familiar phenomena and by new observations (or mentally simulated experiments). All steps in this scenario can be implemented based on schemas using the Cognitive Constructor framework.

Example 2: A Pulley Problem

This our implemented rapid prototype of Science Microworld and its usage illustrate the spirit of the proposed intervention. Details of the scenario are represented by a sequence of screenshots (Figure 4). The text input supplied by the student in this

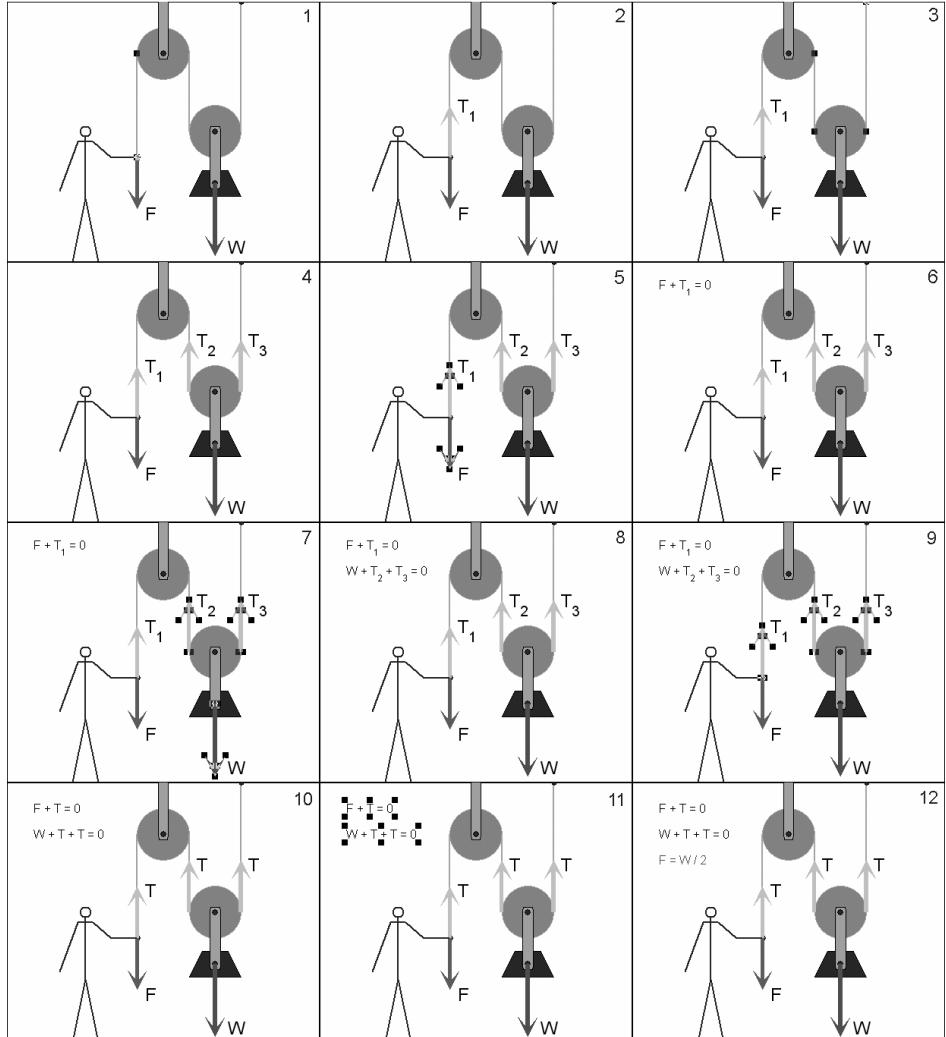


Figure 5. Simulation of a rapid prototype of Cognitive Constructor. Figure panels 1 – 12 show the actual sequence of screen snapshots taken from a session of a rapid prototype of Cognitive Constructor implementing a pulley problem. The task is to relate the force F to the weight W . It takes six steps to solve the problem. At the first step (1), the user selects the leftmost vertical segment of the rope (the fact of selection is indicated by small black squares at the extremities of the object) and types on console: *Show the rope tension*. The effect of this input is that Cognitive Constructor draws a gray arrow (the original demo is in color) representing the tension of the rope, assigns a variable T_1 to it, and places the label next to the arrow (2). At the next step (3), the user selects the remaining two segments of the rope and types: *Show tension of the rope*. Again, Cognitive Constructor performs the requested action (4). At the next step (5), the user selects two arrows representing the applied force and the rope tension and types: *These forces are in equilibrium (therefore their sum equals zero)*. The result is that Cognitive Constructor writes the corresponding equation in the top left corner (6). At the next step (7), the user selects three forces: T_2 , T_3 and W , and types: *These forces are also in equilibrium*. As a result, Cognitive Constructor writes another equation (8). At the next step (9), the user selects all arrows representing the rope tension and types: *The tension of the rope is constant therefore forces are equal*. As a result, Cognitive Constructor changes all three variables to T : in the diagram and in the two equations (10). Finally, the user selects the equations and types: *Based on the equations, express F in terms of W* (11). Then Cognitive Constructor produces the solution (12). The implemented rapid prototype allows for any valid strategy that may be used to solve the problem.

session is given below each panel. We can see that in this case the processing of input is robust, and the process of problem solving is reduced to a construction process performed by the student using our *Harry-Potter-style interface*. All elements of reasoning, from very concrete (blocks and ropes) to abstract (e.g., equilibrium of forces) are represented as virtual objects at the same level, available for manipulation.

3. Comparison with Existing Approaches and Future Perspectives

First and foremost, we should emphasize that our approach is not competing with the existing state of the art in intelligent tutoring. On the contrary, we are building on the state of the art, its lessons and its underlying developments. As we lay the ground for achieving new goals, we bring together advances in cognitive psychology, in pedagogy, and in artificial intelligence, in particular, including self-regulation and other metacognitive learning techniques that prove to be useful in schools, methods of bootstrapped learning that emerge in artificial intelligence, existing feasibility and usability of CBLE in elementary school as a starting point for our development.

There is substantial experimental evidence supporting the feasibility and usability of approaches based on self-regulated learning. Extensive research with novice learners shows that setting strategic process goals, self-monitoring, and self-evaluation play an important role in increasing skill, self-efficacy, positive self-reactions and intrinsic interest in the task [14-17]. GMU BICA is a perfect match for SRL, because it offers metacognitive abilities based on its built-in theory of mind and the Self concept.

Finally, among all practical applications of advanced cognitive architectures developed in artificial intelligence, pedagogical applications start taking the leading role. For example, one of the most widely known biologically inspired cognitive architectures is ACT-R [32] underlies Cognitive Tutors for Mathematics that are now used in thousands of schools across the United States. GMU BICA shares features with ACT-R, Soar [35], EPIC [37], LIDA [50], CLARION [51], Polyscheme [52] and other cognitive architectures, and it goes to the next level by introducing the unique notion of Self and enabling related to it metacognitive functionality. Speaking generally, based on predictions of next steps in artificial intelligence [53, 54], we anticipate a global paradigm shift in education in the near future.

To complete the picture, we should point to a related prospective application of GMU BICA in the field of natural language acquisition. Of particular practical interest is a paradigm in which the agent is exposed to virtually unlimited electronic text resources and learns general world knowledge (together with the language itself) step by step, starting with minimal semantic knowledge and minimal linguistic capabilities (an interesting question is how big is the necessary minimum, or the “critical mass” [55]). It is arguable that, regardless of the starting level of intelligence, in order to be successful in this paradigm, the agent needs to (a) interact with a tutor or a peer, and (b) be embodied, at least virtually, at least in the sense of mental simulation ability, so that new acquired knowledge can be tested and interpreted practically. The cognitive system per se may include three modules: a syntactic and semantic parser on its input, a formal reasoning component with ontological representations as the output, and the GMU BICA architecture as the core. The mechanism of text understanding can be based on multi-level application of schemas of increasing depth of analysis, while the mechanism of learning will consist in creation of new schemas based on experience and associating them with new vocabulary. In this scenario, linguistic knowledge is

acquired in parallel with commonsense knowledge. Similar principles can be used to generate a text output, which would facilitate communications with a human instructor.

In this scenario, the presence of a Self in the agent understood as outlined above becomes critical. Firstly, the agent needs it in order to be able to understand a human peer or instructor, and also the study material, if it involves social aspects. Secondly, in order to be able to grove cognitively being driven by internal stimuli, the agent needs the ability to develop *dreams* (i.e., imaginary situations, capabilities and scenarios), store them in episodic memory, and construct a system of values of them. Theoretically, all this could be done based on formal logical reasoning about conceptual knowledge; however, practically this scenario appears highly unlikely, because there are too many possibilities to explore. In order to succeed in truncation of possibilities, the agent needs the guidance of an anthropomorphic sense of Self. The mental state template and the minimal Self axioms constitute only the starting point in this process. A grown-up self-awareness and self-understanding in the agent will emerge through the process of cognitive development up to a point when external reward and punishment will no longer be necessary as a driving force. The agent will be driven by its own, self-generated system of values and associated with them emotions. In particular, social interactions will become one of the primary needs.

Through these developments, GMU BICA will be able to provide a detailed, quantitative computer model of human cognition, human development and human subjective experience. Joined with its counterpart [56], one day this approach may lead to creation and successful authentication of consciousness in an artificial general intelligence system.

References

- [1] Hadwin, A. F., and Winne, P. H. (2001). CoNoteS2: A software tool for promoting self-regulation. *Educational Research and Evaluation* 7 (2-3): 313-334.
- [2] Butler, D. L. and Winne, P.H. (1995). Feedback and self-regulated learning: A theoretical synthesis. *Review of Educational Research* 65: 245-281.
- [3] Winne, P.H., and Perry, N.E. (2000). Measuring self-regulated learning. In P. Pintrich, M. Boekaerts, and M. Seidner (Eds.), *Handbook of self-regulation*, pp. 531-566. Orlando, FL: Academic Press.
- [4] Perry, N.E., Phillips, L., and Hutchinson, L.R. (2006). Preparing student teachers to support for self-regulated learning. *Elementary School Journal* 106: 237-254.
- [5] Zimmerman, B.J. (1990). Self-regulated learning and academic achievement: An overview. *Educational Psychologist* 25: 3-17.
- [6] Baylor, A.L. (2002). Expanding preservice teachers' metacognitive awareness of instructional planning through pedagogical agents. *Educational Technology Research and Development* 50: 5-22.
- [7] Kitsantas, A., and Dabbagh, N. (2004). Promoting self-regulation in distributed learning environments with web-based pedagogical tools: An exploratory study. *Journal of Excellence in College Teaching*, 15 (1-2), 119-142.
- [8] Dabbagh, N., and Kitsantas, A. (2005). Using web-based pedagogical tools as scaffolds for self-regulated learning. *Instructional Science*, 33(5-6), 513-540.
- [9] Dabbagh, N., and Bannan-Ritland, B. (2005). *Online Learning: Concepts, Strategies, and Application*. Pearson Education, Inc.
- [10] Ericsson, K. A., and Charness, N. (1994). Expert performance: Its structure and acquisition. *American Psychologist* 49: 725-747.
- [11] Cleary, T.J., and Zimmerman, B.J. (2006). Teachers' perceived usefulness of strategy microanalytic assessment information. *Psychology in the Schools* 43 (2): 149-155.
- [12] Kitsantas, A., Zimmerman, B.J., (2000).
- [13] Zimmerman, B. J. (2000). Attaining self-regulation: A social-cognitive perspective. In M. Boekaerts, P. Pintrich, and M. Seidner (Eds.), *Self-regulation: Theory, Research, and Applications* (pp.13-39). Orlando, FL: Academic Press.

- [14] Zimmerman, B. J., and Kitsantas, A. (1996). Self-regulated learning of a motoric skill. The role of goal setting and self-monitoring. *Journal of Applied Sport Psychology* 8: 69-84.
- [15] Zimmerman, B. J., and Kitsantas, A. (1997). Developmental phases in self-regulation. Shifting from process goals to outcome goals. *Journal of Educational Psychology* 89: 29-36.
- [16] Kitsantas, A., and Zimmerman, B. J. (1998). Self-regulation of motoric learning: A strategic cycle view. *Journal of Applied Sport Psychology* 10: 220-239.
- [17] Kitsantas, A., Zimmerman, B.J., and Cleary, T. (2000). The role of observation and emulation in the development of athletic self-regulation. *Journal of Educational Psychology* 92 (4): 811-817.
- [18] Azevedo, R., and Hadwin, A. F. (2005). Scaffolding self-regulated learning and metacognition - Implications for the design of computer-based scaffolds. *Instructional Science*, 33(5-6), 367-379.
- [19] Azevedo, R., Cromley, J. G., and Seibert, D. (2004). Does adaptive scaffolding facilitate students' ability to regulate their learning with hypermedia? *Contemporary Educational Psychology*, 29(3), 344-370.
- [20] Azevedo, R., Cromley, J. G., Winters, F. I., Moos, D. C., and Greene, J. A. (2005). Adaptive human scaffolding facilitates adolescents' self-regulated learning with hypermedia. *Instructional Science*, 33(5-6), 381-412.
- [21] Kim, Y., and Baylor, A. L. (2006). A social-cognitive framework for pedagogical agents as learning companions. *Educational Technology Research and Development*, 54(6), 569-596.
- [22] Vygotsky, L.S. (1978). *Mind in Society*. Cambridge, MA: Harvard University Press.
- [23] Vygotsky, L. S. (1979). The collected works of L. S. Vygotsky: Vol. 4, The history of the development of higher mental functions. R. W. Rieber, Ed. M. J. Hall, Trans. New York: Plenum. (Originally written 1931.)
- [24] Biswas G., Leelawong K., Schwartz D., and Vye N. (2005). Learning by teaching: A new agent paradigm for educational software. *Applied Artificial Intelligence* 19: 363-392.
- [25] Graesser, A.C., Jackson, G.T., and McDaniel, B. (2007). AutoTutor holds conversations with learners that are responsive to their cognitive and emotional states. *Educational Technology* 47: 19-22.
- [26] VanLehn, K., Graesser, A.C., Jackson, G.T., Jordan, P., Olney, A., and Rose, C.P. (2007). When are tutorial dialogues more effective than reading? *Cognitive Science* 31: (1) 3-62.
- [27] Slater, M., and Usoh, M. (1994). Depth of presence in virtual environments. *Presence-Teleoperators and Virtual Environments*, 3 (2): 130-144.
- [28] Usoh, M., Catena, E., Arman, S., and Slater, M. (2000). Using presence questionnaires in reality. *Presence-Teleoperators and Virtual Environments* 9(5): 497-503.
- [29] Herrera, G., Jordan, R., and Vera, L. (2006). Agency and presence: A common dependence on subjectivity? *Presence: Teleoperators and Virtual Environments* 15 (5): 539-552.
- [30] Slater, M., and Sanchez-Vives, M. (2006). Presence and consciousness in virtual environments: Workshop-tutorial given at Tucson-2006 "Toward a Science of Consciousness" Conference. In S. Hameroff et al., editors, *Toward a Science of Consciousness 2006, Consciousness Research Abstracts: A Service from Journal of Consciousness Studies*, page 5. Tucson, AZ: Center for Consciousness Studies and Thorverton, UK: Imprint Academic.
- [31] Slater, M., Pertaub, D. P., Barker, C., and Clark, D. M. (2006b) An experimental study on fear of public speaking using a virtual environment, *Cyberpsychology and Behavior* 9(5), 627-633.
- [32] Slater, M., Antley, A., Davison, A., Swapp, D., Guger, C., Barker, C., Pistrang, N., Sanchez-Vives, M. V. (2006a). A virtual reprise of the Stanley Milgram obedience experiments. *PLoS ONE* 1 (1): e39. doi:10.1371/journal.pone.0000039. (www.plosone.org)
- [33] Anderson, J. R., and Lebiere, C. (1998) *The Atomic Components of Thought*. Mahwah: Lawrence Erlbaum Associates.
- [34] Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., and Qin, Y. (2004). An integrated theory of the mind. *Psychological Review* 111 (4): 1036-1060.
- [35] Newell, A. (1990) *Unified Theories of Cognition*. Cambridge, MA: Harvard University Press.
- [36] Laird, J.E., Rosenbloom, P.S., and Newell, A. (1986). *Universal Subgoal and Chunking: The Automatic Generation and Learning of Goal Hierarchies*. Boston: Kluwer.
- [37] Meyer, D.E., and Kieras, D.E. (1997). A computational theory of executive cognitive processes and multiple task performance: Part I. Basic mechanisms. *Psychological Review* 63, 81-97.
- [38] Mueller, E. T. (2006). *Commonsense Reasoning*. San Francisco, CA: Morgan Kaufmann Publishers.
- [39] Johnson-Laird, P. N. (1983). *Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness*. Cambridge, MA: Harvard University Press.
- [40] Gentner, D., and Stevens, A.L., Eds. (1993). *Mental Models*. Hillsdale, NJ: Lawrence Erlbaum Assoc.
- [41] Samsonovich, A. V., and De Jong, K. A. (2005). Designing a self-aware neuromorphic hybrid. In K. R. Thorisson, H. Vilhjalmsson and S. Marsela (Eds.), *AAAI-05 Workshop on Modular Construction of Human-Like Intelligence: AAAI Technical Report* (Vol. WS-05-08, pp. 71-78). Menlo Park, CA: AAAI Press.

- [42] Samsonovich, A. V., and Nadel, L. (2005). Fundamental principles and mechanisms of the conscious self. *Cortex*, 41 (5): 669-689.
- [43] Moore, C., and Lemmon, K., Eds. (2001). *The Self in Time: Developmental Perspectives*. Mahwah, NJ: Lawrence Erlbaum Associates.
- [44] Maasen, S., Prinz, W., and Roth., G. (2003). *Voluntary Action: Brains, Minds, and Sociality*. Oxford, UK: Oxford UP.
- [45] Tulving, E. (1983). *Elements of Episodic Memory*. Oxford: Oxford University Press.
- [46] Samsonovich, A. V. Ascoli, G. A., De Jong, K. A., and Coletti, M. A. (2006). Integrated hybrid cognitive architecture for a virtual roboscout. In M. Beetz, K. Rajan, M. Thielscher, and R.B. Rusu, editors. *Cognitive Robotics: Papers from the AAAI Workshop, AAAI Technical Reports*, volume WS-06-03, pp. 129–134. Menlo Park, CA: AAAI Press.
- [47] OKeefe, J., and Nadel, L. (1978) *The Hippocampus as a Cognitive Map*. Clarendon, New York, NY.
- [48] Samsonovich, A. V., and Ascoli, G. A. (2005). A simple neural network model of the hippocampus suggesting its pathfinding role in episodic memory retrieval. *Learning & Memory*, 12 (2): 193-208.
- [49] Nichols, S., and Stich, S. (2003). *Mindreading: An Intergrated Account of Pretence, Self-Awareness, and Understanding Other Minds*. Oxford: Oxford University Press.
- [50] Franklin, S., (2007). A foundational architecture for artificial general intelligence. In B. Goertzel and P. Wang (Eds.). *Advances in Artificial General Intelligence: Concepts, Architectures and Algorithms. Proceedings of the AGI Workshop 2006. Frontiers in Artificial Intelligence and Applications*, vol. 157, pp. 36-54. IOS Press: Amsterdam, The Netherlands.
- [51] Sun, R. (2004). The CLARION cognitive architecture: Extending cognitive modeling to social simulation. In: Ron Sun (Ed.), *Cognition and Multi-Agent Interaction*. Cambridge UP: New York.
- [52] Cassimatis, N.L., Trafton, J.G., Bugajska, M.D., and Schultz, A.C. (2004). Integrating cognition, perception and action through mental simulation in robots. *Journal of Robotics and Autonomous Systems* 49 (1-2): 13-23.
- [53] Albus, J. S., and Meystel, A. M. (2001). *Engineering of Mind: An Introduction to the Science of Intelligent Systems*. New York: Wiley.
- [54] Samsonovich, A. V., and Ascoli, G. A. (2002). Towards virtual brains. In G. A. Ascoli (Ed.), *Computational Neuroanatomy: Principles and Methods*, pp. 423-434. Totowa, NJ: Humana.
- [55] Samsonovich, A. V. (2007). Universal learner as an embryo of computational consciousness. In: Chella, A., and Manzotti, R. (Eds.). *AI and Consciousness: Theoretical Foundations and Current Approaches. Papers from the AAAI Fall Symposium*. AAAI Technical Report FS-07-01, pp. 129-134. Menlo Park, CA: AAAI Press.
- [56] Samsonovich, A. V. and Ascoli, G. A. (2007). Cognitive map dimensions of the human value system extracted from natural language. In Goertzel, B. and Wang, P. (Eds.). *Advances in Artificial General Intelligence: Concepts, Architectures and Algorithms. Proceedings of the AGI Workshop 2006*, pp. 111-124. IOS Press: Amsterdam, The Netherlands.

Transfer Learning and Intelligence: an Argument and Approach

Matthew E. TAYLOR, Gregory KUHLMANN, and Peter STONE

Department of Computer Sciences

The University of Texas at Austin

{mtaylor, kuhlmann, pstone}@cs.utexas.edu

Abstract. In order to claim fully general intelligence in an autonomous agent, the ability to learn is one of the most central capabilities. Classical machine learning techniques have had many significant empirical successes, but large real-world problems that are of interest to generally intelligent agents require learning much faster (with much less training experience) than is currently possible. This paper presents *transfer learning*, where knowledge from a learned task can be used to significantly speed up learning in a novel task, as the key to achieving the learning capabilities necessary for general intelligence. In addition to motivating the need for transfer learning in an intelligent agent, we introduce a novel method for selecting types of tasks to be used for transfer and empirically demonstrate that such a selection can lead to significant increases in training speed in a two-player game.

Keywords. Transfer Learning, Game Tree Search, Reinforcement Learning

1. Introduction

A generally intelligent agent deployed in any non-trivial environment must be able to learn: no designer can anticipate every possible encountered situation. Specifically, intelligent agents need to learn how to *act*, which is the purview of *Reinforcement learning* [1,2]. Reinforcement learning (RL) problems are defined as those in which learning agents sequentially execute actions with the goal of maximizing a reward signal, which may be time-delayed. The RL framework appeals when considering the design of a generally intelligent agent as it is established, flexible, and powerful.

RL approaches have been gaining in popularity in recent years as methods have matured that are able to handle complex problems with noisy sensors, noisy actuators, and continuous state spaces (e.g., helicopter control [3] and Keepaway [4]). One of the main advantages to RL, unlike many machine learning approaches, is that no labeled examples are required. Such flexibility allows learning from general environmental feedback (a *reward signal*) but it comes at a price. When RL methods begin learning without any background knowledge, mastering difficult tasks may be slow or infeasible. If an agent requires months or years worth of experience to master relatively simple tasks, it will be a stretch to claim that it is generally intelligent. Thus a critical component to achieving intelligent behavior is not only the ability to successfully learn, but also the ability to learn *quickly*, from limited training experience.

In order to utilize RL algorithms to control a generally intelligent agent, we must overcome three shortcomings, one or more of which are typically present in current applications. Firstly, RL algorithms have typically been applied to simple tasks, such as the discrete task of gridworld. Secondly, many algorithms are sample-inefficient and require millions of training examples in order to perform well [5]. Thirdly, substantial amounts of human knowledge must often be used in order to define the learning problem and direct the learner towards good solutions [3,4]. How can we hope to allow an agent to learn complex tasks, without human guidance, and with relatively few examples?

The key difference between traditional RL settings and that of our hypothetical generally intelligent agent is that our agent would be able to leverage experience gained in previous related tasks. The idea of *lifelong learning* has been studied before in traditional machine learning [6] but it has only recently been applied to reinforcement learning, which is of specific interest to general intelligence.

Consider, for example, a generally intelligent household transportation assistant. While current agents are typically special purpose, a generally intelligent agent would be required to act in multiple related tasks. Such an agent may be expected to: retrieve food for a meal from the grocery store, drive the children to school, and retrieve the parents from work. All of these tasks utilize skills for operating a vehicle, obeying traffic laws, scheduling, and navigation. Due to this overlap, one would expect training on subsequent transportation tasks to take substantially less time than on the first such task. If the agent is tasked with a new request, such as delivering laundry to the cleaner, the users would expect the agent to quickly master the new task due to similarities with previously learned tasks. We next describe *transfer learning* for reinforcement learning, a general approach that would allow an RL agent to leverage past knowledge when learning novel tasks, potentially enabling effective lifelong learning for a generally intelligent agent.

In the transfer learning paradigm a learner is presented a novel *target task*. The agent may elect to first train on a (set of) simpler *source task(s)* rather than learning the target task directly. A typical goal of transfer is to reduce the time needed to learn a target task after first learning a source task, relative to learning the target task without transfer. This *target task goal* can be achieved whenever the learner can effectively transfer useful information from the source task into the target task. A more difficult goal is to reduce the total training time so that learning the source task and target task is faster than learning the target task directly. Such a *total time goal* is attainable only if the source task is faster to solve than the target task, and the speedup in target task training time overcomes the time spent on learning the source task.

Transfer between tasks has long been studied in humans [7] and the ability to transfer knowledge between different tasks is one reasonable criterion for intelligence. School curricula are designed around the principle of developing students' abilities and increasing the knowledge gradually over time. For successful autonomous transfer, an agent must effectively identify analogies between different tasks.¹ Hofstadter even argues [9] that analogical reasoning is the core of intelligence because humans form, over their lifetime, a *mental lexicon* of categories and information by using analogies.

In order to enable autonomous transfer, the agent must:

1. select a source task appropriate for a given target task,
2. effectively transfer knowledge from the source task into the target task.

While there has been recent work on step #2 [10,11], relatively little work has concentrated on the more difficult step of task selection. Such a selection ability will be necessary if the agent is to determine which source tasks to train on before tackling a more difficult target task, or if the agent has experienced multiple source tasks and it must select a subset which are most similar to the current target task.

In order to facilitate the selection of source tasks for a particular target task, this work introduces the concept of a *transfer hierarchy*. Such a structure defines types of tasks that require more or less information to solve and can be used to rank tasks by

¹Other research [8] suggests that humans are not reliably proficient at discovering analogies between very dissimilar tasks unless prompted that such an analogy exists.

their relative solution complexity. Such a task ordering can be used to identify source tasks that will take significantly less time to solve than a particular target task, reducing the impact of source task training on the total training time. Our hope is that such a hierarchy will be useful in future work where transfer learners *automatically* select a source task for a given target task. In this paper we begin to evaluate the effectiveness of our proposed hierarchy by manually constructing source tasks for a specified target task where the selection of source tasks are motivated by the transfer hierarchy.

To empirically demonstrate transfer between source and target tasks taken from our transfer hierarchy, we utilize the game of *Mummy Maze*. This game is an appropriate choice for two reasons. First, it has been released as a sample domain in the *General Game Playing* [12] (GGP) contest, an international competition developed independently at Stanford. Second, the Mummy Maze task is easily modifiable so that it can conform to each task type in our transfer hierarchy. Our results show that a transferred heuristic is able to significantly improve the speed of search, even if the generated source tasks differ from the target tasks along a number of dimensions. We show both that transferred knowledge can effectively reduce the amount of time required to learn the target task, and that the total time required to learn the target task may be reduced by first training on a set of simpler source tasks. This result is a small but important step towards autonomous transfer in both planning and RL domains, which we believe to be on the critical path for development of a generally intelligent agent.

2. A Transfer Hierarchy

Mapping problem characteristics to the correct solution type is an important open problem for AI. Given a control problem, should the solution be solved optimally or approximately? Is planning or RL more appropriate? If RL, should the solution be model-based or model-free? This work assumes that such an appropriate mapping exists; given certain characteristics of a game, we propose an appropriate solution method. The characteristics we select are based on the amount of information provided to a player about the game's environment and opponent.

For instance, if a learner has a full model of the effects of actions and knows how its opponent will react in any situation, the learner may determine an optimal solution by “thinking” through the task using *dynamic programming* [13] (DP). At the other extreme, a learner may have to make decisions in a task where the opponent’s behavior is initially unknown and possibly stochastic. In this more difficult scenario, the solution strategy must work to sample the environment and opponent’s policy repeatedly, which suggests an RL approach.

Interactions with the environment and an opponent accrue cost: simulators use computational resources, physical robots may take significant amounts of wall-clock time, and opponents think before making decisions. When using DP, the only cost is cycles spent determining an optimal policy. When using RL, one must account for both interactions with the environment and opponent. By considering these differences in resource requirements, we propose a hierarchy to define game characteristics which require more resources to solve. We then leverage the solution hierarchy to find an appropriate type of source task to transfer from, given a target task.

Suppose that a learner could make some simplifying assumptions about a target game so that it could derive a simpler version of the task. For instance, in a 2-player maze task, the agent could generate a series of randomly constructed mazes, with some approximate model for the opponent’s behavior. The source tasks could be solved very

quickly using DP. When the “real” target mazes are presented, the learner should be able to leverage its source task knowledge to solve the target mazes more quickly than if it had not used transfer.

In this work, we consider two-player games set against a specific, fixed opponent. A game is defined as a set of states, a set of (possibly state-dependent) actions for each player, a reward function for each player, and a transition function that maps a state and the players’ actions to a next state. To define the transfer hierarchy, we consider four characteristics of the game in question:

1. **Is the transition function known?** If the effect of actions are known, the learner may not have to interact with the environment to determine a good policy.
2. **Is the opponent’s policy known?** Can the player anticipate the opponent’s action in any state?
3. **Is the opponent queriable?** Is the opponent willing to answer the question, “What would you do in this state?” If so, we can assume that there is some cost to querying the opponent, but we may jump to different locations in a game tree rather than being forced to play each game from start to end.
4. **Is the opponent deterministic?** A stochastic policy must be sampled repeatedly while a deterministic policy need only be experienced once in each state.²

Given these task characteristics, we construct a hierarchy of solution methods in Figure 1. The method *Transition Learner* concentrates on only learning the effect of moves in the given task since the opponent’s policy is completely known. It is difficult to imagine such a scenario where the opponent’s strategy is defined but the learner does not know the transition model (none of the games commonly played by humans fall into this category). Another less familiar solution method is *Active RL* [14]. In this scenario the learner uses reinforcement learning, but may focus on sections of the state space with the most uncertainty.

In addition to mapping task characteristics to possible solution methods, Figure 1 also defines a *Transfer Hierarchy*. Learners that have more information are able to solve tasks with fewer environmental or opponent interactions. Given a target task with little information, the learner may be able to generate similar tasks but give the learner more information. A central hypothesis for this work is that a learner may train relatively quickly on a simpler source task and then use its learned information to speed up learning

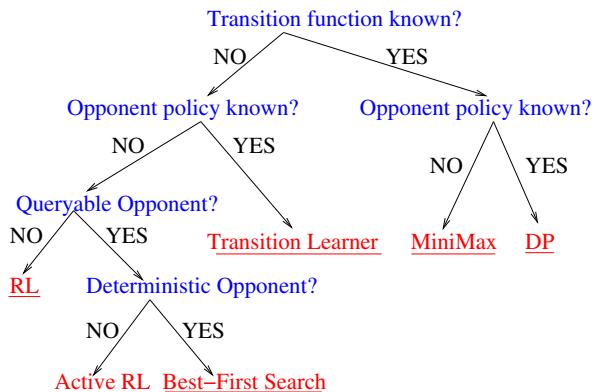


Figure 1. Characteristics of a given task define which solution method is most appropriate. More knowledge leads to solution methods which require fewer interactions with the environment and/or opponent.

²We do not consider non-stationary (e.g., learning) opponents and leave this extension to future work.

the target task which must use a more complex solution method (i.e., one to the left of the source task method which has less information available to the learner). In this paper we empirically test one such pairing: we first learn a series of constructed source tasks via DP to speed up learning a target task via best-first search.

3. Case Study: Transfer in a 2-player Game

To test our transfer hypothesis we utilize the Mummy Maze task, one of many games simulated in General Game Playing. Specifically, we will focus on a target task where the maze is unknown, the opponent’s policy is unknown, and the opponent is both queriable and deterministic. To speed up learning this task using best-first search (as described in Section 3.3.2), we first construct a series of source mazes and a test opponent, which are solvable with DP.

3.1. General Game Playing

Creating programs that can play games at a high level has long been a challenge and benchmark for AI. However, traditional game playing systems are limited in that they play only one particular game. In contrast, the GGP challenge motivates research on creating agents capable of playing many previously unseen games, given only a description of the game’s rules. Since 2005, AAAI has held an annual GGP competition in which agents designed by different researchers compete on a wide variety of games.

In the Game Description Language (GDL) used in the competition, games are modeled as state machines. An agent can derive its legal moves, the next state given the moves of all players, and whether or not it has won by applying resolution theorem proving on the rules of the game combined with the asserted facts for the present state. The language is fairly low-level and is able to describe multiplayer, deterministic, perfect-information games. Syntactically, GDL is a first-order logical description language based on KIF [15]. The next section introduces the game, described in GDL, used in our experimental work.

3.2. Mummy Maze

Mummy Maze³ is a game in which the *explorer* attempts to escape a maze. The opponent *mummy* follows a fixed policy to attempt to stop the explorer. The explorer has 5 deterministic actions: moving one step in each of the four cardinal directions {N, S, E, W} or standing still. The mummy has the same action set, but takes *two* serial actions on each turn. The explorer and mummy alternate moves and neither may transition through walls. The challenge for the explorer is to exploit the mummy’s fixed policy so that he may reach the exit despite the speed disadvantage. The explorer receives a reward of +100 if he reaches the exit and a reward of 0 if the mummy catches the explorer or if the explorer has taken some maximum number of turns (typically 50) without escaping.

A mummy following the *vertical behavior* policy will deterministically move towards the explorer on every move, preferring vertical moves over horizontal moves when both types of move would reduce the players’ distance. Figure 2 shows an example maze, with the solution for the explorer. As the explorer moves to the grid location 1E, the mummy moves North on each move until it moves West and becomes trapped at 1M. Once the mummy is trapped in the cul-de-sac, because it never moves away from the explorer, the explorer may proceed South to the exit. A mummy that follows the *horizon-*

³The .kif file which fully describes the game in GDL may be found at <http://games.stanford.edu/gamemaster/games-mummy/mummymaze1p-horiz.kif>

tal behavior policy prefers to move East or West towards the explorer if possible. Figure 3 demonstrates how the explorer's policy must change to exploit this mummy policy, given the same wall configuration, start state, and goal state. Notice that if the explorer attempted the previous solution path, the mummy would catch the explorer at the cell marked by the red circle.

Mummy Maze is an appropriate choice for this work because we can easily adjust the game definition so that each of the solutions described in the transfer hierarchy is appropriate. For instance, if the explorer is not told where the walls are located, the mummy's policy is unknown, and the mummy is not queriable, RL would be the most appropriate solution strategy. The next section discusses Mummy Maze formulations where DP and best-first search are applicable.

3.3. Mummy Maze Solution Methods

A number of strategies may be employed to solve Mummy Maze, depending on the amount of information the explorer has. In this paper we consider two cases:

1. The transition function is known (i.e. the placement of all the walls in the maze is known) and the opponent's policy is known.
2. The transition function and opponent's policy are unknown, and the opponent is both queriable (i.e. the explorer can ask the mummy, "If I were here and you were there, how would you act?") and deterministic.

In the following sections we explain how Mummy Maze tasks can be solved with dynamic programming, with best-first search, and with transfer from dynamic programming to best-first search.

3.3.1. Dynamic Programming

In its original construction, Mummy Maze is a single player puzzle game, in which the mummy is controlled by a known deterministic policy, specified as part of the environment. Given a task in which the transition function and opponent behavior are deterministic and known, the optimal agent policy may be found by simply enumerating all of the game's states and transitions between them. Such a problem may be solved with dynamic programming.

The dynamic programming algorithm begins by enumerating all states in the game's state set, S . All terminal states are marked as either wins or losses, based on the game's description. Then, all non-terminal states that transition to a terminal state are marked. Any action leading to a win is a win. If all actions lead to a loss, then the originating state is a loss. The iteration continues, marking states that transition to states marked in the previous iteration. One can recover the policy for the solution by simply adding some extra bookkeeping to record the winning transitions between states.

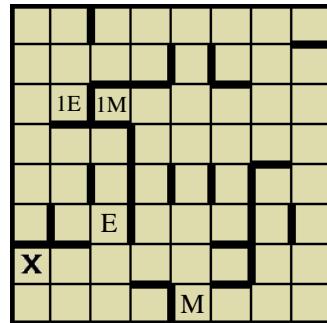


Figure 2. An example solution to a maze with vertical mummy behavior. The explorer moves directly to the 1E space and the mummy is trapped at 1M, allowing the explorer to double back to the exit, denoted by an 'X'.

DP is able to find the optimal solution from all possible initial states for a given goal state. Although the algorithm is very generally applicable, it is only practical on games with reasonably small state spaces. The running time of the algorithm is $O(l^*|A||S|)$ where l^* is the longest solution length, in steps, and A is the set of actions available to the agent. For Mummy Maze on an 8×8 grid, $l^*|A||S| = 50 \times 5 \times 64^2$, which is only roughly one million evaluations.

3.3.2. Best-First Search

In the second variation of Mummy Maze this paper considers, we utilize a search to determine a (possibly sub-optimal) solution to a given maze, if one exists. We utilize a learned heuristic (as specified in the next section) to perform greedy best-first search. If we do not use a heuristic, best-first search reduces to breadth-first search.

We modified the standard best-first search algorithm in a subtle but important way to incorporate domain knowledge. In Mummy Maze, a solution can be broken down into a series of subgoals, each of which trap the mummy and allow the explorer to move to another location. We capture this knowledge by prioritizing a state not solely by its heuristic value, but by the sum of the values of its ancestors. States with high heuristic values are likely subgoals and thus search is guided to explore the children of states that encounter subgoal states along the way.

In the worst case, best-first search must expand the entire game tree. Thus, its running time is proportional to the number of states in the game. Although the computational complexity of the algorithm is less than that of Dynamic Programming, it has a significantly higher constant factor. In each state it must query the opponent for their move, which is an expensive operation.

3.4. Transfer Methodology

In this paper we concentrate on learning a search heuristic for best-first search by solving one or more source tasks with dynamic programming. In this section we discuss how to construct a search heuristic from source task solutions. In the following section, we empirically verify that such a heuristic can speed up search in the target task, even if the source task and target task differ in wall configuration, opponent behavior, size, start state, or goal state.

The main insight for heuristic learning is that rather than learn a heuristic for a *particular* source task, that is one for a particular maze, we learn over a state abstraction. For this task, we chose an abstract representation centered on the Mummy which considers the walls adjacent to it and the direction from the mummy to the explorer. The intuition is as follows. A state where the mummy is in a corner or in a cul-de-sac and the explorer is on the opposite side of the wall is a relatively good position for the explorer. On the other hand, a state where the mummy is in an open area with no walls is less desirable for the explorer because the mummy has a high degree of mobility. In this simple abstraction there is no notion of distance between the mummy and explorer, nor between the explorer and the exit.

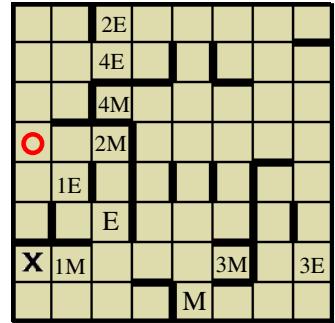


Figure 3. A solution for horizontal mummy behavior. If the explorer attempted the previous solution, the mummy would catch the explorer at the red circle. The explorer must move to squares 1E-4E, trapping the mummy at squares 1M-4M, before exiting.

We use a function GETABSTRACTSTATE which takes the current board configuration and returns the index for the mummy's current abstract state. There are 15 possible wall configurations for the walls directly adjacent to the mummy.⁴ There are 8 possible directions from the mummy to the explorer, which yields 128 possible abstract states, while a standard 8×8 game has 4,096 true states (64 explorer positions \times 64 mummy positions). Although this abstraction is hand coded, we would ideally like to use automated abstractions (e.g., Jong and Stone [16]) in the future.

After solving a source task, the number of wins and losses for each abstract state is tallied. The win percentage $(\frac{\# \text{ wins}}{(\# \text{ wins}) + (\# \text{ losses})})$ for each abstract state is calculated, as well as the average win percentage and the standard deviation. When calculating the heuristic for a state in the target task, we first find the corresponding abstract state. If $\text{winPercentage} \geq \text{aveWinPercentage} + \text{stDev}$ then the heuristic returns +1. If $\text{winPercentage} \leq \text{aveWinPercentage} - \text{stDev}$ then the heuristic returns -1. Otherwise the heuristic returns 0.⁵

4. Case Study: Transfer Results

To test our transfer methodology we perform a number of experiments in which the source and target tasks have different characteristics. In every experiment we construct a set of target tasks and record how many steps the best-first search takes to solve the task with and without transfer. In this setup, the “steps taken” is equivalent to how many times the Explorer must ask the Mummy, “What action would you take in this state?” This is equivalent to the number of connections the Explorer agent must make to the GGP server to query for the opponent’s move. Each target maze is solved 10 times as the best-first search breaks ties randomly. Roughly 25% of the mazes constructed have *no* solution because of the start state and/or wall configuration. Impossible tasks are ignored in the evaluation as no search method could possibly find a solution.

When using transfer, the source task mazes are randomly generated using the same wall-generation algorithm that the target tasks are generated with and thus the mazes in the source and task are drawn from the same distribution of possible mazes. However, because the opponent behavior is different in the two sets of tasks, the distributions of source and target tasks are qualitatively different.⁶ All source task mazes have the same start state and goal state, as depicted in Figure 2. Additionally, all source tasks utilize a horizontal mummy behavior.

4.1. Different Opponent Behavior

All transfer experiments in this paper utilize different mummy behaviors in the source and target tasks. As stated above, the source tasks all use a horizontal behavior Mummy. In the target task the Mummy uses a deterministic mixture of the horizontal and vertical behaviors, denoted *HV-behavior*. HV-behavior specifies that the mummy utilize horizontal behavior if its x and y cell coordinates have the same parity (both are even or both

⁴We do not allow a cell to be surrounded by four walls as it would be unreachable.

⁵Rather than using the *winPercentage* directly as heuristic values, which would tend to explore the states with the highest individual values first, we instead cluster states into three categories: good, neutral, and bad. By doing so, the priority of a state during best-first search is dominated by the number of good states in its history rather than by how good those states are independent of their history. We intend to explore using the continuous version of this heuristic in future work.

⁶If the learner had access to the target task mazes and trained on them, rather than using random mazes for the source task, transfer could be trivially accomplished by memorizing the solution to each maze.

are odd) and act like a vertical mummy if the parity of its x and y cell coordinates are different. Thus the mummy's behavior is deterministic but is qualitatively different from the source task's mummy behavior.

To evaluate experiments in this domain, we define *transfer percentage* to be the ratio between the total number of steps to solve all mazes with and without transfer:

$$100 \times \frac{\sum_{\text{TargetMazes}} (\text{Steps to solve maze with transfer})}{\sum_{\text{TargetMazes}} (\text{Steps to solve maze without transfer})}.$$

To test transfer between source tasks with a horizontal behavior mummy and target tasks with an HV-behavior mummy, we first generate 200 HV-behavior target task mazes. Each is solved 10 times without transfer. Next, 20 horizontal-behavior source tasks are analyzed and the learned heuristic is used to solve each target task 10 times with transfer. We find that the transfer percentage is 73, which means that, on average, using transfer results in a 27% reduction in the number of queries the explorer must make of the GGP server. As may be expected, we found that more difficult target tasks (those requiring relatively more steps to solve) benefited more from transfer on average.

4.2. Different Numbers of Source Tasks

To test the effect of the number of source tasks on transfer, we ran experiments with different numbers of source tasks. The results are reported in Table 1, which shows that even with a very small number of source tasks, transfer can significantly reduce the number of steps needed to solve target tasks.

4.3. Comparison to a Simple Hand-coded Heuristic

In order to better evaluate our learned heuristic, we compared our results to those generated from a simple hand-coded heuristic. If the mummy was able to move in every direction we labeled the state as *bad* and if the mummy was unable to move towards the explorer the state was *good*. Using this metric we observed a transfer percentage of 75, which our learned heuristics either tied or beat (unless fewer than 3 source tasks are used). This suggests that our algorithm is not only able to learn a heuristic autonomously, but that the learned heuristic captures more useful information than a simple hand-coded heuristic.

4.4. Different Target Task Sizes

The 10×10 maze has 10,000 unique states and we expected that our transfer percentage would improve when solving larger target mazes. To test this theory, we again generated 20 8×8 source task mazes, but the target task mazes were 10×10 . We found the resulting transfer percentage to be 66, a slight improvement over 73.

4.5. Different Start State

Up to this point all source and target tasks have been generated such that the mummy and explorer always began at the same coordinates and the exit was in the same location. To evaluate how dependent our method was on the start state, we kept the source task start state fixed but allowed the target task start state to be chosen randomly. We found that the transfer percentage was effectively unchanged, as it now averaged 69 (as compared to 73 when the target tasks' start states were fixed).

# Source Tasks	Transfer Percentage
1	97
2	79
3	74
5	73
10	75
20	73
50	71
100	70
200	71
400	73

Table 1. Results show significant transfer benefit, even with few source tasks.

4.6. Different Start State and Goal State

We next allowed both the start state and the exit to vary in the target tasks. Our setup allowed the exit to be anywhere on the board, which resulted an average transfer percentage 92. We hypothesized this was because our abstract states did not account for relative placement of the exit. Thus our heuristic learned a bias that favored the explorer's mobility towards the Southwest corner of the board in source tasks and when the exit was in a different location, this bias was less helpful (although it was still better than searching without a heuristic). To test this, we then allowed random start states and exit positions in the target tasks, but constrained the exit to be in the Southwest quadrant of the board (thus reducing the number of possible exit locations by a factor of four). With the bias now restored, the resulting transfer percentage was 70. This and other experiments are summarized in Table 2.

Target Task Size	Hand-coded Heuristic?	Target Task Random Start State?	Target Task Random Goal State?	Transfer Percentage
8 × 8	Yes	No	No	75
8 × 8	No	No	No	73
10 × 10	No	No	No	66
8 × 8	No	Yes	No	69
8 × 8	No	Yes	Yes (anywhere)	92
8 × 8	No	Yes	Yes (SW quadrant)	70

Table 2. These results summarize a comparison of searching without transfer to searching after analyzing 20 source tasks. All source tasks are 8 × 8 with H mummy behavior, with fixed start and goal states. Results are averaged over 200 target tasks with HV-behavior mummy behavior.

4.7. Total Time Metric

In order to demonstrate a reduction in the total time we must measure both the time used to solve the source tasks with dynamic programming as well as the time used to solve target tasks solved with best-first search. When using dynamic programming, the solution time is determined by the time to simulate taking an action in the environment and then simulating the opponent's action: $\text{num next states simulated} \times \text{internal next state time}$. When solving a task with best-first search, the learner must query the central GGP server for each next state because the learner does not have the transition function. Furthermore, the GGP server must query the opponent to determine its action for a given state. Solving a target task is dominated by the communication delays and opponent's response time: $\text{num search steps} \times (4 \times \text{communication time} + \text{opponent response time})$.

When connecting to the Stanford Game Manager, the time to compute the next state is about 0.1 seconds, the average communication time with the remote server. Using our own python inference engine, we can simulate an average of 5.51×10^4 next states per second on a 3.4 GHz machine. We assume that the opponent responds in one second (which is much faster than is typical in GGP competitions).

Table 3 shows that the transfer percentage increases for larger target tasks. Additionally, we compare the average number of seconds it takes to solve a target task without transfer (breadth-first search) with the total time needed to solve the source tasks and a target task (best-first search). Such an analysis demonstrates that it is likely when using larger target tasks, or if the opponent takes some time to choose its move, total time can be reduced by using the transfer hierarchy to select source tasks. Transfer requires solving extra source tasks, but the speed-up achieved in the target task may outweigh this initial overhead; the last experiment in Table 3 shows a total time reduction of 19%.

Target Task Size	Ave. Target Task Time (no Transfer)	# Source Tasks	Transfer Percentage	Ave. Total Source Task Time	Ave. Total Time (with Transfer)
8×8	178	20	73	372	502
10×10	400	20	66	372	636
12×12	563	20	47	372	657
12×12	563	10	53	186	461

Table 3. Summary of results comparing searching without transfer to searching after 20 source tasks (times are in seconds). All source tasks are 8×8 with an H-behavior mummy, fixed start and goal states. Results are averaged over 200 target tasks with an HV-behavior mummy.

5. Future and Related Work

In this work the opponents in the source and target tasks have slightly different policies. In preliminary experiments there were not qualitatively different results when using identical policies (horizontal behavior to horizontal behavior) or more dissimilar policies (horizontal behavior to vertical behavior). We speculate that this is because all of these policies are similar enough that transfer can provide a useful heuristic.

One direction for future work would be to consider more dissimilar opponent policies, such as a Mummy that could escape from a cul-de-sac with a certain probability. The abstract state representation could also be enhanced in future work, and ideally would be learned automatically. Likewise, rather than using the transfer hierarchy to selecting a type of source tasks for a given target task, it should be possible to have a TL learner use the hierarchy to *automatically* construct a source task, given a target task. Testing more source and target task pairings would further validate the proposed transfer hierarchy.

While this work has focused on determining how to select a type of source task for a particular target task, we have not addressed what properties a source task should have to best assist learning a target task. For instance, if the transfer hierarchy directs the agent to learn a source task with DP, how can the agent ensure with high probability that the sample tasks it constructs are not misleading? If a generally intelligent agent is to transfer successfully in a fully autonomous setting, it should be able to reliably construct, or select, source tasks that do not cause *negative transfer* so that it avoids the situation where transfer hurts performance, rather than helps.

The main novelty of the experiments in this paper is to present a method for heuristic learning via transfer learning. There is a growing body of work using transfer learning to learn tasks sequentially. For instance, our previous work [10] showed that it was possible to transfer a value function between related reinforcement learning tasks. Other work showed that it is possible to speed up learning between related tasks via advice [11]. GGP tasks have also been used successfully for previous transfer work [17,18].

Prior planning research has demonstrated the possibility of generating state-space abstractions automatically from domain descriptions. These methods may be divided into two forms. In *relaxed models* [19], abstractions are obtained by dropping conditions of actions to make them applicable in more states. A different approach is to generate a *reduced model* [20], in which certain terms are dropped entirely from the problem description. Although neither of these methods could produce our particular abstraction, it is possible that, if applied to Mummy Maze, they may yield different useful abstractions.

6. Conclusion

In this paper we have argued that transfer learning is a critical component of any intelligent system. Transfer learning, particularly in a RL context, has recently been growing in popularity due to empirical successes demonstrating significant speed improvements. We have introduced a transfer hierarchy which assists in selecting a type of source task for transfer, given a specified target task. Additionally, we have demonstrated that trans-

fer between two such tasks types is able to both reduce the target task training time and the total training time in a game drawn from the GGP domain. We view this work as one small, but important, step towards the lofty goal of enabling human-level knowledge transfer, a critical component of any generally intelligent agent.

Acknowledgments

We would like to thank the anonymous reviewers for helpful suggestions. This research was supported in part by DARPA grant HR0011-04-1-0035, NSF CAREER award IIS-0237699, and NSF award EIA-0303609.

References

- [1] Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, May 1996.
- [2] Richard S. Sutton and Andrew G. Barto. *Introduction to Reinforcement Learning*. MIT Press, 1998.
- [3] Andrew Y. Ng, Adam Coates, Mark Diel, Varun Ganapathi, Jamie Schulte, Ben Tse, Eric Berger, and Eric Liang. Inverted autonomous helicopter flight via reinforcement learning. In *International Symposium on Experimental Robotics*, 2004.
- [4] Peter Stone, Richard S. Sutton, and Gregory Kuhlmann. Reinforcement learning for RoboCup-soccer keepaway. *Adaptive Behavior*, 13(3):165–188, 2005.
- [5] Gerald Tesauro. TD-Gammon, a self-teaching backgammon program, achieves master-level play. *Neural Computation*, 6(2):215–219, 1994.
- [6] Sebastian Thrun. Is learning the n -th thing any easier than learning the first? In *Advances in Neural Information Processing Systems*, pages 640–646, 1996.
- [7] E. Thorndike and R. Woodworth. The influence of improvement in one mental function upon the efficiency of other functions. *Psychological Review*, 8:247–261, 1901.
- [8] Mary L. Gick and Keith J. Holyoak. Analogical problem-solving. *Cognitive Psychology*, 12:306–355, 1980.
- [9] Douglas Hofstadter. Analogy as the core of cognition. In D. Gentner, K.J. Holyoak, and B. Kokinov, editors, *The Analogical Mind: Perspectives from Cognitive Science*, pages 499–533. MIT Press, Cambridge, MA, 2001.
- [10] Matthew E. Taylor, Peter Stone, and Yixin Liu. Transfer learning via inter-task mappings for temporal difference learning. *Journal of Machine Learning Research*, 8(1):2125–2167, 2007.
- [11] Lisa Torrey, Trevor Walker, Jude W. Shavlik, and Richard Maclin. Using advice to transfer knowledge acquired in one reinforcement learning task to another. In *The 16th European Conf. on Machine Learning*, 2005.
- [12] Michael Genesereth and Nathaniel Love. General game playing: Overview of the AAAI competition. *AI Magazine*, 26(2), 2005.
- [13] R. E. Bellman. *Dynamic Programming*. Princeton University Press, 1957.
- [14] Lilyana Mihalkova and Raymond Mooney. Using active relocation to aid reinforcement learning. In *Proceedings of the 19th International FLAIRS Conference*, pages 580–585, 2006.
- [15] Michael Genesereth. Knowledge interchange format. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Second Intl. Conference (KR'91)*, 1991.
- [16] Nicholas K. Jong and Peter Stone. State abstraction discovery from irrelevant state variables. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, pages 752–757, August 2005.
- [17] Bikramjit Banerjee and Peter Stone. General game learning using knowledge transfer. In *The 20th International Joint Conference on Artificial Intelligence*, January 2007.
- [18] Gregory Kuhlmann and Peter Stone. Graph-based domain mapping for transfer learning in general games. In *Proceedings of The Eighteenth European Conference on Machine Learning*, September 2007.
- [19] Earl D. Sacerdoti. Planning in a hierarchy of abstraction spaces. *Artificial Intelligence*, 5:115–135, 1974.
- [20] Craig A. Knoblock. Automatically generating abstractions for planning. *Artificial Intelligence*, 68(2):243–302, 1994.

Real Time Machine Deduction and AGI

Peter G. TRIPODES
pgtripodes@cs.com

Abstract: Consistent with the ultimate goals of AGI, we can expect that deductive consequences of large and grammatically varied text bases would not be generated by sequential application of inference rules but would instead be recognized in a single massively parallel pattern matching operation on their semantic structures which executes near instantaneously. We describe an approach to realizing such a capability in graphical terms whereby semantic structures are depicted as certain graphical arrays, and deductive relationships are determined by requisite pattern connections within and among those arrays.

Keywords: Real time deduction, massively parallel pattern matching, semantic structure as graphical array

1.. Realizing Natural Language Deduction Machines

1.1 *Natural Language Deduction Machines*

One goal of AGI is to realize natural language deduction machines, by which I mean machines that can near instantaneously, identify deductive consequences of general text like newspapers, regardless of the number or grammatical complexity of the sentences involved, and do so without the assistance of a human. The difficulties that lie in the path to realizing such machines seem insuperable and the prospect of realizing them to any degree of generality has been virtually abandoned. Yet such machines appear central to the realization of the wider AGI goals involving machine intelligence and reasoning in the handling of natural language¹.

1.2. *Two Problems in Realizing Natural Language Deduction Machines*

Broadly speaking, there are two problems that need to be solved in order to realize such machines. One is the *massively parallel deduction problem*, which is the problem of defining sentence representations and massively parallel deductive operations to operate on them applicable to texts of arbitrary size and grammatical complexity. The second problem is the *conversion problem*, which is the problem of formulating procedures for on-line inter-conversion between sentences and their representations.

1.3. Scope of Paper

In this paper, I suggest an approach to the massively parallel deduction problem which defines sentence representations as certain graphical arrays, and treats deduction as a massively parallel and simultaneously executed pattern matching operation which executes on them, thereby supplanting sequential forms of deductive inferencing [1] [2] [3] [4]. We refer to this proposed operation as *immediate deductive recognition* (IDR). The conversion problem is not treated here inasmuch as it involves complex pragmatic and grammatical issues that are beyond the scope of this paper. Some of these issues are addressed in an unpublished paper by the author entitled “A Theory of Readings [5].”²

1.4. Immediate Deductive Recognition (IDR)

While the theory underlying IDR is essentially that of model theoretic semantics, we describe IDR wholly in graphical terms which much more clearly exhibit the kinds of patterns that are to be matched [6] [7] [8] [9] [10]. Accordingly, we define a *local graph* of a sentence (or of a set of sentences) as a linked array of node-and-arc graphs which collectively graphically depict the denotation of a sentence (or of a set of sentences) relative to a given “permissible” (model theoretic) interpretation which validates it, that is, an interpretation under which that sentence (every sentence in the set) is true. Permissible interpretations are interpretations restricted in certain ways to render local graphs finite and comparable (see section 6.1). And we define the *global graph* of a sentence (or set of sentences) as a linked array of its local graphs relative to all its permissible validating interpretations. IDR executes as a single-step massively parallel operation which simultaneously compares every local graph of a given set of sentences against every local graph of a given sentence and, simultaneously, against every local graph of its negation.³ If every local graph of the given set of sentences is compatible with some local graph of the given sentence, or if every local graph of the given set of sentences is incompatible with every local graph of the negation of that given sentence, the given sentence is thereby determined to be a deductive consequence of that given set of sentences. When this determination is made by machine we will refer to the operation which makes it as *machine IDR*. Compatibility (incompatibility) of a local graph of a set of sentences with a local graph of a given sentence is defined graphically and corresponds to the model theoretic circumstance that there exists (does not exist) a permissible interpretation which validates both the set of sentences and the given sentence and which is such that the local graphs in question respectively depict their denotations relative to that interpretation.

1.5. Human IDR.

The motivation for our approach to machine IDR derives from an apparent human capability, variously noted in the literature, to carry out near-instantaneous deduction (NID) on natural language sentences, particularly when they involve small numbers of simple sentences, and to do so without apparent intervening conscious thought or calculation. *We hypothesize that the cognitive mechanism underlying NID may be a form of pattern matching.*⁴ Accordingly, we will refer to the operation of this hypothesized pattern matching mechanism in humans as *human IDR*.

1.6. Hypothesized Underlying Cognitive Mechanism of Human IDR⁵.

The underlying mechanism of human IDR may be akin to an individual's near-instantaneously recognizing the face of a friend from a photograph by matching (remembered) selected structural characteristics of the face of the friend (such as the set of the eyes or shape of the nose) against those observed in the photograph. That is, an individual's near-instantaneously recognizing simple deductive consequences of given sentences may be carried out by matching selected structural characteristics of the given sentences against selected structural characteristics of those deductive consequence. We will argue below that these structural characteristics may be intuitively perceived patterns of semantic connections among the underlying logical morphemes of those sentences.

1.7. Common Examples of NID

Examples of near instantaneous deduction (NID) appear quite common, and would include cases such as near instantaneously comprehending - without specific practice or familiarization with the specific sentences involved – that the sentences "John loves Mary" and "Mary is loved by John" necessarily follow from each other, and that the sentence "John is dating a waitress" necessarily follows from the sentence, "John is dating many waitresses," as well as from the pair of sentences, "John is dating Mary" and "Mary is a waitress." Instances of NID, while common, appear to be limited for most individuals to small sets of simple sentences, such as the sentences in these examples. This is not to say that humans could not make deductive determinations on larger sets of sentences and of greater complexity, but only that they would not be able (in most cases) to do so without conscious thought and calculation, and certainly could not do so near instantaneously. Near instantaneous deductions of these simple kinds have been widely noted by other researchers, under various names, such as "immediate inference." [11] The difference between the usual treatment of NID and the account given here is that we hypothesize that the underlying mechanism of NID is a form of massively parallel pattern recognition which we refer to as human IDR⁵. The question is how would such a mechanism work? We examine this next.

2. How is Human IDR Possible?

2.1. The Logic of the Language May Be "Hardwired" in the Brain

Human IDR may have its roots in some sort of neural representation of what might be called "the logic of the language," that is, in structures within the brain that represent the semantic structures of logical morphemes and their interconnections which determine deductive relationships [10]. We would regard these structures as specific to certain languages or language groups and hypothesize that an individual's understanding of their logical function develops in the course of his or her "learning" a particular language. The logical morphemes in the above (English) examples would be those expressed there by the word-strings (i.e., morphs), "many," "is," (in the sense of both predication and identity), "es," "ing," "ed," "a," "and," and "by." (Logical morphemes are to be distinguished from so-called non-logical or "lexical" morphemes such as those expressed in the above examples by the word-strings, "John," "Mary,"

and "waitress.") The fact that human IDR is apparently limited to simple sentences may be due to inherent ambiguities in complex sentences deriving from multiple possible functional roles for the logical morphemes occurring in them which obscure their deductive consequences. Simple sentences tend to be less ambiguous in this regard. And the fact that human IDR is apparently limited to small sets of premise sentences may be due to neuro-physiological limits in representing the myriad interconnections among logical morphemes occurring in large numbers of sentences. Machines would, of course, suffer no such limitations.

2.2. Memory: In the Brain and Machines

I use the term "memory" in the sense of language-based (or declarative) memory, and intend it to apply equally to the brain and machines. By language-based memory I mean the store of information held in the brain or in a machine which has, by one means or other, been "entered into it" in the form of sentences. In the case of the brain, these would be sentences which one has either read or heard. In the case of machines, these would be sentences entered into machine memory by any of the customary sentence-input means for machines.

2.3. A Massive Parallelism Assumption Underlying Human IDR

It seems reasonable to assume that the brain executes IDR by a massively parallel mechanism of some kind that is, by a mechanism which executes all subtasks involved in carrying out IDR in parallel and, moreover, executes them globally and simultaneously across the whole of memory. Some version of this assumption is widely shared by neuroscientists and others as necessary to explain how the brain accomplishes certain reasoning tasks so rapidly [1] [2] [3] [10].

2.4. How Memory Is Represented Is Key

The way that sentences are represented in memory wholly conditions the manner in which and the extent to which their deductive consequences can be accessed by massively parallel means. In order to facilitate human IDR, therefore, memory would need to be represented in the brain in such a way that the structure of its deductive consequences is reflected in its own structure to a degree sufficient to permit the massively parallel recognition of at least the simpler of those deductive consequences. And this requires, in turn, that the structure of memory be "global" in the sense that all connections among the components of memory relevant to deduction be included in its representation. These requirements strongly suggest that that representation should be "semantic" rather than "syntactic," that is, that it should represent the semantic structures of sentences rather than their syntactic forms: first, because semantic structures are better preserved under deduction than are syntactic forms; and second, because connections among memory components relevant to deduction are better depicted as semantic structures than as syntactic forms. When memory is represented syntactically, its deductive consequences cannot be directly recognized from that representation, but can only be generated from it and into forms which look very different from those they were derived from. On the other hand, semantic structures of memory can be defined in such a way that their structure is preserved under deductive

inference. We next inquire into the kind of structures of memory would expedite human IDR.

3. Back to the Brain: The Wiring Hypothesis⁵

3.1. Is The Brain "Wired" For IDR?

The apparent capability in humans to carry out deductions on simple sentences near instantaneously and which appears to require no conscious thought or calculation when doing so suggests that the brain may be "wired" for it. Let us suppose that there are neural configurations in the brain that physically encode sentence information as memory. We can refer to these configurations as "sentence representations" (ignoring for the moment whether these representations are more appropriately of sentence forms or of sentence meanings). By the hypothesis that the brain is "wired" for IDR I mean that there may exist pathways in the brain inter-linking stored sentence representations in such a way as to facilitate the immediate recognition of their collective deductive consequences. These pathways, i.e., "wiring", may develop (through an evolutionarily conditioned disposition to do so) in the course of learning a language. Their development may consist in growing new pathways or in reinforcing the capacity of existing pathways to transmit signals, and would progress as the language and the inferences framed in it are practiced. One might regard this hypothesized process as a physical encoding of the "logic of the language" within the brain.

3.2. How Might The Brain Be Wired For IDR? A Proposal

Assuming that the brain is "wired for IDR" in the above sense, what sort of wiring design would be most efficient for its execution? Putting the question teleologically, what would be the most efficient wiring design for the brain to have evolved in order to maximally facilitate the immediate recognition of simple deductive consequences from memory? ("Efficiency", for the brain - as for machines - means compact storage and rapid execution.) To give this question a more determinate form: Let us suppose that a given set X of simple sentences is represented in memory as a pattern X^* of electro-chemical signals, and that some possible (simple) deductive consequence Y of X (being considered) has its negation not-Y represented in memory also as a pattern, say $[not-Y]^*$, of electro-chemical signals. Let us suppose further that the brain would seek to determine whether $[not-Y]^*$ is inconsistent with X^* in the most efficient way possible. Our question then becomes: How might the structures of X^* and $[not-Y]^*$ be related in order to provide the most efficient mechanism for determining that these structures were incompatible, and hence that Y was a deductive consequence of X? We note that it would not be efficient to have X^* sequentially generate patterns in search of some that were inconsistent with $[not-Y]^*$, that is, by sequentially generating a contradiction from X^* and $[not-Y]^*$, for such a "generation procedure" would appear to take much more time than the brain apparently devotes to extract deductive inferences "essentially instantaneously." *The most efficient mechanism would thus appear to be one which required no more storage than that already devoted to memory - to represent X^* - and required now to represent $[not-Y]^*$ as well.*

3.3. A "Most Efficient" Mechanism Would Be Semantic

The requirement that a deductive recognition mechanism be efficient entails that no additional structures be introduced than those already constituting memory and, therefore, that the deductive consequences of memory must already be present and accessible within the structure of memory. We had earlier stated (Section 2.4) that a semantic representation of memory for IDR would tend to be superior to syntactic representations in this regard. However, not all proposed semantic memory structures are equal in this regard: some tend to exhibit their deductive consequences to a greater degree than others. In the following sections we will describe a notion of semantic structure for memory that is specifically designed to be maximally preserved under deductive inference.

4. Back to Machines

4.1. Analogue of Machine IDR in Elementary Algebra

4.1.1. Properties of Algebraic Graphs

Machine IDR for natural language generalizes three familiar properties of elementary classroom algebra of the plane. *The first property of algebraic graphs* is that individual solutions of equations, inequalities, and systems of equations and inequalities can be graphically depicted as points on the plane, and their solution sets can be graphically depicted as figures, i.e., as “graphs” on the plane (in the usual sense). *The second property of algebraic graphs* is that a given equation or inequality is a deductive consequence of a given system of equations and inequalities if and only if every point graphically depicting a solution of the system is either identical with some point graphically depicting a solution of the given equation or inequality or is distinct from every point graphically depicting a solution of the negation of that equation or inequality. *The third property of algebraic graphs* is that in those cases where a machine can make the determination whether given points on the plane are distinct in a massively parallel point-to-point comparison operation, the machine can thereby simultaneously make the determination that the given equation or inequality is or is not a deductive consequence of the system⁴.

4.1.2. Generalizing Algebraic Terminology to Natural Language

In generalizing to machine IDR for natural language deduction we generalize “equation or inequality” to sentence; “system of equations and inequalities” to set of sentences; “solution of an equation or inequality” to “circumstance under which that equation or inequality is true,” “solution of a system of equations and inequalities” to “circumstance under which all sentences of the set are true,” “solution set” to “set of such circumstances,” “point” to “local graph,” “graph” (as of an equation, inequality, or system) to global graph; and the relation of “distinctness among points” to the relation of “graphical incompatibility among local graphs.”

4.1.3. Generalizing the Three Properties of Algebraic Graphs to Natural Language

The first property of algebraic graphs generalizes to natural language as follows: A circumstance under which a given sentence (or every sentence in a given set of sentences) is true can be graphically depicted as an array of local graphs, and the set of all such circumstances can be graphically depicted as a linked array of local graphs called a global graph. *The second property of algebraic graphs* generalizes to natural language as follows: a given sentence is a deductive consequence of a given set of sentences if and only if every local graph of the set is either consistent with some local graph of the given sentence or is incompatible with every local graph of its negation. *The third property of algebraic graphs* generalizes to natural language as follows: in those cases where a machine could determine whether all local graphs of a given set of sentences were compatible with some local graph of a given sentence or incompatible with all local graphs of its negation, and do so in a single massively parallel comparison operation, the machine would thereby simultaneously be making the determination that the given sentence was or was not a deductive consequence of the given set of sentences^{4,6}. While the situation is more complex for natural language than for the much simpler language of elementary algebra, it is clear that the underlying idea is the same.

5. Key Syntactic Notions for Natural Language

5.1. Relation-expressions.

A relation-expression is a syntactic representation of a character string which, in a given occurrence, is regarded as denoting (relative to a given interpretation) an m-place relation which holds among given m-tuples of elements of the universe of discourse.

5.2. Thing- expressions

A thing-expression is a syntactic representation of a character string which, in a given occurrence, is regarded as denoting a "thing" (relative to an interpretation), where a "thing" is a set of subsets of the universe of discourse whose structure is determined by its governing determiners such as "all," "some," "many," etc.

5.3. Sentences

A sentence is a syntactic representation of a character string which, in a given occurrence, is regarded as denoting an "event" or "state of affairs," and is composed of (i.e., formalized as): (1) an m-place relation- expression r^m , together with (2) m thing-expressions a_1, \dots, a_m , each of which denotes a set of sets of elements of the domain of discourse, which elements stand in the relation denoted by r^m .

6. Key Semantic Notions for Natural Language

6.1. Interpretations

An interpretation is a function which assigns a set or individual to every meaningful expression, including individual sentences and sets of sentences, which are referred to as *the local denotation of that expression relative to that interpretation*. We restrict interpretations to *permissible* ones, that is, to interpretations that assign the same finite set to every thing expression and the same function to every non-logical modifier; so that the only way two permissible interpretations could differ would be in the structure of the relation they assign to the same relation expression. The reason for this restriction is to make semantic and graphical structures comparable and computationally coherent. An expression is constant if all permissible interpretations assign the same denotation to that expression; otherwise that expression is said to be variable.

6.2. Set Theoretic Criterion for Deducibility

Every sentence or set of sentences, relative to a given interpretation, has a translation into set theory which expresses – in set theoretic terms - the relationship which must hold among the local denotations of its component expressions – relative to that interpretation - in order for that sentence or all sentences in the set of sentences to be true under that interpretation. This translation into set theory is referred to as the truth condition of that sentence or set of sentences relative to that interpretation. The local denotation of a sentence relative to an interpretation under which it is true is the set of local denotations of its variable component expressions. The local denotation of a set of sentences relative to an interpretation under which all its member sentences are true is the set of local denotations of its variable component expressions. The global denotation of a sentence is the set of its local denotations relative to interpretations under which it is true, and the global denotation of a set of sentences is the set of its local denotations relative to interpretations under which all its member sentences are true. Local denotations of sentences and of sets of sentences are compatible if those denotations are relative to the same interpretation under which they are true, and are incompatible otherwise. A given sentence is a deductive consequence of a given set of sentences if it is true under every interpretation under which all sentences of the set are true [12]. This is equivalent to the following: A given sentence is a deductive consequence of a given set of sentences if every local denotation of the set relative to some interpretation under which its member sentences are true is incompatible with every local denotation of the negation of the given sentence relative to some interpretation under which it is true.

6.3 Graphical Criterion for Deducibility

A local graph of a sentence or set of sentences relative to an interpretation under which it is true is a graphical entity which represents its local denotation relative to that interpretation in the sense that that graphical entity and the local denotation it represents is, in principle, inter-retrievable, and that the set theoretic relationships into which denotations enter are equivalently expressible as graphical relationships among the local graphs which represent them. (In the algebraic case, local denotations are

solutions and local graphs that depict them are points on the plane.) The global graph of a sentence or set of sentences is a connected array of its local graphs, and depicts its global denotation. (In the algebraic case, global graphs are ordinary Cartesian graphs on the plane that depict solutions sets of equations, inequalities, and systems). Two local graphs of sentences or of sets of sentences are compatible if they depict denotations that are compatible, and are incompatible if they depict denotations that are incompatible. (In the algebraic case, points are compatible if and only if they are identical.) A given sentence is a deductive consequence of a given set of sentences if every local graph of the given set of sentences is incompatible with every local graph of the negation³ of the given sentence. (In the algebraic case, the fact that compatibility of local denotations implies identity, whereas in the natural language case it does not, has several interesting consequences. One is that while in the algebraic case, the global graph of a system of equations and inequalities is the graphical intersection of the global graphs of its member equations and inequalities, this does not hold in the natural language case. A second consequence is that the following two statements (a) and (b) are equivalent in the algebraic case but in equivalent in the natural language case: (a) every local graph of the given system is incompatible with every local graph of the negation of a given equation or inequality; (b) every local graph of the system is also a local graph of the given equation or inequality.) In order to avoid attaching any sort of labels to graphs to indicate the semantic interconnections among their components, we use special arcs to join graphical components which represent the same denotation relative to a given interpretation. Accordingly, we refer to graphs which are joined in this way as “linked graphs.”

7. Semantic Representation of Sentences

7.1. Positive and Negative Relational Profiles

If a is a thing-expression, let $\{a\}$ be the result of deleting all determiners from a . Let f be an interpretation, let $r^m(a_1, \dots, a_m)$ be a sentence, and let $CP(r^m(a_1, \dots, a_m))$ be the Cartesian product $f(\{a_1\}) \times \dots \times f(\{a_m\})$. Finally, let $f(r^m)^c$ be the complement of the relation $f(r^m)$. Then we define the *positive relational profile of $r^m(a_1, \dots, a_m)$ under f* , which we write as $POS(f(r^m(a_1, \dots, a_m)))$, to be the intersection of the set $f(r^m)$ with $CP(r^m(a_1, \dots, a_m))$, and we define the *negative relational profile of $r^m(a_1, \dots, a_m)$ under f* , which we write as $NEG(f[r^m(a_1, \dots, a_m)])$, to be the intersection of the set $f(r^m)^c$ with $CP(r^m(a_1, \dots, a_m))$.

7.2. Chain Functions and Traces:

Let f be an interpretation, and let $f(r^m)$, $f(a_1)$, ..., $f(a_m)$, be denotations of r^m , a_1, \dots, a_m , respectively. We define a chain function through the sequence $(f(a_1), \dots, f(a_m))$ as a function g which assigns, for every $1 \leq i \leq m-1$, and for every y belonging to any of the member sets of $f(a_i)$, a set $g(i,y)$ belonging to one of the sets in $f(a_{i+1})$. Let g be a chain function through the sequence $(f(a_1), \dots, f(a_m))$. Then we define the trace of g through $(f(a_1), \dots, f(a_m))$ as the set: $\{(z_1, \dots, z_m) \in D^m // \text{for some } x_1 \in B_1, z_1 \in x_1, \text{ and } z_2 \in g(1, z_1), \text{ and } z_3 \in g(2, z_2) \text{ and } \dots \text{ and } z_m \in g(m-1, z_{m-1})\}$. There are in general many possible chain functions through the sequence $(f(a_1), \dots, f(a_m))$ of thing expressions of $r^m(a_1, \dots, a_m)$ relative to f , but there is exactly one chain function whose trace is identical with the

positive relational profile of $r^m(a_1, \dots, a_m)$ relative to f if $r^m(a_1, \dots, a_m)$ is true under f , and no such chain function if $r^m(a_1, \dots, a_m)$ fails to be true under f .

7.3 Denotations of Sentences and Their Graphical Representations

We define the denotation $f(r^m(a_1, \dots, a_m))$ relative to the interpretation f , in symbols, $\text{Den}(f(r^m(a_1, \dots, a_m)))$, as the set: $\{ \{ \langle f(r^m), v \rangle // v \in \text{POS}(f[r^m(a_1, \dots, a_m)]) \} \} \cup \{ \langle f(r^m)^c, v \rangle // v \in \text{NEG } f(r^m(a_1, \dots, a_m)) \} \}$, if there is a chain function g through the sequence $(f(a_1), \dots, f(a_m))$ such that the trace of g through $(f(a_1), \dots, f(a_m))$ is identical with $\text{POS}(f(r^m(a_1, \dots, a_m)))$; and is $\{\emptyset\}$, otherwise. The set $\text{Den}(f(r^m(a_1, \dots, a_m)))$ can be completely graphically represented as a network of nodes and connecting arcs, where the un-negated connecting arcs graphically represent the relation and each m -tuple of the nodes they connect graphically represents m elements of the domain which stand in that relation, and where negated connecting arcs graphically represent the complement of the relation and each m -tuple of the nodes they connect graphically represent m elements of the domain which fail to stand in that relation.^{7,8}

8. Graphical Representation of Sentences

8.1. Nodes, Arcs, and Paths

Local graphs are composed of two types of basic graphical elements: nodes and arcs. Nodes represent elements of the underlying domain of discourse and arcs represent relations on those elements.

8.1.1. Simple and Compound Arcs

Simple arcs are arcs that join at most two entities. There are three types of simple arcs: (i) arrows, which joins at most two nodes and which can be barred or unbarred; an unbarred arrow represents a relation whose relata are elements of the underlying domain of discourse, and considered in the order indicated by the direction of the arrow, and a barred arrow represents the complement of that relation; (ii) dotted lines, which represent the identity relation when unbarred, and the non-identity relation when barred, and which join either two points to represent that they represent the same or different elements of the underlying domain of discourse, or two arrows to represent that they represent the same relation if both are barred or both are unbarred, or to represent complementary relations if one arrow is unbarred and the other is barred; and (iii) dashes, which represent the logical conjunction of the entities represented by the graphical entities it joins. *Compound Arcs* are arcs formed by joining two or more simple arcs with a graphical unit called a “brace,” and represent many-place relations composed of those simple arcs. An arc that is not a constituent of a compound arc is said to be major.

8.1.2. Dot Paths, Arrow Paths, and Mixed Paths

A path is a simple or compound arc taken together with nodes it joins. If the constituent arcs of the path are all dotted lines, the path is called a dot path. If the constituent arcs of the path are all arrows, the path is called an arrow path, if the constituent arcs of the path are both dotted lines and arrows; the path is called a mixed path. An arc which is a constituent of a path is said to be a major constituent of that path if it is not itself a constituent of another constituent of that path. A path is said to be barred or unbarred according as its major constituent is barred or unbarred. A path represents that the elements of the domain of discourse respectively represented by the nodes of the path stand in the relation represented by the path. Arrow paths and mixed paths represent lexical relations, the place number of which corresponds to the number of nodes in the path. A single node placed at the origin or terminus of an arrow signifies respectively that the element represented by the node is in the domain or range of the relation represented by the arrow.

8.1.3. Similarity and Identity Linked Paths

Two paths are *similarity linked* if they differ only in the placement of bars on one or more of their constituent arcs, and all corresponding nodes and arcs joined by dotted lines. Two paths are *identity linked* if graphical depictions of the same denotation are joined by dotted lines.

8.2. Local and Global Graphs

8.2.1. Local Graphs

A *local graph* is an array of similarity linked paths. For definiteness, that array is organized in such a way that the corresponding nodes are displayed in vertical columns. We refer to a column of corresponding nodes as a node bank, and to the nth such column in a local graph as the nth node bank of the local graph.

8.2.2. Global Graphs as Similarity Linked Local Graphs

Two local graphs are *similarity linked* if they differ only in one or more constituent paths which are similarity linked, and all corresponding nodes and arcs are joined by dotted lines, and a *Global Graph* is an array of local graphs which are pair wise similarity linked.

Endnotes

1. By “central to realizing wider AGI goals,” I mean that without this sort of deductive capability, real time machine execution of other types of reasoning, such as those involved in inductive, probabilistic, or pre-suppositional inference, could probably not be achieved at the level envisioned in AGI. The reason is that deduction structures meaning interconnections within and among expressions which occur in these other types of reasoning as well as functions as a limiting special case for various of them (e.g., inductive and probabilistic inference). As an example where deduction functions as a limiting case for probabilistic reasoning, we note that deductive inference can be generalized to a certain kind of probabilistic inference whereby the probability that a given sentence is true given that all sentences in a given text base are true can be near-instantaneously calculated as the “weighted proportion” of local graphs of the text base which are incompatible with all local graphs of the negation of the given sentence. This same mechanism can be used to show the degree of consistency of the entire text base.

2. A sentence of a natural language is regarded here as a character string to which a specific syntactic and semantic structure, called a “reading,” has been assigned, and which varies generally with the context of utterance. A character string, relative to a given reading, has a unique graphical representation which captures those aspects of its meaning that determine its deductive interconnections with other character strings relative to their readings. We do not address the very difficult problem of developing effective procedures for determining readings of given character strings for given contexts of utterance; however, we have developed, though not included in this paper, effective procedures for obtaining suitable graphical representations of given character strings relative to given readings.
3. A sentence has as many negations as it has readings. When we refer to the negation of a given sentence we are assuming a specific reading of that sentence, and its negation is that sentence whose main relation is interpreted as the set-theoretical complement of the main relation of the given sentence.
4. Regarding the question of whether deduction is too complex or otherwise unsuitable to be treated as pattern matching, the point of this paper is that it can be so treated provided that the representations used are structured to enable it.
5. While the empirical evidence for the proposed mechanism for human IDR is at this point primarily anecdotal, it is still reasonable to speculate regarding the sorts of deductive mechanisms which could be consistent at least with that evidence, and which could serve as starting point for future empirical studies. The virtue of the proposed mechanism is that it is fully explicit and applies to a wide range of sentences.
6. While other sorts of inference mechanisms based on pattern matching have been proposed for inference forms other than deduction, such as abduction or analogy, these appear to be limited to a far narrower range of cases than those which we claim are addressed by the deductive mechanism proposed here.
7. The reason for defining the denotation of a sentence as a singleton-singleton set is that we want to have sentences qualify also as thing-expressions, and the denotations of thing expressions are always sets of sets of elements of the universe of discourse.
8. There are, in general, many possible chain functions on the sequence $(f(a_1), \dots, f(a_m))$ of thing expressions of $r^m(a_1, \dots, a_m)$ relative to f , but there is exactly one chain function whose trace is identical with the positive relational profile of $r^m(a_1, \dots, a_m)$ relative to f if $r^m(a_1, \dots, a_m)$ is true under f , and there is no such chain function if $r^m(a_1, \dots, a_m)$ fails to be true under f .

References

- [1] Hinton, G. E., J. L. McClelland, and D. E. Rumelhart. (1986) Distributed Representations. In D. E. Rumelhart and J. L. McClelland, Eds., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Cambridge, MA: MIT Press,
- [2] Shastri, L., & Ajjanagadde, V. (1993). From simple association to systematic reasoning: A connectionist representation of rules, variables and dynamic bindings using temporal synchrony. *Behavioral and Brain Sciences*, **16**, 417 – 494.
- [3] Shastri, Lokendra (1999) Advances in *Shruti* – A neurally motivated model of relational knowledge representation and rapid inference using temporal synchrony. *Applied Intelligence*, **11**: 79 – 108.
- [4] Johnson-Laird, P. N. & Byrne, R. M. J. (1991) *Deduction*. Hillsdale, NJ: Erlbaum
- [5] Allwein, G, Barwise, J (1996) *Logical Reasoning with Diagrams*, OUP.
Evans, J. St.B.T, Newstead, S. E., & Byrne, R.M.J. 1993. *The Psychology of Deduction*. Hove, UK: Lawrence Erlbaum Associates Ltd
- [6] Gardner, M. , (1958) *Logic Machines and Diagrams*, New York, NY, McGraw-Hill
- [7] Hammer, E. M. (1995) *Logic and Visual Information*, CSLI Publications.
- [8] Peirce, C. S. (1897-1906) Manuscripts on existential graphs. In *Collected Papers of Charles Sanders Peirce*, edited by Arthur W. Burks, vol. 4, (pp 320-410), Harvard University Press.
- [9] Shin, S-J (1994) *The Logical Status of Diagrams*. CUP.
- [10] Sowa, J. F. (1984) Conceptual Structures: Information processing in mind and machine.
- [11] Hummel, J. E. & Choplin, J. M. (2000) Toward an integrated theory of reflexive and reflective reasoning. In L. R. Gleitman & A. K. Joshi (Eds.), *Proceedings of the Twenty-Second Annual Conference of the Cognitive Science Society* (pp.232-237). Mahwah, NJ: Erlbaum.
- [12] Tarski, A. 1956. The Concept of Truth in Formalized Languages. In A. Tarski. *Logic, semantics, and metamathematics: Papers from 1923-1928*. Oxford: Oxford University Press
- [13] Tripodes, P. G. (Unpublished Manuscript) A Theory of Readings.

Text Disambiguation by Educable AI System

Alexander VOSKRESENSKIJ
College MESI, Moscow, Russian Federation

Abstract. The structure of possible text understanding system is discussed. To store concepts and knowledge system uses multilayer ontology based on pragmatic memory model. The way of knowledge searching for system education is described. Main aim of project is development of system for translating text into sign language phrases.

Keywords. Natural language understanding, sign language, concept, memory, ontology, knowledge management, education

Introduction

This paper dwells on the problems related to the machine translation of Russian texts into the Russian sign language (RSL) used by deaf people. The project is now at its early stage implementation and it can be discussed only as a theoretical conception.

While translating from a verbal language into a sign language there comes a problem of disambiguation which in a certain sense different from the problem which we encounter while translating from one spoken language to another. What is unequivocally perceived in a verbal language may have several meanings in a sign language. This difference should be specifically expressed during translation. It is true not only RSL, but also of any other sign language, like Danish, for example [1].

Moreover, the same word in different contexts gets different senses. Therefore, fixed, universal systems of semantic coding are impossible. Certainly, there are words or word combinations whose values are fixed, such as some kinds of terminology (technical, medical, linguistic, etc.) or idioms. But it is frequent that word meanings, especially in informal conversation, are probabilistic. That is, the word does not have a fixed value. Instead, it is the index on some semantic field whose borders can be, in turn, indistinct.

For Nalimov [2], the "... words, on which our culture is based, do not and cannot have an atomistic meaning. It has become possible and even necessary to consider words as possessing fuzzy semantic fields over which the probabilistic distribution function is constructed and to consider people as probabilistic receivers" (p. 56).

Due to facts that any verbal language consists of millions words in contrast to approximately 6 (or maybe 8) thousands of sign language gestures, and sign language grammar differs from verbal language grammar, translating text to signs is possible only by a system which understands input text. The system must to select sign (or chain of signs) representing concept nearer by sense to translated words concept. Thus, text understanding in this case means narration using another set of words.

One of the preconditions of this work is the model of perception and processing of the information [3], partially explaining the probabilistic character of conceptual values. This model tries to find explanations of the distinctions between the cognitive abilities of the deaf and the hearing.

To store information about objects described in text a dynamic multilayer ontology is used which structure is based on said model. The general structure of estimated text understanding system is discussed.

The technique to select concepts with the same (or like) sense was offered which may be useful not only for described task but for searching for texts containing new knowledge for the subject domain also. An experiment has been planned and carried out on documents from the Internet to check the offered approach [4]. As far as we know, this is the first case of the application of design of experiment (DoE) [5] methods in linguistic research.

1. Memory: Pragmatic Approach

In many studies human memory is represented as having short-term memory and long-term memory parts and short-term memory is used for information input and output (see, for example, [6, 7]).

But this model is not effective with pragmatic point of view: we don't know a priori a value of new information so short-term memory can't be used as a filter of useful information entering long-term memory. Moreover, any experiment concerned with presentation of some information to human and subsequent answering interacts with output channel only because processing of information input by human mind is not available for direct experimental studying.

Based on observations presented in [3] possible simplified diagram of memory is shown in Figure 1.

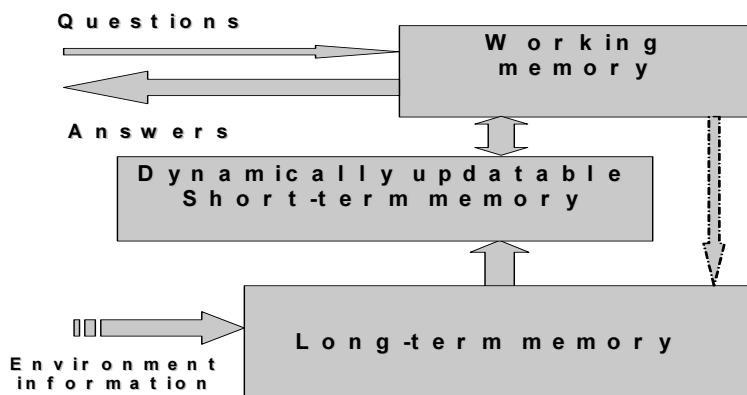


Figure 1. Diagram of possible memory structure

This model of memory may be described in the following way:

- The information from an environment enters directly in long-term memory where it is stored constantly.
- The information is presented by concepts and relationships between them forming knowledge, one of relationship parameters is the level of activation, which rate changes with time in due course from a maximum value down to a minimum value ($R_{min} > 0$), simulating process of forgetting.
- In case the concept and relations are confirmed by input information, level of these relations activation is raised but no exceeds maximum value ($R \leq R_{max}$). Levels of activation for tied concepts are rising too (probably, to a lesser degree).
- The knowledge which level of activation exceeds the certain value is come in the short-term memory (which is updated periodically). Because information entering from environment (in general case it may include visual, audio and other kinds of information, for example, books, films and so on) effects activation levels of concepts in long-term memory, content of short-time memory may be changed every cycle of short-memory updating. So content of short-time memory (or consciousness) is probabilistic to some degree.
- Working memory usually cooperates with short-term memory. The reference from working memory in long-term memory is possible during persevering remembrance of forgotten.
- This model allows reproducing processes of "attenuation" and "replacement" of knowledge, "changes of outlook" as a result of receipt of new knowledge.

The parameters of relationship between two concepts include causation, describing one concept as a cause and other as an effect. Because the model allows for concepts (in long-term memory) more than one relationship having different rates of activation level it's possible to change cause and effect (changing direction of relationship between concepts or selecting another pair of concepts) in short-term memory. It's a way for model's "changes of outlook" or domain tuning. So this model uses term logic rather than predicate logic. In this way it is closer to these AI systems as NARS or Novamente [8].

One of education results is new knowledge stored in mind. But some kind of knowledge is more valuable for subject than other; in many cases (but not always) new knowledge is more valuable than old one. If some kind of knowledge is confirmed more often than other, the former is more valuable than latter. Natural regulator of this phenomenon is forgetting. In our model changing of relationship activation level emulates the process of forgetting.

2. Multilayer Ontology

In our system there is a dynamic multilayer ontology proposed as said memory model.

In brief it may be described in the following way (see Figure 2):

- Each concept is represented by hyper graph which nodes are attributes of current realization of concept.
- Rib of hyper graph singles out concrete realization of concept from the general set of knowledge. The rib has numerical parameter of activation, and also

attributes of time, a place and the action which was a reason for the given rib formation.

- The parameter of activation of a rib has maximal value at formation, gradually decreases in due course.
- The top layer realizing a current “picture of the world” is formed by hyper graph, having ribs with activation level which exceeds the certain value.

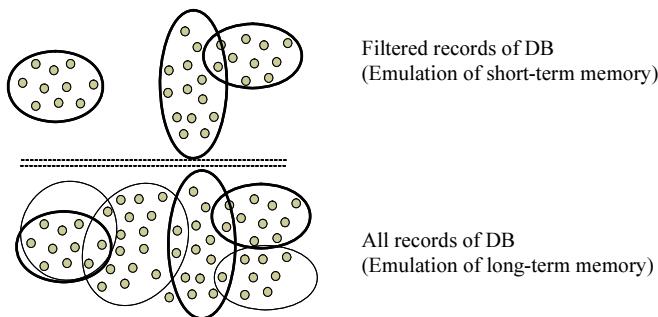


Figure 2. Formation of current “picture of world” by filtering an ontology content.

Realization of such structure is possible by means of a relational DB. The top layer (layers) is virtual and formed by the records of a DB filtered by certain rules.

3. Text Understanding System

On the basis of comparison of processes of verbal and sign languages communications it is shown [9], that the system of understanding of the text, besides the means providing linguistic analysis of the text, should include the block tracing changes of characteristics of objects described in the text (spatial position, the sizes, shape, age, etc.) and storing values of these characteristics in a binding to time of the text and to astronomical time. This way allows creating descriptions of situations at any moments of time.

But description of text objects will be not full without the explanation of their attitudes to each other, for example, friendly or hostile. This information can be absent in the given text, but can be derived from external sources of the information (for the human it can be the knowledge received during training, from the literature or other sources).

Originally language was developed from communication system composed of the vital signals for the human (danger, food, etc.) [10]. So it is possible to assume, that at reading any text the person builds system of the attitudes both to the text as a whole, and to objects described in it (these attitudes can change from negative up to positive, including neutral).

If to add to functions specified above the block which function is to mark objects of the text by various levels of attitudes "good" and "bad", this block will carry out (to some extent) functions of such mental phenomenon as individual "I" (in I.G. Fichte's interpretation).

The marking of objects may be realized on the basis of comparison and analogies to the objects marked at preliminary training of system.

The structure of such system is shown on Figure 3.

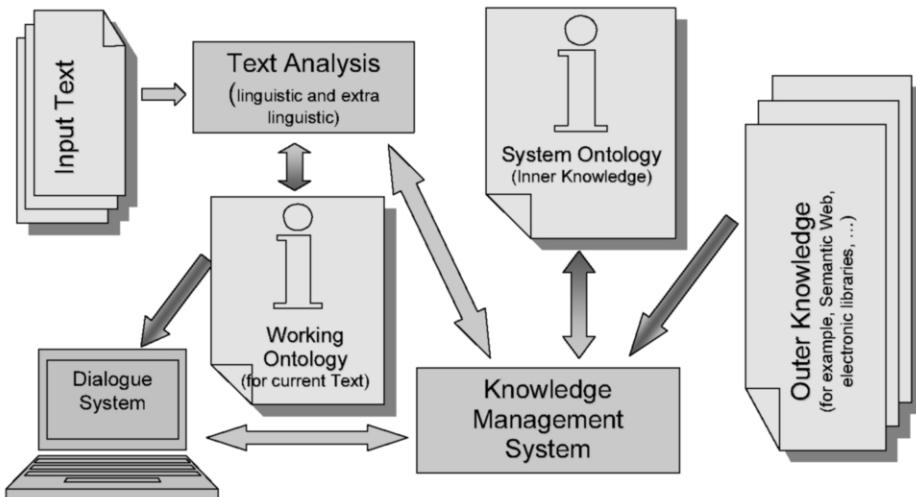


Figure 3. Structure of Text Understanding System

Input text at first is construed using linguistic methods (morphologic and syntactic). Defined objects (main actors, props and locations) with their attributes are loading into working ontology. If it possible objects information are enriched using knowledge from system ontology of computer and/or outer knowledge.

The system uses foregoing multilayer ontology. Such ontology can store the information on changes with time of the objects described in the text (and their moving), moreover, it can be used for short-term forecasting of topic evolution.

Aforesaid ontology includes objects of two types: concepts, containing descriptions of objects, and relationships which connect (group) concepts. Relationships are not attributes of concepts – they are independent objects having own set of attributes which includes references to concepts with which the given relationship cooperates.

Relationship attributes include time of an establishment of the given relationship (for example, astronomical time if relationship concerns to concepts of world around, or time of text if relationship concerns to concepts of a literary work), level of activation (or weight) of relationship and so forth.

Eventually the weight of relationship decreases, that simulates process of forgetting. The minimal weight of relationship is greater than zero, i.e. once the established relationship of the bottom level is never broken off.

At each updating of relationship (simultaneous supervision of the concepts grouped by given relationship) the weight of this relationship increases (but does not exceed the certain maximal value, identical, most likely, for all relationship). Simultaneously under the certain law the weight of each relationship connected with these concepts (in direct connection, and through other concepts) increases.

Interaction of the program agent who is carrying out semantic text processing, occurs only with top level of dynamic multilayer ontology.

The top level of dynamic multilayer ontology includes relationships with weight which exceeds the certain threshold level, and concepts which are tied by these relationships. Updating of top level of dynamic multilayer ontology is made periodically. This process models reception of new knowledge and forgetting old one (maybe like process is a reason for some brain rhythms?).

At inclusion in attributes of relationship the parameter defining a subject domain, fast tuning of ontology top level to the certain subject domain is possible.

Content of ontology top level has probabilistic character, i.e. system perception of the input text rather strongly depends on the information acting from external sources (not including the given input text). Respective alterations of word meanings (semantic fields) are close to the probabilistic model of language described in [2].

4. Education of System

One of definition in Google said that education is the gradual process of acquiring knowledge. If new concepts and/or relationships are contained in input text they will be loaded from ontology of text into bottom layer of computer ontology. Knowledge managing system will try to find new relationships between concepts using deduction rule Eq. (1):

$$\{A \rightarrow B, B \rightarrow C\} \mid A \rightarrow C. \quad (1)$$

Taking into consideration that creating new relationships or updating of existing ones changes activation levels of these and corresponding relationships this process may result to changes in top level of ontology representing learning outcome.

Of course, in the beginning stage when computer ontology is empty, process of self education is impossible. So the preliminary loading of computer ontology is needed. Conditions when self education starts are not known as yet.

For system education may be used outer knowledge sources also.

Ability of AI system for self education is very important, because, for example, we can't teach system all rules of text semantic interpretation: we don't know the whole list of these rules as yet; moreover, it may be changing due to probabilistic nature of language. Ability of AI system constructs these rules using term logic open a way to solve the problem.

5. Search of New Knowledge

To accomplish procedures of self educating the system must have means for access to new knowledge. One of the ways is to search it in Internet. An experiment has been planned and carried out on documents from the Internet to check the offered approach. As far as we know, this is the first case of the application of design of experiment (DoE) methods in linguistic research. Detail description of this experiment was published in Russian [4] and later in English [11]. Here shorten description of this experiment is presented.

Composition of queries, loading these queries into search engine, and results treatment was accomplished ‘by hands’. Of course, these procedures may be automated, but in this case it was inefficient.

5.1. Semantic Search Technique

Dekang Lin has stated that: “*Two different words are likely to have similar meanings if they occur in identical local contexts.*” [12]. This statement was used in cited work for a choice of the correct word meaning in the dictionary, using explanatory texts for entries.

However, while translating the text into sign language, use of a local context appears insufficient. For example, the word “fence” is transferred into different signs of RSL depending on whether (inside or outside of the territory surrounded by a fence) a subject is located. This information can be found only in the general context of the text.

Let’s expand on D. Lin’s statement and formulate it as follows: *Two words or phrases can have a similar value if they co-occur in identical (or similar) contexts.* We have removed the restriction of a local context.

In this case a word or phrase whose value is defined by comparison of the context surrounding it and the context of a cue word or phrase is only a “plug”, reserving a place in a context. This “plug” can be excluded if the measures providing corresponding wildcard in a context are stipulated.

The way to search for similar documents semantically on the basis of the comparison of their lexical vectors is known. But in the task of searching, including all or the most significant components of a lexical vector of the document, the documents concerning different yet close subject domains will be eliminated.

The method presented for semantic search uses the replacement of the most significant component of a query vector by the wildcard. In this case, as a result of search, there are documents that will be retrieved belonging to domains that are different, but close, to the subject domain. These describe distinct concepts corresponding to a component excluded from the lexical vector and some of these concepts are previously unknown to the user.

In Google, the role of wildcard is handled by the symbol “*” where the number of asterisks should not be less than the number of excluded words. Russian search engine Yandex¹ has the special operator for these purposes².

¹ www.yandex.ru

² http://www.yandex.ru/ya_detail.html

5.2. Design of Experiment

A technique for checking Internet documents has been selected in connection with their availability, their huge quantity and the variety of their content.

Two measures of the query results were considered: the whole number of documents found as a result of the query (Y_1) and the number of relevant documents contained in the first 50 selected documents (Y_2). The choice of the second measure was defined by the difficulty for the experimenters to estimate the relevance of all received documents. Thus it is supposed that the search engine returns the majority of relevant documents in the top of the list of references.

The search engine Yandex was used.

The experiment looked at research on the influence of three factors:

A – The fragment of text passed (is substituted by wildcard). The excluded element of the text shall be designated as X .

B – Word order in the query.

C – Morphological forms of words.

We vary for all three factors. Factorial, the experimental plan corresponds [5] to 2^3 including eight combinations of query variants (Table 1), where maximum factor value is designated as “+”, minimum – as “-“.

Table 1. The order of queries to the search engine

Trial	A	B	C
(1)	-	-	-
a	+	-	-
b	-	+	-
ab	+	+	-
c	-	-	+
ac	+	-	+
bc	-	+	+
abc	+	+	+

The queries were transformed to the syntax required by Yandex. When the maximum of the **A** factor is shown it means that it was set by the operator “/(+2 +5)”, meaning that the query terms are separated from one to four other words in the resulting text.

This provides a template in which words and expressions can fall synonymous with the excluded element X . We call this template the “semantic trap”.

For an estimate of model adequacy, each of the experiments included two independent returns. The first return was obtained by the query: «географические атласы стран Европы» (“geographical atlases of the countries of Europe”) where X = «атласы» (“atlases”). The second was: «поиск новых знаний в интернете» (“search for new knowledge on the Internet”) where X = «знания» (“knowledge”).

Thematically, these two phrases are not connected with each other.

The randomization of results was reached through the casual order of the query task. Results are shown (Table 2).

It is apparent (Table 2), that the results are rather changeable, especially for Y_1 . There is an observed stratification of results between returns. The reason for this can be

the influence of unconsidered factors, such as the number of query terms. To eliminate this factor, a normalization function was applied to the return for each query (Eq. 2):

$$\hat{Y}_{ji} = (Y_{ji} - Y_{imin}) / (Y_{imax} - Y_{imin}) \quad (2)$$

Where: j = trial number; i = number of return; Y_{jimin} = minimum value of the return function for the i^{th} result; Y_{imax} = maximum value of the return function for the i^{th} result.

Table 2. The experiment results

Trial	Return	Result Y_1	\hat{Y}_1	Result Y_2	\hat{Y}_2
(1)	1	48607206	0,975772	23	0,437500
	2	279573078	0,999740	15	0,789474
a	1	47980297	0,963187	21	0,375000
	2	279567901	0,999722	15	0,789474
b	1	18	0,000000	9	0,000000
	2	0	0,000000	0	0,000000
ab	1	1098	0,000022	33	0,750000
	2	1188	0,000004	19	1,000000
c	1	49814093	1,000000	13	0,125000
	2	279645655	1,000000	17	0,894737
ac	1	48880091	0,981250	12	0,093750
	2	14676	0,000052	15	0,789474
bc	1	147	0,000003	41	1,000000
	2	17	0,000000	5	0,263158
abc	1	17907	0,000359	30	0,656250
	2	254	0,000001	6	0,315789

Mathematical models of results were constructed and after excluding of insignificant factors a mathematical indicator of query relevance was obtained (Eq. 3 and 4):

$$\hat{Y}_{1t}=0,433-0,865B \quad (3)$$

$$\hat{Y}_{2t}=0,517+0,157A+0,207AB-0,264AC+0,122BC-0,246ABC \quad (4)$$

Here, symbols B and C designate the contribution of syntax (word order) and morphology respectively. It is characteristic that there was only an interaction of these factors is significant.

This shows that in semantic search it is still necessary to consider both the morphology of query words and syntax as well. It is obvious, that for this purpose, it is necessary to analyze the texts of documents but not indexes of contemporary search engines, because search engine indexes do not keep punctuation which is essential for text processing.

5.3. Discussion of Experimental Results

Assumption laid in the basis of the experiment was justified. For example, for second query «поиск новых знаний в интернете» ("search for new knowledge on the

Internet") in place of excluded word «*знанияй*» ("knowledge") these words and word-combinations were obtained: «*такантильных авторов*» ("gifted authors"), «*каналов коммуникаций*» ("channels of communication"), «*информации*» ("information"), «*тематических ресурсов*» ("thematic resources"), «*православных страниц*» ("Orthodox pages") and so on.

It is obvious that content of the query is too small for keep retrieved results in one domain. On the other hand retrieved results offers permissible forms of initial phrase overpatching for other domains. It is open a way for automated smooth widening of knowledge sphere of trainable system not limited by initial dictionary of system.

Other experiment results pertinent to linguistic and data extracting was discussed in [4 and 11].

6. Process of Text Disambiguation

Due to Frege's composition principle sentence meaning is a function of meanings of the sentence parts and way of these parts combining. So to text understanding there is important to find accurate meanings for words and word-combinations of the text.

In our task there are several main types of disambiguation needed to be resolved:

- a) Words polysemy and ambiguity. In most cases it can be resolved by means of morphological and syntactic analysis. If proper meaning will not be founded on this stage, all possible meanings will be loaded in bottom layer of text ontology for posterior disambiguation by semantic value and created relationships with other concepts of text.
- b) Identification of concepts and proper names to aggregate their attributes in bottom layer of ontology. This task is general for automated annotation and referring systems. Example of like task is shown in [13].
- c) Identification of pronoun reference. Be guided by [6, 14] the search of noun or proper name to which a pronoun makes reference is limited, as a rule, by one paragraph. After resolve the reference, attributes defined by pronoun (and its reference) are aggregated in bottom layer of text ontology.
- d) Identification of references in compound sentence. In this case an area of reference resolving is limited by this compound sentence. After resolve the reference, attributes of concepts are aggregated in bottom layer of text ontology.
- e) Resolving of anaphora, ellipsis, incomplete sentence will be accomplished using information extracted from previous part of text and stored in text ontology.

Our context based approach proposes to use surrounding context of a text fragment for definition of this fragment meaning and for to resolve disambiguation (homonymy, homograph, anaphora, and so on).

The frequency analysis techniques are used for selection of words or word-combinations having most probability of accordance with given semantic field (with high occurrence frequency) and discriminate analysis techniques are used for selection of words or word-combinations having most significance in given domain for semantic field delimiting (its occurrence frequency is low – an analogue of "slang" method [15] used, for example, for author's identification).

7. Possible Application

One of the distant learning problems is laboriousness of learning courses creating. To meet specific, immediate, and unique learner's needs for personal goals and tasks IBM Corporation developed the solution named Dynamic Learning [16].

One of this solution features is ability for dynamically create a custom "course" out of modular learning objects. To prepare this learning objects specific procedures are used, but operation of metadata editing is executed manually (Figure 4³).

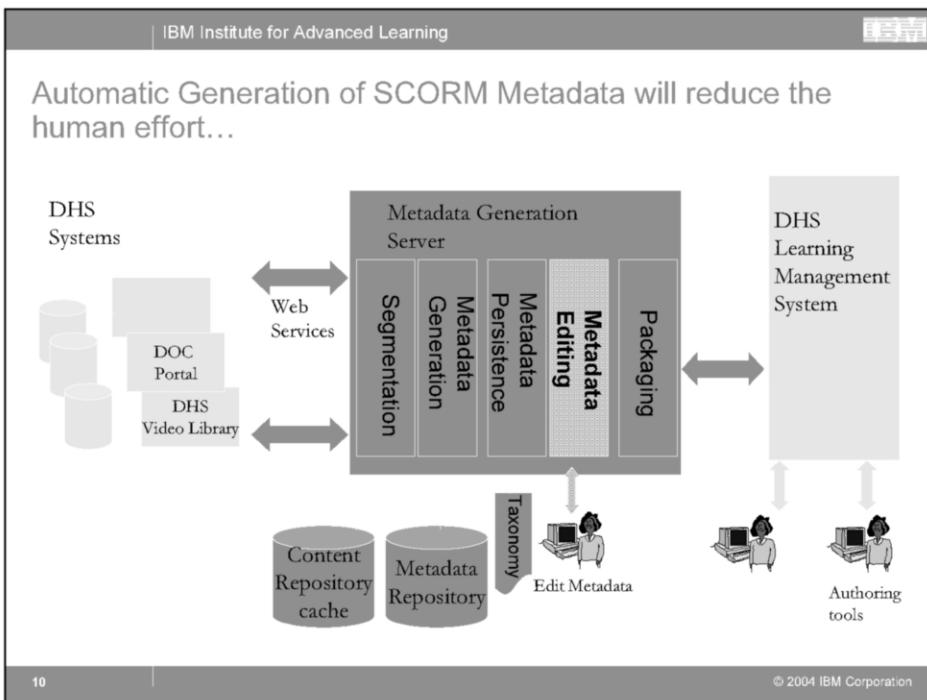


Figure 4. Automatic Generation of SCORM Metadata will reduce the human effort...

Creation of text understanding system can to solve a problem of metadata editing. Furthermore it can to make easier the creating of integrated learning courses by giving access to learning modules related to different taxonomy classes (different educating area) but having common features.

Acknowledgements

This work is partially funded by Human Capital Foundation (<http://hcfoundation.ru>).

³ This slide from [16] is cited with permission of Dr. Y. Ravin.

References

- [1] Jette Kristoffersen, Thomas Troelsgaard, Janne Boye Niemelä, Bo Haardell. How to describe mouth patterns in the sign language dictionary. *Theoretical Issues in Sign Language Research 9*. Florianópolis, December 06 to 09 2006, Universidade Federal de Santa Catarina. Florianópolis, SC Brazil. (<http://www.tislr9.ufsc.br/index.htm>).
- [2] Vasily Nalimov. *Realms of the Unconscious; The Enchanted Frontier*. ISI Press, 1982, 320 p.
- [3] Alexander Voskresenskij. Forgetting as a Factor for Knowledge Forming. *Proceedings of First Russian Internet-conference on Cognitive Science, 2004* (In Russian: Материалы Первой Российской Интернет-конференции по когнитивной науке — М., УМК «Психология», 2004, С. 150 – 155.)
- [4] A. Voskresenskij and G. Khakhalin. Composition of queries to search engine for knowledge retrieval from Internet. *Proceedings of International Conference on Computing Linguistics and Intellectual Technologies “Dialogue'2005”* (In Russian: Воскресенский А.Л., Хахалин Г.К. Формирование запросов к поисковой машине для извлечения знаний из Интернета. // Компьютерная лингвистика и интеллектуальные технологии: Труды международной конференции "Диалог'2005" — М.: Наука, 2005. С. 86 – 91.; <http://www.dialog-21.ru>)
- [5] D.C. Montgomery. *Design and Analysis of Experiments* (4th ed.), New York: Wiley, 1997.
- [6] A. Fenk, G. Fenk-Oczlon. Information processing limitations and linguistic structure. *Proceedings of the Second Biennal Conference on Cognitive Science*. Saint-Petersburg, Russia, 2006.
- [7] Carlo Geraci, Marta Gozzi, Costanza Papagno, Carlo Cecchetto. Short term memory and sign languages. Reduced resources and full languages. *Theoretical Issues in Sign Language Research 9*. Florianópolis, December 06 to 09 2006, Universidade Federal de Santa Catarina, Florianópolis, SC Brazil. (<http://www.tislr9.ufsc.br/index.htm>).
- [8] Artificial General Intelligence /B. Goertzel, C. Pennachin (eds). — Springer, 2007.
- [9] A. Voskresenskij and G. Khakhalin.. About model of NL-text understanding. *Proceedings of the Second Biennal Conference on Cognitive Science*. Saint-Petersburg, Russia, 2006. (In Russian: Воскресенский А.Л., Хахалин Г.К. О модели понимания ЕЯ-текста. // Вторая международная конференция по когнитивной науке: Тезисы докладов: В 2 т. Санкт-Петербург, 9 – 13 июня 2006 г. — СПб.: Филологический факультет СПбГУ, 2006. — Т. 1, С. 238 – 239).
- [10] V.G. Red'ko. Problem of Cognitive Evolution Modeling. *Proceedings of First Russian Internet-conference on Cognitive Science, 2004*. (In Russian: Ред'ко В.Г. Задача моделирования когнитивной эволюции. // Материалы Первой Российской Интернет-конференции по когнитивной науке / Под ред. А.Н. Гусева, В.Д. Соловьева — М., УМК «Психология», 2004. С. 14 – 28).
- [11] A. Voskresenskij and G. Khakhalin. Semantic Search Engine in a Multimedia Russian Sign Language Dictionary. *Proceedings of XII International Conference “Speech and Computer” SPECOM'2007*. October 15 – 18, 2007. Moscow, Russia. Volume 2. pp. 739 – 744.
- [12] D. Lin. Using syntactic dependency as local context to resolve word sense ambiguity. *Proceedings of the 35th annual meeting on Association for Computational Linguistics*. Madrid, Spain, 1997.
- [13] Z. Kazi and Y. Ravin. Who's Who? Identifying Concepts and Entities across Multiple Documents. *Proceedings of the 33rd Hawaii International Conference on System Sciences – 2000*. (0-7695-0493-0/00).
- [14] A.A. Kibrik. Reference and Work Memory: On Interaction of Linguistic with Psychology and Cognitive Science. (In Russian: Кибрик А.А. Референция и рабочая память: о взаимодействии лингвистики с психологией и когнитивной наукой. // Материалы Первой Российской Интернет-конференции по когнитивной науке / Под ред. А.Н. Гусева, В.Д. Соловьева — М., УМК «Психология», 2004. С. 29 – 43.)
- [15] S. Khaitun. *Sciencesmetrics*. (In Russian: Хайтун С.Д. Наукометрия. Состояние и перспективы. — М.: Наука, 1983).
- [16] Y. Ravin. Innovation in Learning from IBM: Examples from the Institute of Advanced Learning. // *Learning-on-Demand European Meeting*, London, England, 6 December 2004.

What Do You Mean by “AI”?

Pei WANG

Temple University, Philadelphia, USA

pei.wang@temple.edu

Abstract. Many problems in AI study can be traced back to the confusion of different research goals. In this paper, five typical ways to define AI are clarified, analyzed, and compared. It is argued that though they are all legitimate research goals, they lead the research to very different directions, and most of them have trouble to give AI a proper identity. Finally, a working definition of AI is proposed, which has important advantages over the alternatives.

Keywords. Intelligence, working definition, research paradigm

1. The Problem of Defining “Intelligence”

A research project should have a clearly specified research goal; a research field should consist of research projects with related research goals. Though these requirements sound self-evident, Artificial Intelligence (AI) seems to be an exception, where people not only disagree on what is the best solution to the problem (which is usual in any branch of science and engineering), but also on what the problem is (which is unusual, at least given the extent of the disagreement). As evidence of this situation, at the 50th anniversary of the field, the AAAI Presidential Address still asked the question “what really is AI and what is intelligence about?” [1].

It is well known that people have different understandings to what “intelligence”, or “AI”, means. However, this issue has not been explored to the extent it deserves, mainly due to two widely spread opinions:

- There is a *natural* definition of the term “intelligence”, while the different understandings are just different aspects of the same notion, and the various AI schools are exploring different trails to the same summit, or working on different parts of the same whole.
- Like most terms in natural languages, the term “intelligence” cannot be defined, therefore people can keep whatever understanding they like about it, as far as their research produce useful results.

Though these two opinions take opposite positions on whether intelligence can be defined, they lead to the same attitude toward this issue, that is, they see the discussion on the definition of “intelligence” as a waste of time.

The aim of this paper is to show that both above opinions are wrong. In the following, I will clarify various understandings of AI, analyze their relations, and evaluate their potentials. I will then argue for the necessity and possibility of giving “intelligence” a proper “working definition” for the need of AI research. For the field as a whole, multiple working definitions exist, and it will remain to be the case in the near future. Even so, to clearly understand their difference is still very important.

For the emerging field of “Artificial General Intelligence” (AGI), this discussion has special importance. Since AGI treats “intelligence” as a whole [2], a project in this field will be inevitably guided and judged by its working definition of intelligence.

This paper follows my previous discussions on this topic [3, 4], and addresses the topic in a more accurate and comprehensive manner. As discussed in [3], a “working definition” of a term, like “intelligence”, is a definition to be used as the goal of a research project. To carry out the research consistently and efficiently, every researcher needs to select or establish such a working definition. At the early stage of a field, no working definition can be agreed by every researcher, but it does not mean that no one is better than another. A working definition should be *sharp, simple, faithful* to the original term, and *fruitful* in guiding the research. Since these requirements usually conflict with one another, the final choice is typically a compromise and tradeoff among various considerations. For instance, a working definition often can neither fully agree with the everyday usage of the term (which is fuzzy and vague), nor be fully formalized (which will be too far away from the everyday usage).

Though people have different opinions on how to accurately define AI, on a more general level they do agree on what this field is about. Human beings differ from animals and machines significantly in their *mental* ability, which is commonly called “intelligence”, and AI is the attempt to reproduce this ability in computer systems. This vague consensus sets important constraints on how AI should be defined:

- Since the best example of “intelligence” is the human mind, AI should be defined as *identical to human intelligence* in certain sense. At the early stage of research, this “identical to” (a matter of yes/no) can be relaxed to “similar to” (a matter of degree), and the progress of research can be indicated by the increased degree of similarity.
- Since AI is an attempt to duplicate human intelligence, not to completely duplicate a human being, an AI system is different from a person in certain other aspects. Otherwise the research would be aimed at “artificial person”, rather than intelligent computer. Therefore, it is not enough to say that an AI is similar to human without saying where the similarity is, since it cannot be in every aspect.

To make the analysis and comparison precise, in this paper human beings and computer systems are all specified as “agents” that “receive percepts from the environment and perform actions” [5]. At a given moment t , the full history of an agent can be represented as a triple $\langle P, S, A \rangle$, where $P = \langle p_0, \dots, p_t \rangle$ is the sequence of *percepts*, $A = \langle a_0, \dots, a_t \rangle$ is the sequence of *actions*, and $S = \langle s_0, \dots, s_t \rangle$ is the sequence of *internal states* the system has gone through. When a typical human mind is represented as $H = \langle P^H, S^H, A^H \rangle$, and a typical intelligent computer as $C = \langle P^C, S^C, A^C \rangle$, a working definition of AI corresponds to a definition of *similarity* between C and H , when the two are described at a certain level of abstraction.

Since this discussion is about the qualitative difference among AI working definitions, not about the quantitative difference in intelligence among systems, in the following no attempt will be made to establish a numerical measurement of this similarity. Instead, the focus will be on identifying the *factors* that are relevant to this similarity. To simplify the discussion, it is assumed that two sequences (of percepts, actions, or states) are similar as far as their corresponding components are similar to each other, and that the similarity between two percepts, two actions, and two states can be meaningfully evaluated in certain way.

Limited by length, this paper concentrates on the major types of working definitions of AI, without analyzing every proposed definition in detail. For the same reason, the paper will not address how to build an AI system according to a given working definition.

2. Typical Ways to Define AI

Following the distinction introduced in [3], typical ways to define AI are divided into five types, each of which evaluates the similarity between C and H by *structure*, *behavior*, *capability*, *function*, and *principle*, respectively. They are discussed in the following, one by one.

(1) By Structure

Since the best known instance of intelligence is produced by the human brain, it is natural to assume that AI can be achieved by building a brain-like *structure*, consisting of massive neuron-like processing units working in parallel.

This idea has been further developed in various forms, such as Connection Machine [6] and Artificial Neural Networks [7]. More recent brain-oriented AGI works include those by Hawkins [8] and de Garis [9].

Due to the complexity of the human brain and its fundamental difference from computer hardware, none of these projects plans to be faithful to the brain structure in all the details. Instead, they only take the brain as the source of inspirations, and the resulting systems approximate to the brain at a certain level and scope of description.

Even so, many people inside and outside the field of AI still believe that accurate “brain modeling” will provide the ultimate solution to AI, when it is allowed by our knowledge of the human brain and the available computer technology. According to this opinion, “the ultimate goals of AI and neuroscience are quite similar” [10].

I will call this type of definition “Structure-AI”, since it requires the structural similarity between an AI system and the human brain. In the agent framework, it means that C is similar to H in the sense that

$$\langle P^C, S^C, A^C \rangle \approx \langle P^H, S^H, A^H \rangle$$

that is, the two have similar streams of percepts and actions, as well as similar state transforming sequences, due to their similar internal structure. According to this understanding of AI, even though it is impossible to accurately duplicate the brain structure in the near future, we should try to move to that goal as close as possible, and the distance to it can be used to evaluate the research results.

(2) By Behavior

Since intelligence seems to be more about the human mind than the human brain, many people believe that it is better to concentrate on the system’s *behavior* when evaluating its intelligence. The best known idea in this category is the Turing Test [11]. Though Turing proposed his test only as a sufficient condition, not a necessary condition, for intelligence, it nevertheless is taken by many people as the definition of AI [12, 13].

A representative approach towards AGI, following this path, can be found in Newell's discussion of the Soar project [14], which was presented both as an AI system and a model of human psychology. According to this opinion, AI is identified with “cognitive modeling”, where the computer-produced results are evaluated by comparisons with psychological data produced by human subjects. In its later years, Soar has been moving away from this strong psychological orientation, so at the current time a better example for this category is ACT-R [15], though it is not proposed as an AI model, but a psychological model.

Another example of this understanding of AI can be found in the field of “chatbot”, where the intelligence of a system is evaluated according to how much it “talks like a human”, such as in the Loebner Prize Competition [16].

I will call this type of definition “Behavior-AI”, since it requires the behavioral similarity between an AI system and the human mind. In the agent framework, it means that C is similar to H in the sense that

$$\langle P^C, A^C \rangle \approx \langle P^H, A^H \rangle$$

that is, the two should have similar streams of percepts and actions. Here the two systems are treated as “black box”, whose internal structure and state do not matter. Of course, the AI system may be similar to a human mind only after a certain period of training, and that can be accepted in the above representation by setting the starting moment of the percepts and actions at the completion of the training.

(3) By Capability

For people whose interest in AI mainly comes from its potential practical applications, the intelligence of a system should be indicated by its *capability* of solving hard problems [17]. After all, this is how we usually judge the intelligence of a person. Furthermore, the progress of a research field will eventually be evaluated according to the usefulness of its results.

Partly because of such considerations, the earliest practical problems studied by AI were typical intellectual activities like theorem proving and game playing — if a person can solve these problems, we call the person “intelligent”; therefore, if a computer can do the same, then we may have to call the computer “intelligent”, too. Driven by similar motivations, a large number of application-oriented AI projects are “expert systems” in various domains — experts are intelligent, so if a computer can solve a problem that only an expert can, the computer must be intelligent, too.

Especially, a computer is often considered as intelligent if it solves a problem that previously could only be solved by human beings, but no computers. Consequently, AI becomes an expanding frontier of computer application.

The biggest AI achievements so far, according to this understanding, include Deep Blue, the chess-playing system that defeated the world champion, and Stanley, the self-driven vehicle that finished a 132-mile trek in 7 hours.

I will call this type of definition “Capability-AI”, since it requires an AI system to have human capability of practical problem solving. In the agent framework, it means that C is similar to H in the sense that there are moments i and j such that

$$\langle p_i^C, a_i^C \rangle \approx \langle p_j^H, a_j^H \rangle$$

that is, the action (solution) the computer produces for a percept (problem) is similar to the action produced by a human to a similar percept — to make the discussion simple, here I assume that a single percept can represent the problem, and a single action can represent the solution. Since here what matters is the final solution only, it is irrelevant whether the computer goes through a human-like internal process or produce human-like external behavior beyond this problem-solving process.

In the AGI context, it follows that systems with higher intelligence can solve more and harder problems. A recent form of this idea is Nilsson’s “employment test”: “To pass the employment test, AI programs must be able to perform the jobs ordinarily performed by humans. Progress toward human-level AI could then be measured by the fraction of these jobs that can be acceptably performed by machines.”[18] Among existing AGI projects, a representative one of this type is Cyc, which encodes vast amounts of commonsense knowledge to achieve human-like problem-solving capability [19].

(4) By Function

Since most AI researchers are computer scientists and engineers, they prefer to represent the ability of an agent as some *function* that maps input (percepts) into output (actions), which is how a computer program is specified.

Typical opinions are like “Intelligence is the computational part of the ability to achieve goals in the world” and “What is important for AI is to have algorithms as capable as people at solving problems”, both from McCarthy [20]. A more systematic and influential description came from Marr: “a result in Artificial Intelligence consists of the isolation of a particular information processing problem, the formulation of a computational theory for it, the construction of an algorithm that implements it, and a practical demonstration that the algorithm is successful” [21].

Guided by such opinions, the field of AI is widely seen as consisting of separate cognitive functions, such as searching, reasoning, planning, learning, problem solving, decision making, communicating, perceiving, acting, etc., each having its various computational formulations and algorithmic implementations [5].

I will call this type of definition “Function-AI”, since it requires an AI system to have cognitive functions similar to those observed in humans. In the agent framework, it means that C is similar to H in the sense that there are moments i and j such that

$$a_i^C = f^C(p_i^C), \quad a_j^H = f^H(p_j^H), \quad f^C \approx f^H$$

that is, the function that maps a percept (problem) into an action (solution) in the computer is similar to that of a human. Since here the focus is on the functions, the actual percepts and actions of the two agents do not matter too much.

In the AGI context, such a working definition implies that a system should have many cognitive functions working together. Representative projects moving in this direction include LIDA [22] and Novamente [23].

(5) By Principle

Science always looks for simple and unified explanations of complicated and diverse phenomena. Therefore, it is not a surprise that some AI researchers attempt to identify

the fundamental *principle* by which human intelligence can be explained and reproduced in computer at a general level.

Intuitively, “intelligence” is associated with the ability to get the best solution. However, such a definition would be trivial if it asks the agent to exhaustively evaluate all possible solutions and to select the best among them. To be more realistic, Simon proposed the notion of “Bounded Rationality”, which restricts what the agent can know and do [24]. Russell argued that intelligent agents should have “Bounded Optimality”, the ability to generate maximally successful behavior given the available information and computational resources [25].

Among AGI projects, AIXI [26] and NARS [4] can be seen as different attempts to build AI as some type of rational or optimal system, though they specify rationality in different ways, and make very different assumptions on the environment of the system. AIXI aims at the highest expected reward, under the assumption that the system has sufficient resources and the environment is a Turing Machine. On the other hand, in NARS “intelligence” is defined as “adaptation with insufficient knowledge and resources”, which puts no restriction on the environment, while requiring the system to be finite, real-time, and open.

I will call this type of definition “Principle-AI”, since it requires an AI system to follow similar normative principles as the human mind. In the agent framework, it means that C is similar to H in the sense that

$$A^C = F^C(P^C), \quad A^H = F^H(P^H), \quad F^C \approx F^H$$

that is, the function that maps the whole stream of percepts into the whole stream of actions in the computer is similar to that of a human. Again, here the focus is on the function, not the actual percepts and actions. Here the function is called a “principle”, to stress that it is not just about a single problem and its solution, but about the agent’s life-long history in various situations, when dealing with various types of problems.

3. The Necessity of Distinction

The above five types of working definition all set legitimate research goals, but they are different from each other.

- Structure-AI contributes to the study of the human brain. It also helps to explain how the brain carries out various cognitive activities, but if the research goal is in the behavior, capability, function, or principle of the mind, then to duplicate the brain structure is often not the best way (in terms of simplicity and efficiency), because the brain is formed under biological and evolutionary restrictions largely irrelevant to computers.
- Behavior-AI contributes to the study of human psychology. Very often, “the human way” gives us inspirations on how to use a computer, but it is not the best way to solve a practical problem, or to implement a cognitive function or principle. Also, behavior similarity does not necessarily require structural similarity.
- Capability-AI contributes to various application domains, by solving practical problems there. However, due to the lack of generality of the solutions, this

kind of solution usually contributes little to the study of brain or mind outside the scope of the domain problems.

- Function-AI contributes to computer science, by producing new software (sometimes also hardware) that can carry out various type of computation. However, the best way to implement the required computation is usually not exactly the way such a process is carried out in the human mind/brain complex. Since a cognitive function is generalized over many concrete problems, it is not necessarily the best way to solve each of them. If an agent is equipped with multiple cognitive functions, they are not necessarily designed according to the same principle.
- Principle-AI contributes to the study of information processing in various situations, by exploring the implications of different assumptions. Given the generality of a principle, it cannot explain all the details of the human brain or the human mind, nor does it provide the best way to solve every practical problem. Even though a principle-based system usually does carry out various cognitive functions, they are not necessarily separate processes, each with its own computational formulation and algorithmic implementation.

Therefore, these five trails lead to different summits, rather than to the same one.

If these working definitions of AI all originated from the study of the human brain/mind, how can they end up in different places? It is because each of them corresponds to a different level of description. Roughly speaking, the five types of working definitions in the above list are listed in the order of increasing generality and decreasing specificity, with Structure-AI being the most “human-centric” approach, and Principle-AI the least (though it still comes from an abstraction of human thinking). Each level of description comes with its concepts and vocabulary, which make certain patterns and activities more visible, while ignore other patterns and activities visible in the other levels, either above it or below it. In this context, there is no such a thing as the “true” or “correct” level of description, and each of the five can be used as goals of legitimate scientific research.

To distinguish five types of AI definition does not mean that they are unrelated to each other. It is possible to accept a working definition as the primary goal, and also to achieve some secondary goals at the same time, or to benefit from works aimed at a different goal. For example, when implementing a principle, we may find that the “human way” is very simple and efficient, which also provides good solutions to some real-world problems. However, even in such a situation, it is still necessary to distinguish the primary goal of a research from the additional and secondary results it may produce, because whenever there is a design decision to make, it is the primary goal that matters most.

Even though each of the five types of AI definition is valid, to mix them together in one project is not a good idea. Many current AI projects have no clearly specified research goal, and people working on them often swing between different definitions of intelligence. Such a practice causes inconsistency in the criteria of design and evaluation, though it may accidentally produce some interesting results.

A common mistake is to believe that there is a “true” (“real”, “natural”) meaning of “intelligence” that all AI research projects must obey. Some people think that AI should follow the common usage (i.e., the dictionary definition) of the word “intelligence”. This is not going to work. The meaning of “intelligence” in English (or a similar word in another natural language) was largely formed before AI time, and therefore is mainly about human intelligence, where the descriptions at various levels

(structure, behavior, capability, function, principle, etc.) are unified. On the contrary, for computer systems these aspects become different goals, as discussed above.

For similar reasons, AI cannot simply borrow the definition of “intelligence” from other disciplines, such as psychology or education, though the notion does have a longer history in those fields. This is not only because there are also controversies in those fields about what intelligence is, but also because the notion “intelligence” is mainly used there to stress *the difference among human beings* in cognitive ability. On the contrary, for AI this difference is almost negligible, and the notion is mainly used to stress *the difference between human beings and computer systems*. Also for this reason, it may not be a good idea to use IQ test to judge the ability of AI systems.

Some people argue that “AI is simply what the AI researchers do”. Though a survey of the field provides a valid *descriptive definition* of AI, it is not a valid *working definition*, which should be precise and coherent to guide a research project. Under the common name “AI”, AI researchers are actually doing quite different things, as described previously. Even if there is a majority point of view, it does not necessarily become the “true meaning” of AI that everyone must concur.

It is true that in many science disciplines the basic notions become well defined only after long-term research. However, in those disciplines, at least the *phenomena* to be studied are clearly identified at the beginning, or the disagreements in working definitions of those notions do not make too much difference in the direction of the research. On the contrary, in AI each researcher has to decide *which phenomena* of the human intelligence should be studied and at which level of description. Such a decision is inevitably based on an explicitly or implicitly accepted working definition of intelligence. There is no way to be “definition-neutral”, because otherwise the research would have nowhere to start — a phenomenon is relevant to AI only when the term “AI” has meaning, no matter how vague or poor the meaning is.

Furthermore, the existing working definitions of AI are incompatible with each other, as discussed previously, to the extent that progress toward one may be moving away from another. It is very difficult, if meaningful, to design or evaluate an AI system without considering its research goal first.

The confusion among different definitions is a common root of many controversies in AI. For example, there has been a debate on whether Deep Blue is a success of AI [27, 28]. According to the above analysis, the conclusion should clearly be “yes” if “AI” is taken to mean “Capability-AI”, otherwise the answer should be “not much”, or even “no”. We cannot assume people are talking about the same thing only because they are all using the term “AI”.

4. The Possibility of Comparison

To say there are multiple valid working definitions of intelligence (and therefore, AI) does not mean that they cannot be compared, or that they are equally good.

In [3], four criteria of a good working definition were borrowed from Carnap's work (when he tried to define “probability”) [29]:

- It should have a *sharp* boundary.
- It should be *faithful* to the notion to be clarified.
- It should lead to *fruitful* research.
- It should be as *simple* as possible.

Given their forms as defined previously, the five types of definition are not too different with respect to the requirements of *sharpness* and *simplicity*. Therefore, the following discussion focuses on the other two criteria, *faithfulness* and *fruitfulness*.

As mentioned before, in general it is hard to say which of the five is more faithful to the everyday usage of the word “intelligence”, because each of them captures a different aspect of it. Similarly, each of the five produces important results, and which one is “more fruitful” can only be determined after decades or even longer.

Therefore, instead of trying to decide which working definition is the best *in general*, in the following I will focus on one aspect of this topic: which working definition will give the field “AI” a proper *identity*, which should explain how the field differs from the other fields, as well as elicits the common natures of its subfields.

AI has been suffering from a serious identity crisis for years. Many AI researchers have complained that the field has not got the recognition it deserves, which is sometimes called “The AI Effect” — as soon as a problem is solved, it is no longer considered as a problem for AI anymore [30]. Within the field, fragmentation is also a big problem [1] — each subfield has its own research goal and methods, and to collectively call them “AI” seems only to have a historical reason, that is, they all more or less came out of attempts of making computer “intelligent”, whatever that means.

For AI to be considered as a field of its own, its definition must satisfy the following conditions:

- AI should not be defined in such a *narrow* way that takes human intelligence as the only possible form of intelligence, otherwise AI research would be impossible, by definition.
- AI should not be defined in such a *broad* way that takes all existing computer systems as already having intelligence, otherwise AI research would be unnecessary, also by definition.

Now let us analyze the responsibility of each type of working definition with respect to the identity problem the field AI faces. Especially, how the definition posits AI with respect to human psychology and computer science.

There is no doubt that the best example of “intelligence” is “human intelligence”, and therefore all working definitions attempt to make computer systems “similar” to humans, in various senses, and to various extents. However, Structure-AI and Behavior-AI seem to leave too little space for “non-human intelligence”, so they may be sufficient conditions for “intelligence”, but unlikely to be necessary conditions. If an intelligent system must have human brain structure or produce human cognitive behaviors, then some other possibilities, such as “animal intelligence”, “collective (group) intelligence”, and “extraterrestrial intelligence”, all become impossible *by definition*. It would be similar to defining “vision” by the structure or behavior of the human visual organ. For AI, such a definition will seriously limit our imagination and innovation of novel forms of intelligence. Human intelligence is developed under certain evolutionary and biological restrictions, which are essential for human, but hardly for intelligence in general. After all, “Artificial Intelligence” should not be taken to mean “Artificial Human Intelligence”, since “Intelligence” should be a more general notion than “Human Intelligence”.

On the other hand, Capability-AI and Function-AI seem to allow too many systems to be called “intelligent”. It is not hard to recognize that works under the former is just like what we usually call “computer application”, and the latter, “computer science”, except that the problems or tasks are those that “humans can do or try to do” [27]. Do these definitions give enough reason to distinguish AI from Computer Science (CS)?

Marr's computation-algorithm-implementation analysis of AI [21] can be applied to every problem studied in CS, and so does the following textbook definition: "we define AI as the study of agents that receive percepts from the environment and perform actions" [5]. This consequence is made explicit by the claim of Hayes and Ford that AI and CS are the same thing [31].

If the only difference between AI and CS is that the "AI problems" are historically solved by the human mind, then how about problems like sorting or evaluating arithmetic expression? Some people have taken the position that all programs are intelligent, and their difference in intelligence is just a matter of degree. Such a usage of the concept of "intelligence" is coherent, except that the concept has been trivialized too much. If intelligence is really like this, there is no wonder why AI has got little credit and recognition — if everything developed in the field of AI can be done in CS, and the notion "intelligent agent" has no more content than "agent", or even "system", what difference does it make if we omit the fancy term "intelligence"?

Furthermore, the widely acceptance of Capability-AI and Function-AI as working definitions of AI is responsible for the current fragmentation of AI. Both of them define AI by a *group* (of capabilities and functions, respectively), without demanding much commonality among its members. As a result, AI practitioners usually assume they can, and should, start to work on a single capability or function, which may be integrated to get a general intelligence in the future. Since the best ways to solve practical problems and to carry out formal computations differ greatly from case to case, there is not too much to be learned from each other, even though all of them are called "AI". As far as people continue to define their problems in this way, the fragmentation will continue.

The above analysis leaves us only with Principle-AI. Of course, like the other four types discussed above, Principle-AI is not a single working definition, but a group of them. Different members in the group surely lead to different consequences. Obviously, if the "principle" under consideration is too broad, it will include all computer systems (which will be bad); if it is too narrow, it will exclude all non-human systems (which will be bad, too). Therefore we need something *in between*, that is, a principle that

- (1) is followed by the human mind,
- (2) can be followed by computer systems,
- (3) but are not followed by traditional computer systems.

An example of such a working definition of AI is the one accepted in the NARS project. Briefly speaking, it identifies "intelligence" with "adaptation with insufficient knowledge and resources", which implies that the system is finite, works in real-time, is open to novel tasks, and learns from experience [3, 4]. There are many reasons to believe that the human mind is such a system. The practice of NARS shows that it is possible to develop a computer system following this principle. Finally, traditional computer systems do not follow this principle. Therefore, such a working definition satisfies the previous requirements. Though NARS can be studied in different aspects, the system cannot be divided into independent functions or capabilities, since the components of the system tangle with one another closely, so cannot be treated in isolation. The notion of "intelligence" is not an optional label in this research, since it does introduce ideas not available in computer science or cognitive psychology. Designed in this way, NARS has shown many interesting properties [4], though to discuss them is far beyond the scope of this paper.

To prefer the NARS definition of AI does not mean that it can replace the others for all purposes. As discussed before, each valid working definition of AI has its value. Principle-based definitions are often described as "looking for a silver bullet", labeled

as “physics envy”, and rejected by arguments like “intelligence is too complicated to be explained by a few simple principles”. However, all these criticisms take such a definition (of Principle-AI) as the *means* to achieve other *ends* (Structure-AI, Behavior-AI, Capability-AI, or Function-AI), which is a misconception. The NARS definition may give AI a better identity than the other definitions do, though it does not produce all the values that can be produced by the others.

Obviously, the NARS definition of AI is not a *descriptive definition* of the term “AI”, that is, its common usage in the field, and most of the existing “AI systems” do not satisfy this definition. However, it does not necessarily mean that this definition should be rejected, but may imply that the field should change into a more coherent and fruitful discipline of science.

5. Conclusion

Though intuitively everyone agrees that AI means to build computer systems that are similar to the human mind in some way, they have very different ideas on where this similarity should be. Among the many existing definitions [32], typical opinions define this similarity in terms of structure, behavior, capability, function, or principle [3].

These working definitions of AI are all valid, in the sense that each of them corresponds to a description of the human intelligence at a certain level of abstraction, and sets a precise research goal, which is achievable to various extents. Each of them is also fruitful, in the sense that it has guided the research to produce results with intellectual and practical values.

On the other hand, these working definitions are different, since they set different goals, require different methods, produce different results, and evaluate progress according to different criteria. They cannot replace one another, or be “integrate” into a coherent definition that satisfies all the criteria at the same time.

The common beliefs on this topic, “AI cannot be defined” and “All AI definitions are roughly equivalent” are both wrong. AI can have working definitions that serve as ultimate research goals. Every researcher in the field usually holds such a definition, though often implicitly. To improve the coherence and efficiency of research and communication, it is better to make our working definitions explicit.

This topic matters for AI, since the current AI research suffers from the confusion of various goals and the missing of an identity. Consequently, many debates are caused by different meanings of the term “AI”, and the field as a whole is fragmented within, as well as has trouble to justify its uniqueness and integrity to the outside world.

This topic is crucial for AGI, given its stress on the “big picture” of an intelligent system. Even though at the current time no working definition is perfect or final, to dismiss the issue will damage the consistency of system design and evaluation.

Though there are many valid ways to define AI, they are not equally good. We will not reach a consensus on which one is the best very soon, so in the field the different working definitions will co-exist for a long time. Even so, it is important to understand their difference and relationship.

Different working definition gives the field of AI different identities. To solve the problems of internal fragmentation and external recognition, the most promising way is to define AI by a principle of rationality that is followed by the human mind, but not by traditional computer systems. The NARS project shows that such a solution is possible.

Reference

- [1] Ronald J. Brachman, (AA)AI: more than the sum of its parts, *AI Magazine*, 27(4):19-34, 2006.
- [2] Pei Wang and Ben Goertzel, Introduction: Aspects of artificial general intelligence, In Advance of Artificial General Intelligence, B. Goertzel and P. Wang (editors), 1-16, IOS Press, Amsterdam, 2007.
- [3] Pei Wang, On the working definition of intelligence, Technical Report No. 94, Center for Research on Concepts and Cognition, Indiana University, Bloomington, Indiana, 1994.
- [4] Pei Wang, Rigid Flexibility: The Logic of Intelligence, Springer, Dordrecht, 2006.
- [5] Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*, 2nd edition, Prentice Hall, Upper Saddle River, New Jersey, 2002.
- [6] W. Daniel Hillis, *The Connection Machine*, MIT Press, Cambridge, Massachusetts, 1986.
- [7] Paul Smolensky, On the proper treatment of connectionism, *Behavioral and Brain Sciences*, 11:1-74, 1988.
- [8] Jeff Hawkins and Sandra Blakeslee, *On Intelligence*, Times Books, New York, 2004.
- [9] Hugo de Garis, *Artificial Brains*, In *Artificial General Intelligence*, Ben Goertzel and Cassio Pennachin (editors), 159-174, Springer, Berlin, 2007.
- [10] George N. Reeke and Gerald M. Edelman, Real brains and artificial intelligence, *Dædalus*, 117:143-173, 1988.
- [11] Alan M. Turing, Computing machinery and intelligence, *Mind*, LIX:433-460, 1950.
- [12] Ian F. Brackenbury and Yael Ravin, Machine intelligence and the Turing Test, *IBM Systems Journal*, 41:524-529, 2002.
- [13] Lenhart K. Schubert, Turing's dream and the knowledge challenge, Proceedings of the Twenty-first National Conference on Artificial Intelligence, 1534-1538, Menlo Park, California, 2006.
- [14] Allen Newell, *Unified Theories of Cognition*, Harvard University Press, Cambridge, Massachusetts, 1990.
- [15] John R. Anderson and Christian Lebiere, The atomic components of thought. Erlbaum, Mahwah, New Jersey, 1998.
- [16] Michael L. Mauldin, ChatterBots, TinyMuds, and the Turing Test: entering the Loebner Prize competition, Proceedings of the Twelfth National Conference on Artificial Intelligence, 16-21, 1994.
- [17] Marvin Minsky, *The Society of Mind*, Simon and Schuster, New York, 1985.
- [18] Nils J. Nilsson, Human-level artificial intelligence? Be serious! *AI Magazine*, 26(4):68-75, 2005.
- [19] Douglas B. Lenat, Cyc: a large-scale Investment in Knowledge Infrastructure, *Communications of the ACM*, 38(11):33-38, 1995.
- [20] John McCarthy, What is Artificial Intelligence?
On-line paper at www-formal.stanford.edu/jmc/whatisai/whatisai.html, 2004.
- [21] David Marr, Artificial intelligence: a personal view, *Artificial Intelligence*, 9:37-48, 1977.
- [22] Stan Franklin, A foundational architecture for artificial general intelligence. In Advance of Artificial General Intelligence, B. Goertzel and P. Wang (editors), 36-54, IOS Press, Amsterdam, 2007.
- [23] Moshe Looks, Ben Goertzel, and Cassio Pennachin, Novamente: An Integrative Architecture for General Intelligence, In papers from AAAI Fall Symposium on Achieving Human-Level Intelligence through Integrated Systems and Research, 54-61, 2004.
- [24] Herbert A. Simon, *Models of Man: Social and Rational*, John Wiley, New York, 1957.
- [25] Stuart Russell, Rationality and intelligence, *Artificial Intelligence*, 94:57-77, 1997.
- [26] Marcus Hutter, *Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability*, Springer, Berlin, 2005.
- [27] James F. Allen, AI growing up: The changes and opportunities, *AI Magazine*, 19(4):13-23, 1998.
- [28] Drew McDermott, *Mind and Mechanism*, MIT Press, Cambridge, Massachusetts, 2001.
- [29] Rudolf Carnap, *Logical Foundations of Probability*, University of Chicago Press, Chicago, 1950.
- [30] Roger C. Schank, Where is the AI? *AI Magazine*, 12(4):38-49, 1991.
- [31] Patrick Hayes and Kenneth Ford, Turing test considered harmful, Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, 972-977, 1995.
- [32] Shane Legg and Marcus Hutter, A Collection of Definitions of Intelligence, In Advance of Artificial General Intelligence, B. Goertzel and P. Wang (editors), 17-24, IOS Press, Amsterdam, 2007.

Using Decision Trees to Model an Emotional Attention Mechanism

Saman HARATI ZADEH
Saeed BAGHERI SHOURAKI

Ramin HALAVATI

{*harati, sbagheri, halavati*}@ce.sharif.edu

Computer Engineering Department, Sharif University of Technology, Tehran, Iran

Abstract: There are several approaches to emotions in AI, most of which are inspired by human emotional states and their arousal mechanisms. These approaches usually use high-level models of human emotions that are too complex to be directly applicable in simple artificial systems. It seems that a new approach to emotions, based on their functional role in information processing in mind, can help us to construct models of emotions that are both valid and simple. In this paper, we will try to present a model of emotions based on their role in controlling the attention. We will evaluate the performance of the model and show how it can be affected by some structural and environmental factors.

Keywords: Emotions, Attention, Artificial Intelligence

Introduction

There exist several emotion theories that have been applied to produce emotional artificial systems and researchers such as Elliott [1, 2], Dyer [3, 4], Pfeifer [5, 6, 7, 8] and Reilly [9, 10] have implemented models of emotion management mechanisms in AI systems.

The emotion generation mechanism in human's mind presents several features that have key roles in information processing tasks like resource management, attention, learning and decision making. Modeling these features can help us not only to understand the concept of emotions but also to construct systems with a higher performance. For example, Harati Zadeh et. al [11] showed that a resource management approach to emotion could be applied in agents' decision-making system to improve its performance and generate a behavior that can be interpreted as emotional by human observers.

In this paper we will focus on the role of emotions in controlling the attention, and will try to model this feature of emotions for a system that has a single limited capacity

input channel that can check one input at a time. Our goal is not to present a complete emotion enabled system or to construct a perfect attention mechanism for a complex intelligent system. But we will try to show how emotions could have an attention-controlling role in artificial intelligent systems and how a simple emotion driven attention system could be affected by some key parameters of the system and the environment in which it is applied.

A final note that seems to be necessary is that in this paper we will refer to the proposed models as "emotional" or "emotion-driven" ones. Our purpose is to emphasize that some aspects of those models are inspired from emotion system of human and it does not mean that we believe that they are complete models of emotions.

1. Attention and Decision Making

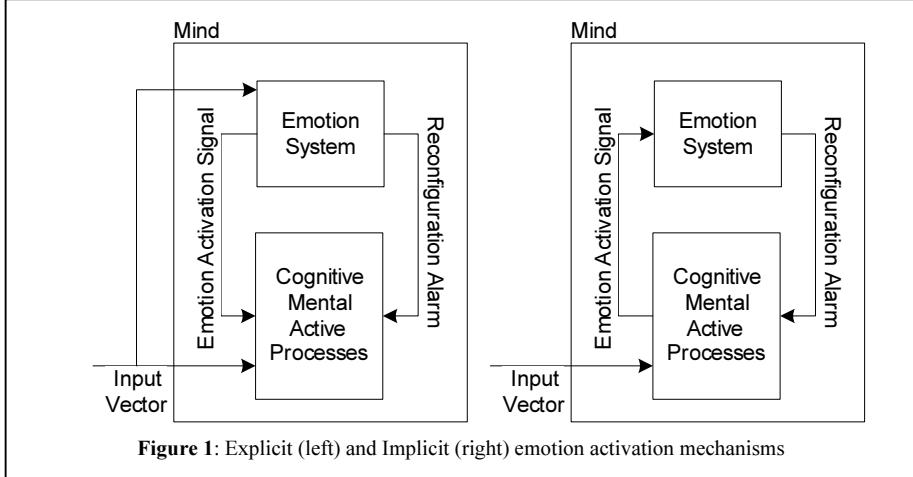
When someone enters some emotional state, he/she becomes more sensitive to some inputs, while ignoring the other ones that may be important in non-emotional states. For example someone who has been frightened strongly, possibly will temporarily forget the signals that say he/she is hungry.

New cognitive/computational theories of emotions, define emotional states as configuration of mental resources [12]. According to this definition, each emotional state, reconfigures the mental resources and helps the mind to change its behavior according to current situation. One of the possible effects of emotions could be the adjustment of the attention mechanism.

In AI domain, usually the researchers assume that the system receives all of *available* input values from its environment, based on which the system can define the current state and decide about the next action. However when the input channel of the agent has a limited band width, that is the agent must pay for checking each single input value, an attention guide system will improve its performance by saving the computation time and space.

One of the concepts in AI that supports a simple form of attention is the decision tree. A decision tree can be translated to a set of checking priorities over the possible inputs. Through these priorities, the decision making system in each step puts a single input in its attention window based on the values of inputs checked so far. Therefore, the attention mechanism follows some general strategy that is implicitly defined by the decision tree.

There are several algorithms to construct decision trees [13]. Some of these algorithms construct the tree after the learning phase, and they assume that the agent has already gathered a complete set of knowledge from its experiences in the environment [14]. There exist other algorithms that let the agent to construct the tree incrementally [15]. In addition, some heuristics have been proposed to construct the trees in situations that the agent's knowledge is incomplete or inaccurate [16]. However, from the view point of attention, all of these algorithms are more or less the same. To make our discussion more clear, here we will assume that the agent's



knowledge, and the decision tree constructed based on it, is correct, complete and sufficient for making correct decisions.

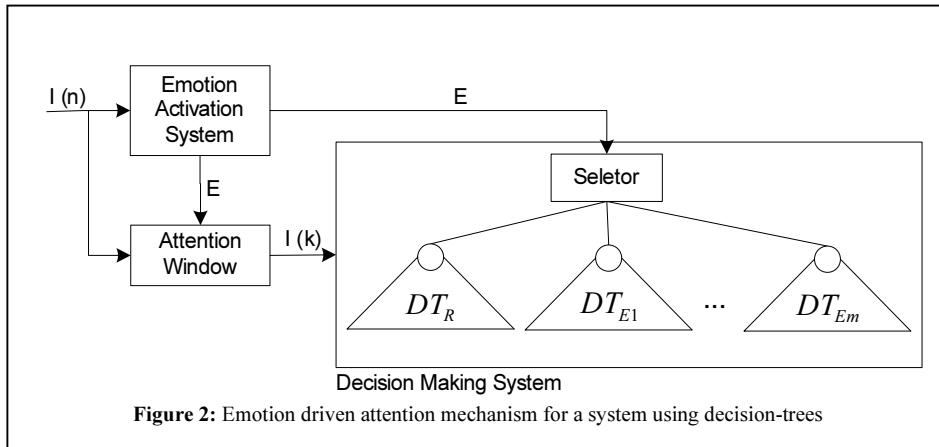
2. A Model for Emotional Attention

Amygdala is a part of the brain that is responsible for most of emotion-related mental jobs [17]. This part receives sensory inputs as well as signals from various other parts of brain and generates emotional signals to pass to other areas in the brain especially to those that have important roles in decision-making [18].

Therefore, an emotional state can be initiated if Amygdala recognizes some certain pattern in raw sensory inputs, or if it receives some emotion arousing processes information from other parts of the brain (figure 1).

The main difference between these two types of emotion arousal is that in the first one, the emotion-activation mechanism monitors the input signals subconsciously, and activates an emotional state at the same time that the emotion-arousing pattern appears in input channel, however, in the second mechanism, subconscious input monitoring is not performed and therefore, the emotion activation is delayed until the conscious cognitive mechanisms identify the situation as emotional. In this paper we will refer to the first approach as explicit emotion activation and to the second one as implicit emotion activation.

In both cases there must be some sort of alarm system that informs mind about the new active emotion according which the mind could be reconfigured to prepare for the new situation [19]. However, from the viewpoint of attention mechanism, this reconfiguration means deactivating the current attention strategy and activating a new



one. Therefore, the general model of emotion driven attention system will be like the one depicted in figure 2.

In this schema, “I” stands for input vector, “n” and “k” determine the number of inputs in the input vector and “k” is smaller than or equal to “n”. The Attention Window could be imagined as a filter or mask that passes useful input values and hide other ones from decision making system. The signal “E” going out from Emotion Activation System reconfigures attention window such that the input vector passing it is reduced to a subset of input values that are useful to making decision in current situation. The same signal goes to Decision Making System to inform it about current active emotion.

The agent uses decision trees for making decision so it must have a decision tree constructed based on each “k” values that it receives through the attention window. In other words the decision maker will have a set of decision trees of which, one is for non-emotional states and the others are for emotional states.

As shown in this schema, the selector subsystem uses the current active emotion signal “E” to select the suitable decision tree that is the suitable attention-shifting sequence, for current state. DT_R is the decision tree for non-emotional state and DT_{E1} to DT_{Em} are decision trees for emotional states constructed according to the input values that the decision making system receives in those states.

In the next subsections we will explain how the explicit and implicit emotion activation mechanisms can be embedded in presented general model for attention. We will refer to the attention mechanism based on explicit emotion activation as "explicit emotional attention system" and the one based on the implicit emotion activation as "implicit emotional attention system".

2.1. Explicit emotional attention system

This version of attention system is based on subconscious input monitoring that does not use sophisticated cognitive mechanisms. Therefore we will try to keep the

- 1- Construct the Main Decision Tree (MDT) based on complete knowledge base.
- 2- Compute the average number of input checks for MDT.
- 3- For each input “i”
 - a. For each possible value “v” for “I”:
 - i. If there is no rule in knowledge base in which $i = v$, ignore v.
 - ii. Construct the decision tree $DT_{i=v}$ based on those rules in which $i = v$
 - iii. Compute the Average number of Input checks “AIC” for $DT_{i=v}$
 - iv. Compute the Average number of Input Checks “MAIC” in MDT, for the states in which $i = v$.
 - v. Define the average check reduction ACR as “MAIC – AIC”.
- 4- If no (i, v) causes a check reduction bigger than zero terminate the process.
- 5- Find the input “I” and value “V” such that $DT_{I=V}$ presents the maximum average check reduction.
- 6- Add emotion $E_{I=V}$ with the condition $I = V$ to the emotion interrupt system.
- 7- Exclude the rules in which $I=V$ from the knowledge base.
- 8- If the emotion system has space to add another emotion, go to step 1.

Figure 3. The algorithm for defining emotion interrupts

mechanism of emotion activation as simple as possible. To do so, we decided to construct the emotion interrupt system such that each emotion is sensitive to only one input variable and the corresponding emotion signal is triggered when that input variable takes a certain value. In addition, we have assumed that the maximum number of emotions is known in advance.

Assuming that the system already has the required rule base for making the best decisions, the algorithm presented in figure 3 will use the agent's knowledge to construct its emotion set. In this algorithm, all the inputs have been assumed to take values in a discrete domain. This algorithm checks all the situations that may arise by setting one of the inputs to a specific value in its domain. For each (input, value) pair there is a set of rules in the rule base that can be applied at least in one of those situations. Based on these rules, the agent can construct a decision tree that is smaller than the main decision tree constructed based on the whole rule base, but is valid only when “input = value”. Therefore, if the agent be somehow informed that this condition is met, for example through the emotion system, it will be able to check fewer inputs by using it instead of the main decision tree. However, since the capacity of interrupt system is limited, the agent must assign interrupt signals to the situations that their corresponding decision trees cause more reduction in input checking.

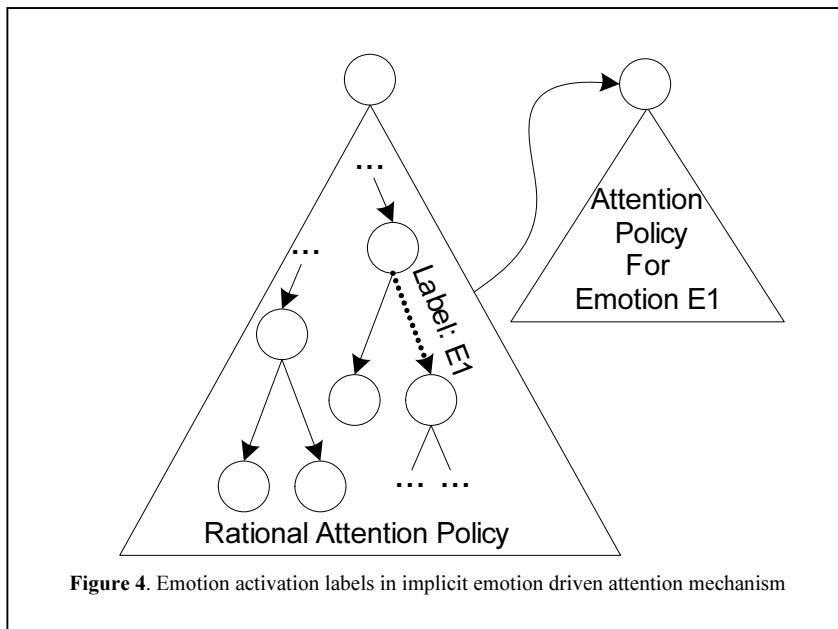
An important aspect of this algorithm is that it excludes the part of rule base that matches with activation criteria for emotion defined so far before going to the next turn to find the next emotion (line 7). In the emotion system, the emotions extracted first will have a higher priority than those ones that have been extracted later.

2.2. Implicit emotional attention system

In many applications, like most software intelligent systems, the agent is not always able to have an interrupt mechanism and it must pay for each single input that is checked. To evaluate the performance of an emotion driven attention system in such environments we replaced the explicit emotional attention mechanism with the implicit one that is initiated inside the agent's decision making system (figure 4).

As presented in figure 5, in this model, the agent assigns emotion labels to certain input values. When the agent faces one of these input values during its non-emotional decision-making, it deactivates the current input priorities, and activates the emotion assigned to that input value. The active emotion proposes its own policy for agent's attention system, and the agent uses that policy as long as the emotion proposing it is active. When the agent returns to non-emotional state, the non-emotional attention policy will be activated again.

The algorithm shown in figure 5 shows the emotion extraction mechanism for this version of the attention system. As it is clear, this algorithm does not exclude the part of knowledge matching with emotions defined so far from the knowledge base. Therefore, the MDT does not change during the emotion definition process.



3. Implementation and Results

To implement the proposed model we used a model of an agent living in an environment called Logic World. In Logic World, there are a set of Boolean variables

- 1- Construct the Main Decision Tree (MDT) based on complete knowledge base.
- 2- Compute the average number of input checks for MDT.
- 3- For each input “i”
 - a. For each possible value “v” for “I”:
 - i. If there is no rule in knowledge base in which $i = v$, ignore v.
Construct the decision tree $DT_{i=v}$ based on those rules in which $i = v$
 - ii. Compute the Average number of Input Checks “AIC” for $DT_{i=v}$
 - iii. Compute the Average number of Input Checks “MAIC” in MDT, for the states in which $i = v$.
 - iv. Define the average check reduction ACR as “MAIC – AIC”.
- 4- Find the input “I” and value “V” such that “ $DT_{I=V}$ ” presents the maximum average check reduction bigger than zero and no emotion with condition $I = V$ has already been defined.
- 5- If no such “I” and “V” found terminate the process.
- 6- Add emotion $E_{I=V}$ with the condition $I = V$ to the emotion system.
- 7- Add the label “ $E_{I=V}$ ” to all of arcs in MDT that assign value “V” to input “I”.
- 8- Go to step 3.

Figure 5. The algorithm for defining emotions for implicit emotional attention system

and values of these variables define the current state of the environment. Each state remains active for a predefined period and then the values of variables change randomly. The correct action that the agent must do in each state is defined by a function called behavioral function. We have assumed that the agent has already learnt the behavioral function through previous experiences during its life. Its knowledge base contains a set of rules assigning an action to each state; however, the agent is not aware of the current state of the environment before checking the inputs it receives. In each experience, if the correct action is selected, the agent receives a positive reward value and if not, it is punished by a negative penalty.

The agent uses emotion extraction process, to build an explicit or implicit emotional attention control mechanism. We have performed a series of tests to examine the effect of these emotion driven mechanisms on the average reward that the agent receives in long term.

The emotion arousal process and its effect on the system’s performance could be affected by several parameters, among which we focus on some of the most important ones, for which the results are presented. In following test, the average reward is computed for 10 million decisions. The agent receives “+1” reward for a correct action and “-1” penalty for a wrong one. Emotional agent has five emotions with inertia equal to 5 cycles, 10 possible actions and receives an input vector with 15 inputs. We will mention it if a test has different values for these parameters.

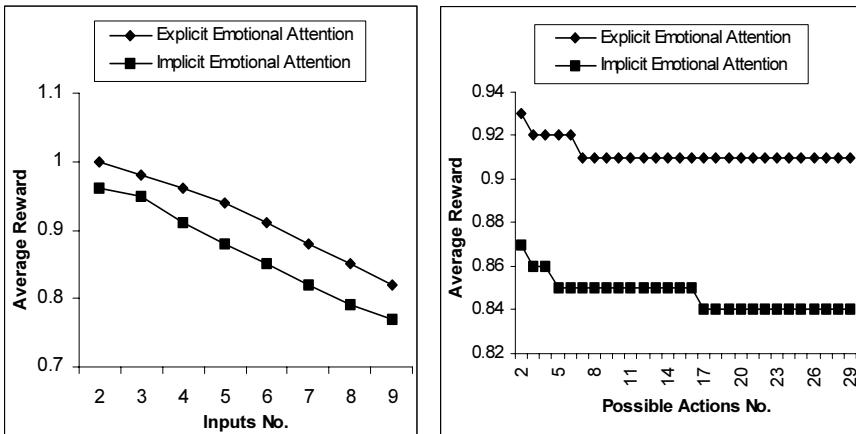


Figure 6. Average received reward vs. size of input vector (left) and number of possible actions (right)

- **Complexity of the environment and Diversity of the actions space**

In Logic World the complexity of the environment is solely defined by the number of inputs that have a role in behavior function. In simple environments the correct action could be deduced from a smaller set of inputs while in more complex environments agent must check several inputs before it can decide. The plot shown in figure 6 (left) presents the average reward that the agent receives for different sizes of input vector.

Another important parameter in an agent's performance is the number of actions it can choose. For an agent with a small set of actions, the behavior is more reactive, while for an agent with a rich set of actions, there are more choices in each cycle, and therefore more deliberation is required to make a correct decision. The graphs shown in figure 6 (right) presents the average reward received by the agent using each of emotional attention control systems with different number of possible actions to choose.

The results show that the average reward decreases as the size of input vector increases. It says that the more complex is the agent environment, the richer set of emotions is required for keeping the agent's performance at a constant level. There is no important difference between the two trajectories, except that the average reward received by the agent using explicit emotional attention system is higher than that of the agent using the other one. This is an expectable result, because an interrupt system like the one used in explicit emotional system does not waste the agent's time to activate the suitable emotion.

Also it can be seen that the explicit emotional attention mechanism is less sensitive to the number of actions compared with the implicit one. One explanation for this could

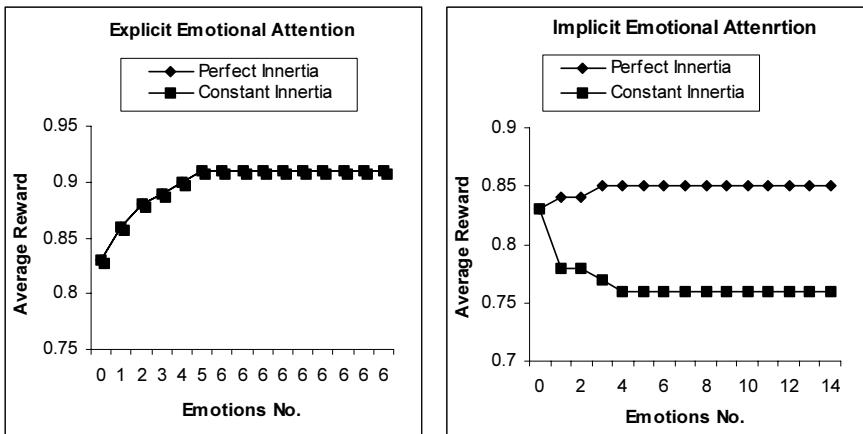


Figure 7. Average received reward vs. maximum number of definable emotions

be that the size of the rational decision tree that increases if we increase the number of possible actions. Therefore, the average number of inputs that the agent checks before it realizes that some emotional tree must be activated grows. This reduces the average reward that the agent with the implicit emotional attention system receives. The explicit emotional attention mechanism does not face this problem because the emotional situation is recognized directly and is not dependent to the size of the rational decision tree.

Another important result of this experiment is that both attention systems are more sensitive to the size of input vector compared to the number of possible actions. The performance of the agent remains constant if the number of possible actions stays in a certain domain, while it is decreased by each single input added to the input vector.

- **The number of emotions that the system supports and the effect of emotions' inertia**

Figure 7 shows the effect of increasing the number of emotions supported by the system on its performance. There are two sets of results for each system, one for emotions with constant predefined inertia and one for emotions with the perfect inertia. An emotion has a perfect inertia if it is active only if the current environmental state recommends it.

The results show that in explicit emotional system, the maximum number of emotions is smaller than that of the implicit emotional system. It is because the size of the non-emotional decision tree is reduced by adding each emotion to the explicit emotional system. Therefore, after constructing certain number of emotional interrupts, the non-emotional decision tree becomes so small that no new emotion can reduce its size, and the emotion extraction process stops.

Another important result of this test is that for the explicit emotional system the performance is the same for perfect and constant inertia. However for the implicit emotional system, the performance increases by new emotions with perfect inertia being added to the system, while adding emotions with constant inertia has a degrading effect on the average received reward. One can conclude from these results that the inertia assigned to each emotion has a very important role in system average performance. A wrong inertia for an emotion can cause the agent to stay in an emotional state while the environmental state has changed and does not recommend the active emotion anymore. This could be a very challenging problem if we want a system to assign the best inertia to its emotional states.

4. Conclusion

In this paper, we tried to present two simple models of emotion-guided attention system. We evaluated the effect of some parameters that seem to be important in performance of such systems. The results of this experiment could be summarized as follows:

- An emotion driven attention mechanism, if implemented explicitly as an interrupt system, can improve the performance of the system. Explicit interrupt based emotional state are more beneficial without inertia.
- An implicit emotional attention mechanism can potentially increase the performance of the system given that the inertia values assigned to emotions are adjusted precisely. If inertia values are not picked accurately, the resulting emotion-enabled system may not be helpful and can lead to a performance lower than that of the emotion-less system. It is important to note that adjusting the emotions' inertia could be a difficult problem, because it has a strong relation with the definition of the emotional states and assignment of the environmental states to emotional states. Therefore these problems should be solved together.
- An emotion subsystem can improve the decision making at least in two ways:
 - By reducing the amount of processing that the agent performs non-emotional situations.
 - By providing a cheaper way to decide in emotional states.
 Therefore, the goal of constructing each emotion determines the method using which one must construct it. In other word, the emotion definition is a context dependent task.
- Both the diversity of state space and the agent's action space have an important role in the performance of the presented emotion driven attention system, however, the first one affects the system more seriously.
- The performance of emotion activation mechanism is an important factor in emotion-enabled systems. An inefficient mechanism for recognizing the suitable emotion for current state can reduce the performance of whole system

seriously. However, there is a trade off between allocating low extra resources for emotion subsystem and the performance that one can expect from it. Assigning some inertia to the emotions can help us to reduce the amount of processing that should be dedicated to the emotion activation mechanism.

In this paper we assumed that the required knowledge for building the emotional attention system has already been gathered by the agent and the emotion extraction process is performed on this knowledge. However finding incremental algorithms for generating emotion management mechanisms during agent's learning phase could be beneficial.

References

- [1] Elliott, C. (1992), The Affective Reasoner: A Process Model of Emotions in a Multi-agent System. Ph.D. Dissertation, Northwestern University, The Institute for the Learning Sciences, Technical Report No.32.
- [2] Elliott, C. (1997), I picked up catapia and other stories: A multimodal approach to expressivity for "emotionally intelligent" agents. In: Proceedings of the First International Conference on Autonomous Agents.
- [3] Dyer, M.G. (1982). In-depth understanding. A computer model of integrated processing for narrative comprehension. Cambridge, MA: MIT Press.
- [4] Dyer, M.G. (1987). Emotions and their computations: Three computer models. *Cognition and Emotion*, 1 (3), 323-347.
- [5] Pfeifer, R. (1982). Cognition and emotion: An information processing approach. Carnegie-Mellon University, CIP Working Paper Nb. 436.
- [6] Pfeifer, R. (1988). Artificial intelligence models of emotion. In V. Hamilton, G. Bower, & N. Frijda (Eds.). Cognitive perspectives on emotion and motivation: Proceedings of the NATO Advanced Research Workshop (pp. 287-320), Dordrecht: Kluwer.
- [7] Pfeifer, R. (1994). The "Fungus Eater" approach to the study of emotion: A view from Artificial Intelligence. Techreport #95.04. Artificial Intelligence Laboratory, University of Zürich.
- [8] Pfeifer, R. (1996). Building "Fungus Eaters": design principles of autonomous agents. In P. Maes, M. J. Mataric, J.-A. Meyer, J. Pollack, and S. W. Wilson (Eds.), Proceedings of the fourth international conference of the society for adaptive behavior (pp. 3-12). Cambridge, MA: MIT Press.
- [9] Reilly, W. S. and Bates, J. (1992). Building emotional agents. Technical Report CMU-CS-92-143, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.
- [10] Reilly, W.S. (1996). Believable social and emotional agents. PhD thesis. Technical Report CMU-CS-96-138, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.
- [11] Harati Zadeh, S., Bagheri SHouraki, S., Halavati, R. (2006), Emotional Behavior: A Resource Management Approach. *Adaptive Behavior*, 14 (4), 357-380.
- [12] Minsky, M.,(2006), The Emotion Machine: Commonsense Thinking, Artificial Intelligence, and the Future of the Human Mind, Simon & Schuster.
- [13] Kothari, R. and Dong, M. (2001) Pattern Recognition: From Classical to Modern Approaches: 169-184. Singapore : World Scientific.
- [14] Quinlan, J. R. (1986), Induction of Decision Trees. *Machine Learning* 1(1): 81-106.
- [15] Berkman N. C., Utgoff P. E. and Clouse, J. A.. (1997), Decision tree induction based on efficient tree restructuring. *Machine Learning*, 29(1): 5-44.
- [16] Quinlan, J. R., (1993) C4.5 Programs for Machine Learning. California: Morgan Kaufmann.
- [17] LeDoux, J. E. (1996). The emotional brain: The mysterious underpinnings of emotional life. New York: Simon and Schuster.

- [18] Moren, J., Balkenius, C. (2000), A computational model of emotional learning in the amygdale, in “From animals to animats”, Mayer J.A., et al. eds., MIT press.
- [19] Sloman, A.(2002), Architecture-Based conception of Mind, In P. Gardenfors, J. Wolenski and K. Kijina-Placek (Eds.), The scope of logic, methodology and philosophy of science, Vol. II, (pp.403-427), Dordrecht: Kluwer.

This page intentionally left blank

Position Statements

This page intentionally left blank

Fusing Animals and Humans

Jonathan CONNELL

IBM T.J. Watson Research Center, Yorktown Heights NY, jconnell@us.ibm.com

Abstract. AI has many techniques and tools at its disposal, yet seems to be lacking some special “juice” needed to create a true being. We propose that the missing ingredients are a general theory of motivation and an operational understanding of natural language. The motivation part comes largely from our animal heritage: a real-world agent must continually respond to external events rather than depend on perfect modeling and planning. The language part, on the other hand, is what makes us human: competent participation in a social group requires one-shot learning and the ability to reason about objects and activities that are not present or on-going. In this paper we propose an architecture for self-motivation, and suggest how a language interpreter can be built on top of such a substrate. With the addition of a method for recording and internalizing dialog, we sketch how this can then be used to impart essential cultural knowledge and behaviors.

Keywords. Agent architecture, Motivation, Learning, Language, Clicker training

1. What is intelligence?

In order to achieve AI it helps to know how the end product will be evaluated. In particular, we focus on “perceived intelligence” – not what might count as the true Platonic ideal of intelligence, but what properties an external observer might take as evidence that there is something there. In the natural world the layered ordering shown below is what is typically observed. There are arguably more organisms that are aware than ones with personality (e.g. ants), and there are animals that are social without having much in the way of abstraction capabilities (e.g. guinea pigs). Yet it is not clear that the lower levels are an absolute prerequisite for the higher levels. For instance, there are many computer programs that deal in abstractions but have no personality.

Criteria for *Perceived* Intelligence

- 1. Animate** – Coordinated movement, many degrees of freedom
- 2. Aware** – Responds and changes actions based on environment change
- 3. Personality** – Individuals have different likes / dislikes, preferences learned
- 4. Social** – Aware of social order, use other beings as agents
- 5. Abstract** – Conceptualize situations remote in space and time, planning
- 6. Communicative** – Express internal ideas and ingest situational descriptions

To better understand the above criteria imagine applying them to a robot toy. Obviously if it just sits there and does nothing it is pretty boring – it has failed criterion 1. Now suppose that it can zoom around at high speed but constantly runs smack into things. It seems vaguely alive but not very smart at all – it has failed criterion 2. The next step is to exhibit some sort of personal preferences for things, activities, situations, or people. The Roomba vacuum cleaner, created as a tool, fails criterion 3. By contrast, Furby, an animatronic toy that complains about being turned upside down, succeeds. Furby also partially passes criterion 4 because it is forever badgering its owners to “Feed me! Yummm.” and having them comply. Although annoying, Furby is arguably closer to what we want in a true AI than many other artifacts (or at least it has what most other AI programs lack).

The next two criteria seem more applicable to animals than robots. There is a lot of literature on parrots, crows, pigs, dogs, dolphins, monkeys, and chimps concerning tool use, sequential tasks, and delayed reward scenarios – all activities requiring some proficiency at criterion 5. By contrast, criterion 6 seems largely limited to humans. Much of this is tied to language, something animals have been able to acquire only to a limited extent [1]. In many ways language is absolutely required for being human. People do not exist as singletons: we are all part of a progressively more tightly coupled “super-organism”. Language is the basis for this coupling – it allows us to have knowledge of things our bodies have never experienced directly, particularly things remote in space or time (cf. criterion 5). To be human is to be able to participate in this cultural super-organism, and to do that requires language.

Looking to existence proofs for inspiration, when watching an animal that has undergone clicker training it really feels like “someone” is there. The whole training paradigm provides a nice mechanism for autonomous goals and learning [2]. On the other hand, there are programs like BORIS [3] which interprets narratives and handles natural language questions. This exhibition of deep language understanding is hard to ignore. Moreover, BORIS allows its deductions to be tailored by instruction, something that feels very human. To get the best of both we propose fusing animals and humans.

2. The animal part – general purpose motivation

The animal part is largely about motivation: why animals do what they do and when they do it. Yet it is important to realize that animals spontaneously do things all the time. Pets do not await a command from their owners, and wild animals obviously have no external agency calling the shots. In this way they are very different from contemporary computer programs which amplify and elaborate on imperatives supplied by their users. Moreover, although goals can be internally generated rather than supplied from the outside, it seems unlikely that animals are always operating in a goal-driven manner, except in the loosest sense. That is, much of what animals seem to do is routine or reflexive – there is no specific articulated goal which they are pursuing. Ongoing activities such as foraging, grooming, and migration can not be traced back through a logical chain of reasoning to a concrete goal proposition. To mirror this observation, the motivation system proposed here has only a loose coupling between goals and actions. As shown in Figure 1, the bulk of the activity is controlled reactively using situation-action policies [4, 5]. Some of these policies are active all the time, while others can be switched in or out depending on the situation [6]. Much of the

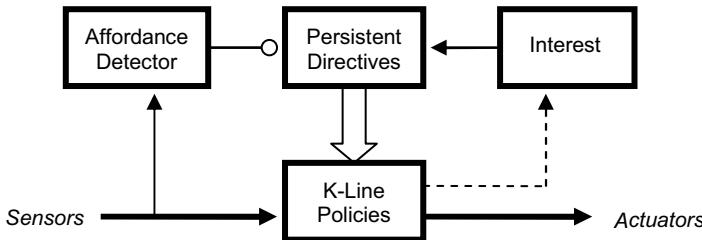


Figure 1. A loosely coupled architecture for self-motivation.

switching of polices is governed by a set of persistent directives. In this respect the action component of the architecture is similar to Minsky's K-lines [7].

Yet where do these policies and directives come from? One possible answer is that we can exploit $\langle S, E, A \rangle$ triples. Here E is an exciting event, S represents the observed situational context, and A captures the current actuator settings. These $\langle S, E, A \rangle$ triples are not recorded all the time, but only when something "interesting". The rules governing their formation and use are detailed below:

$$\begin{array}{ll} S \& E \& A \& I(E) \rightarrow \langle S, E, A \rangle & \langle S, E, A \rangle \& S \& I(E) \rightarrow D(E) \\ D(E) \& \langle S, E, A \rangle \& S \rightarrow A & D(E) \& \langle S, E, A \rangle \rightarrow I(S) \end{array}$$

There are three uses for $\langle S, E, A \rangle$ triples. First they can function as affordance detectors – determining when the environment may be offering a useful opportunity to the agent. In this case the recorded S is used to predict the interesting additional observation E that might already hold, or may be forced to become present through some action. Although E was originally "interesting" when it was recorded, the organism's desires and needs routinely change over time. Thus each such proposed affordance is evaluated against the current "interest" metric (I). Then, if sufficiently stimulating, a desire (D) for this event is latched in as one of the persistent directives that control the behavioral policies. The other major use for $\langle S, E, A \rangle$ triples is as part of a policy. If the E part matches one of the current directives then when situation S occurs the agent will be prompted to try action A. Triples can also be used for loose backwards chaining. If E is one of the active directives then the agent should become interested in situation S. In this way it can serendipitously learn how to accomplish subgoals in the course of an activity (whether or not the associated action is taken).

As an example, suppose you are walking along the shore of a pond and there is a sudden splash as a frog jumps into the water. Since sudden noises are intrinsically interesting this would prompt the formation of a triple like $\langle \text{pond}, \text{splash}, \text{walk-along} \rangle$. Now every time you see a pond the splash will be brought to mind. If this is still an interesting occurrence you will latch it in as something you desire to happen. This in turn activates one or more policies that bias you into doing things that might lead to a splash, such as if you see a pond you should walk along its shore. The desire for splashing may also activate other triples such as $\langle \text{pond} \& \text{rock}, \text{splash}, \text{drop-rock} \rangle$. Since the situation part of this is not yet fulfilled its action is not performed. Yet the set of interesting things will be expanded temporarily to include rocks, prompting background learning about where to find rocks (for this or any other purpose).

3. The human part – a language interpreter

The bulk of human-level intelligence is due to acquired cultural knowledge, at least observationally. If a person suffers amnesia and forgets his family and culture, in many ways he has been erased as the person he previously was. Thus it is essential for an AI to have the ability to learn patterns of cultural interaction. Language is both the key for learning this and the medium by which proficiency in culture is evidenced [8]. It is a sort of mind-sharing “telepathy” that makes an agent part of a cultural super-organism and allows its peers to “program” it to act in certain ways in specific situations [13].

At the bottom, language can be viewed as a general-purpose scripting system that invokes sensory/motor specific subroutines as needed. One way to become proficient in a language is to build an interpreter that allows the agent to receive descriptions and instructions via language then evaluate these structures to operationalize the knowledge. The starting point for this is to ground the fragmentary acoustic utterances in sensory and actuator experiences [9, 10]. As a prerequisite the system must have some innate mechanisms of parsing the world into physical objects and temporal events. It also needs some reflexes which will exercise all the actuator options and an initial good/bad reinforcement signal as feedback. An intrinsic means to guide attention is also very useful. All these abilities can be bootstrapped to higher levels of sophistication with a method like clicker training (i.e. operant conditioning with staging and shaping) [11].

Once a basic level of linguistic grounding has been achieved, learning new concepts and tasks becomes much easier [12]. For instance, one can say “No, this is a moth not a butterfly. Look at its fuzzy antennae.” This explanation is much simpler and faster than showing the agent tens of contrasting examples and having it guess which features are relevant to the classification. Similarly, language can be used to directly impart routines for accomplishing sequential tasks, such as opening a jar of pickles. Instead of letting the agent fiddle with it for hours and shouting “Hurray!” when the top finally pops off, language allows the teacher to say “Hold the jar in your left hand, grasp the top with your right hand, and twist hard”. If this sequence of verbal directions is memorized and played back through the interpreter, the agent has essentially learned a plan for how to do the task in one shot (perhaps without ever actually performing it!).

Recording is one aspect of internalizing dialog [8], but there is more. The agent may start out with a running stream of remembered commentary: “Okay, the stop sign is coming up. Slow down and watch for other cars at the intersection. If the car on the cross street arrives first he gets to go first. It looks like there is no one around, so you can go now ...” Yet this scaffolding can eventually be compiled out [14] until all that is left are the relevant behavioral policies. In particular, *compiling out* responses can yield internal deduction without requiring any special-purpose “declarative assertion” action module. The agent predicts what it is going to say and what this will sound like. It can then get the effect of the interpreter running on that acoustic input directly without actually speaking out loud. That is, the middle two steps below get compiled out.

(see: shaggy animal → say: “It’s a dog”) → (hear: “It’s a dog.” → represent: dog)

see: shaggy animal → represent: dog

Similarly, *compiling in* questions can yield behaviors for directing attention and more thoroughly processing input. Here a teacher-narrated sequence about identifying birds is condensed by means of removing the dialog components in the middle.

(see: bird → hear: “It’s a bird!”) → (hear: “What shape is its beak?” → look: at beak)
 see: bird → look: at beak

4. Discussion

Our basic recipe for intelligence has three steps. First, the animal part provides a substrate for learning an interpreter for language. Second, language in turn unlocks a method for rapid transmission of concepts and behavior patterns. Third, competent performance within such a defined culture comprises the human part. So why not just build a language interpreter and jettison the animal heritage? Presumably the agent could be trained to be properly curious, ask questions when appropriate, and initiate further information gathering activities when needed. Perhaps the answer is that the animal part lets the agent fall back on *weak methods* [15] when its programming fails. Instead of just crashing and staring blankly ahead until it receives new top level instructions, if the agent maintains the proper set of directives and interest biases it can muddle through to some state where the more detailed program can pick up once again.

In this paper we proposed six criteria for perceived intelligence, sketched an architecture for the lower three, and argued that top one was largely a function of language. What then of the middle two? It has been posited that deep social intelligence is another hallmark of humanity [16]. Yet some interesting work [17] has shown how a beginning theory-of-mind can be built on top of mechanisms like shared gaze, all implemented in an animal-like architecture. Abstract thought, the remaining criterion, may need some pre-existing mechanism (e.g. spatial navigation) for language to latch on to. Then again, it may be that abstract abilities like planning can emerge as a natural outgrowth of remembered or internalized activity patterns (cf. opening the pickle jar).

References

- [1] E. Kako, “Elements of syntax in the systems of three language-trained animals”, *Animal Learning & Behavior*, 27(1), pp. 1-14, 1999.
- [2] L. Saksida, S. Raymond, and D. Touretsky, “Shaping Robot Behavior Using Principles from Instrumental Conditioning”, *Robotics and Autonomous Systems*, 22 (3/4):231, 1998.
- [3] Michael Dyer, In-Depth Understanding, MIT Press, 1983.
- [4] R. Brooks, “A Robust Layered Control System for a Mobile Robot”, *Journal of Robotics and Automation*, RA-2, pp. 14-23, 1986.
- [5] J. Connell, Minimalist Mobile Robotics, Academic Press, 1990.
- [6] J. Connell and P. Viola, “Cooperative Control of a Semi-Autonomous Mobile Robot”, *Proc. of the IEEE Conf. on Robotics and Automation (ICRA-90)*, pp. 1118-1121, 1990.
- [7] M. Minksy, “K-Lines: A Theory of Memory”, MIT AI Memo 516, 1979.
- [8] Lev Vygotsky, Thought and Language, MIT Press, 1962.
- [9] L. Steels and F. Kaplan, “AIBO’s first words”, *Evolution of Communication*, 4(1), pp. 3-32, 2001.
- [10] D. Roy, “Semiotic Schemas”, *Artificial Intelligence*, 167(1-2), pp.170-205, 2005.
- [11] Karen Pryor, Don’t Shoot the Dog, Simon & Schuster, 1984.
- [12] J. Connell, “Beer on the Brain”, *Proc. of the 2000 AAAI Spring Symposium, My Dinner with R2D2: Natural Dialogues with Practical Robotics Devices*, pp. 25-26, 2000.
- [13] B. F. Skinner, Verbal Behavior, Appleton-Century Crofts, 1957.
- [14] John R. Anderson, Rules of the Mind (Chapter 4), Lawrence Erlbaum, 1993.
- [15] J. Laird, A. Newell, and P. Rosenbloom, “Soar: An Architecture for General Intelligence”, *Artificial Intelligence*, 33(1), pp. 1-64, 1987.
- [16] E. Herrmann, J. Call, M. Hernández-Lloreda, B. Hare, and M. Tomasello, “Humans Have Evolved Specialized Skills of Social Cognition”, *Science*, vol. 317, pp. 1360–1366, 2007.
- [17] B. Scassellati, “Theory of Mind for a Humanoid Robot”, *Autonomous Robots*, vol. 12, pp. 13–24, 2002.

Four Paths to AI

Jonathan CONNELL^a and Kenneth LIVINGSTON^b

^a IBM T.J. Watson Research Center, Yorktown Heights NY, jconnell@us.ibm.com

^b Psychology Department, Vassar College, Poughkeepsie NY, livingst@vassar.edu

Abstract. There are a wide variety of approaches to Artificial Intelligence. Yet interestingly we find that these can all be grouped into four broad categories: Silver Bullets, Core Values, Emergence, and Emulation. We will explain the methodological underpinnings of these categories and give examples of the type of work being pursued in each. Understanding this spectrum of approaches can help defuse arguments between practitioners as well as elucidate common themes.

Keywords. Emergence, Animal Models, Consciousness, Language Understanding

1. Introduction – How can we achieve AI?

Artificial Intelligence has been pursued for over 40 years and has given rise to hundreds of different approaches. Early progress seemed rapid but halfway to Turing's goal of human-level AI the enterprise seemed to stall. In recent years a new generation of researchers has proposed a variety of ways to re-animate the search for general purpose AI. These proposals are diverse, and it is difficult to place bets on which approach might eventually prove successful, in large measure because the varied landscape of approaches is difficult to comprehend in a glance. We suggest, consistent with Lakatos's view of how scientific value is actually judged [1], that what is really needed is an effective way to catalog the different approaches. This then gives us a way to comprehend and evaluate the relative progress made by pursuing different approaches to building AI. This classification scheme can also help sort out whether an objection to some piece of work is directed at the technology itself, or rather at its methodological class. For instance, imagine a pacifist confronting a hawk. Instead baldly asserting that, "A gun will not solve your problem," a more constructive response would be, "Well, if you are going to fight, a gun is a reasonable weapon."

2. Silver Bullets – Just add missing mechanism X!

The approaches described here all have in common the suggestion that most of the necessary technology is already in place, we just need to resolve some particular nugget and then whole system will finally exhibit true intelligence. With this approach there is still some worry as to whether we have picked the right hole to fill. After all, a silver bullet will kill a werewolf, but not a vampire.

Fancy Logic – The idea is that first-order logic seems inadequate to the task of building AI, but that it can be achieved by moving to some more complex formal

system of symbol manipulation. Techniques include various extensions of logic (e.g. second-order, non-monotonic, epistemic, deontic, modal, etc.) as well as other mechanisms like circumscription, abduction [2], and inductive logic programming.

Inexact Reasoning – The premise here is that formal symbol manipulation, like first-order logic, is too brittle for the real world. Things are not black-and-white but rather shades of gray, and AI systems need to be able to reason in this manner. Some interesting progress has been made using Fuzzy Logic for mobile robots [3].

Deep Language – An AI cannot be expected to be fully competent straight “out of the box”, instead it needs to learn from sympathetic humans and/or from reading written material. To do this it must have a deep understanding of human language. To do this often involves a tight intermingling of syntactic and semantic [4].

Embodiment – Proponents of the embodiment solution to finding AI argue that you cannot achieve human-like intelligence unless the system has a body and can interact with the real physical world. Being embodied makes you care about objects, space, uncertainty, and actions to get tasks accomplished. In an important sense the body is just a special purpose computational engine, one that has evolved to solve very specific problems that are computationally expensive or even intractable any other way [5].

Quantum Physics – This line of argument suggests that consciousness is essential for true general intelligence, and that consciousness itself is based in quantum-level events. To achieve AI, therefore, will require finding ways to make quantum computing a reality. Although versions of the theory have been worked out in some detail as they might apply to the human case [6], the hypothesis has not been subjected to direct empirical test.

3. Core Values – Just make sure it can do X!

Much of this argument has to do with overall control structure, not specific types of computation. In fact, this approach argues that others are wrongheaded to concentrate on such details. If we just implement the correct central organizing principle everything else will fall into place. Yet such a strong core conceptualization brings its own vulnerabilities. Bad choices about the core principles can be disastrous because so much else builds from this core.

Situatedness – The reason none of our systems have achieved AI is they are not part of the real world – they do not have a gut feel for how interactions occur nor do they have a real stake in the outcome. This topic is concerned with the motivational structure of systems [7] as well as understanding physical and temporal processes.

Emotionality – Here the reasoning goes that emotion is not just a vestigial animal leftover or a mere a by-product of cognition but is instead an essential ingredient [8]. Emotion is crucial to regulating attention, mediating memory formation and retrieval, and arbitrating between actions in uncertain circumstances.

Self-Awareness – As a part of consciousness it is important to be able to recursively access information about one's own states. This gives a sense of a unitary self who deliberately chooses actions and experiences the benefits and costs of their

consequences. It also forms the basis for predicting, imitating, and empathizing with other agents [9].

Hierarchy & Recursion –The ability to abstract from particulars to categorical representations is much more difficult than simple generalization. In fact, the ability to abstract recursively appears to be extremely rare and may even be limited to the human case [10]. The argument is that the most basic feature of general intelligence is a computational mechanism that takes any input, including its own outputs, and finds the pattern of differences and similarities that allow grouping into still more abstract categories [11]. This mechanism gives the data compression needed to produce meaningful but tractable understanding of very complex environments.

4. Emergence – Just add more X!

On this view, we actually have a pretty good grasp of the essentials but we haven't figured out how to implement them at the right scale. If we just add enough knowledge, speed, experience, etc. the system will "magically" come to life. This is a particularly popular mindset currently with the advent of fast processors, large memories, and so much machine-readable content online. Sometimes this strategy works well, as in the Deep Blue chess machine. It had a clever position evaluator, but the bulk of its strength came from a deep search of the game tree. Other times there is too much of an element of unreasoned faith involved. Galvani made a frog's leg twitch using a battery, so imagine (as Mary Shelley did) what a bolt of lightning would do!

Axiomatization – Classical first order logic underpins all human thought. It is a mere matter of identifying and formally codifying all the specific forms of reasoning and then writing the correct axioms for time, space, gravity, emotion, economics, social obligation, self-awareness, etc. There are a lot of these subfields to be encoded and this is the grist necessary for the mill of intelligence. This camp draws supporters from traditional logic backgrounds [12] as well as those working on Qualitative Physics.

Commonsense – This point of view says that we simply need to have the system understand the million or so facts that cover everyday existence. Most reasoning can then be based either directly on this pre-existing corpus, or on minimal extensions through analogy [13].

Learning – It is too hard (or even impossible) to program a creature to react appropriately in all situations. A more robust and flexible approach is to provide guidance about what are good situations versus bad ones and let it learn how to respond itself. All it needs is many time steps of experience in successively less sheltered environments. Reinforcement learning has been particularly successful here [14].

Evolution – This approach posits that the key to AI is self-improving systems. Even if the incremental steps are very small, as long as there is no theoretical bound then the system should be able to bootstrap its way to human-level performance (and beyond!). We just need lots of individuals and generations. Some interesting work has shown that physical structures [15] as well as control algorithms can be evolved.

Integration – A human is not just a brain in a box, it has eyes, ears, arms, legs, etc. How can an AI ever truly appreciate the meaning of a word like "red" without

grounding it in some bodily sensation? We need to put everything we know how to do together in one place and let this creature experience the real physical world. The humanoid robot Cog [16] is one such ambitious attempt, but it is hard to have the best-of-breed technology in all categories simultaneously.

5. Emulation – Just faithfully copy X!

The emulation approach is pessimistic about whether we even have *any* of the proper mechanisms to create intelligence. Instead it advocates that existence proofs be copied. Technology often precedes science, so perhaps we can just re-implement some example in silicon and at least use it. Understanding and good theory can come later. Here simulating even (or especially) the faults of the underlying system is considered a virtue. Yet, since there is no underlying theory, it is hard to tell whether the details being copied are really relevant. For example, artificial feathers and flapping turn out not to be needed to create airplanes.

Neural simulation – All our computer metaphors for the brain may be entirely wrong. We need to simulate, as accurately as possible, the actual neural hardware and see how it responds to various stimuli. Without such detailed modeling (e.g. [17]) we may completely miss key aspects of how humans function.

Neural networks – The human mind presumably is a program that runs on hardware comprised of the human brain. However brains are organized very differently from standard digital computes, so perhaps starting with more a biologically-faithful substrate will make the AI problem easier. Particularly notable are subsymbolic approach to reasoning and language [18].

Animal models – Arguably humans evolved from “lower” animals and genetically the difference is quite small. This suggests that many of the mechanisms and behaviors present in animals underlie human intelligence and that the robust substrate provided by this heritage may be essential for cognition as we understand it. For instance, work on Skinner-bots [19] has shown how a robot can learn to fetch and recycle color objects in much the way a dog would be trained to do the same task.

Human development – How can we expect AI's to spring forth fully competent in several hours when infant development occurs over the course of many years? A lot is known about the various cognitive stages children progress through and it has been suggested that potential AI's follow this same developmental program [20].

Sociality – To be part of a larger cultural entity an AI needs to associate and communicate with other humans and robots [21]. To do this it needs to understand how to effectively participate in social interactions such as advice taking, negotiation, and collaboration. One of the most eye-catching projects here is the robot Kismet [22].

6. Discussion

Grouping things into categories, as in the periodic table of the elements, should serve to predict similar structure among entries in the same region of the table, as well as suggesting that one should observe related responses to various sorts of conditions. For

instance, forgetting for the moment which secret ingredient is being promoted, is there any commonality about the “standard recipe” to which this ingredient will be added? Is it a symbolic substrate or a more diffuse set-based representation? When looking for a central organizing principle and asking the big questions, is the probe modality primarily verbal? Can relevant responses only be elicited in social situations? Similarly, for the emergent camp is there any way to predict how much of a resource will be needed to accomplish one task based on experience with another? Can we tell whether performance will asymptote (perhaps at an unacceptably low level) based on observed incremental improvement with added resource? And for emulation, how do we know whether a model is “faithful enough”? And are there any principles, even vague ones, pervading multiple types of emulation? Perhaps auto-encoders, entropy reduction, or reinforcement are recurring themes.

Taking this ten thousand foot view of the landscape it is even possible that insights gained along one path might have useful implications for another (e.g. the primacy of language, the necessity for task feedback). At the very least there is value in having a big picture view of where progress is being made and where it is stalled [1]. This lets us judge where resources of time and funding ought to be directed, and may be the closest thing available to an optimal search strategy for finding the right path or paths to AI.

References

- [1] I. Lakatos, “Falsificationism and the Methodology of Scientific Research Programmes,” in I. Lakatos and A. Musgrave (eds.), *Criticism and the Growth of Knowledge*, Cambridge University Press, 1965.
- [2] J. Hobbs, M. Stickel, P. Martin, and D. Edwards, “Interpretation as Abduction”, *Proc. Annual Mtg. of the Assoc. for Computational Linguistics*, pp. 95–103, 1990.
- [3] A. Saffiotti, “Fuzzy Logic in Robot Navigation: A Case Study”, Université Libre de Bruxelles, IRIDIA Tech Report 95-25, 1997.
- [4] Michael Dyer, *In-Depth Understanding*, MIT Press, 1983.
- [5] N. Kushmerick, “Software Agents and Their Bodies”, *Minds and Machines*, vol. 7, pp. 227–247, 1997.
- [6] N. Woolf and S. Hammeroff, “A Quantum Approach to Visual Consciousness”, *TRENDS in Cognitive Sciences*, 5(11), pp. 472–478, 2001.
- [7] S. Wilson, “Explore/Exploit Strategies in Autonomy”, *From Animals to Animats* (Proc. SAB-96), 1996.
- [8] Marvin Minsky, *The Emotion Machine*, Simon and Schuster, 2006.
- [9] B. Scassellati, “Theory of Mind for a Humanoid Robot”, *Autonomous Robots*, vol. 12, pp. 13–24, 2002.
- [10] D. Premack, “Is Language the Key to Human Intelligence?”, *Science*, 303(5656), pp. 318–320, 2004.
- [11] Jeff Hawkins, *On Intelligence*, Owl Books, 2005.
- [12] J. McCarthy, “The Well-Designed Child”, <http://www-formal.stanford.edu/jmc/child1.html>, 1999.
- [13] H. Liu and P. Singh, “ConceptNet – A Practical Commonsense Reasoning Toolkit”, *BT Technology Journal*, 22(4), pp. 211–226, 2004.
- [14] L. Lin, “Self-Improving Reactive Agents: Case Studies of Reinforcement Learning Frameworks”, *From Animals to Animats* (Proc. SAB-90 Conf.), pp. 297–305, 1990.
- [15] L. Lichtensteiger and P. Eggenberger, “Evolving the Morphology of a Compound Eye on a Robot”, *Proc. of Eurobot*, pp. 127–134, 1999.
- [16] R. Brooks, C. Breazeal, R. Irie, C. Kemp, M. Marjanović, B. Scassellati, and M. Williamson, “Alternative Essenses of Intelligence”, Proc. of AAAI Conf., pp. 961–968, 1998.
- [17] H. Markram, “The Blue Brain Project”, *Nature Reviews Neuroscience*, vol. 7, pp. 153–160, 2006.
- [18] R. Miikkulainen, “Natural Language Processing with Subsymbolic Neural Networks”, in *Neural Network Perspectives on Cognition and Adaptive Robotics*, A. Browne (ed.), Taylor & Francis, 1997.
- [19] L. Saksida, S. Raymond, and D. Touretsky, “Shaping Robot Behavior Using Principles from Instrumental Conditioning”, *Robotics and Autonomous Systems*, 22 (3/4):231, 1998.
- [20] A. Arsenio, “Children, Humanoid Robots and Caregivers”, *Workshop on Epigenetic Robotics*, 2004.
- [21] E. Herrmann, J. Call, M. Hernández-Lloreda, B. Hare, and M. Tomasello, “Humans Have Evolved Specialized Skills of Social Cognition”, *Science*, vol. 317, pp. 1360–1366, 2007.
- [22] Cynthia Breazeal, *Designing Sociable Robots*, MIT press, 2002.

Adversarial Sequence Prediction

Bill HIBBARD

University of Wisconsin - Madison

Abstract. Sequence prediction is a key component of intelligence. This can be extended to define a game between intelligent agents. An analog of a result of Legg shows that this game is a computational resources arms race for agents with enormous resources. Software experiments provide evidence that this is also true for agents with more modest resources. This arms race is a relevant issue for AI ethics. This paper also discusses physical limits on AGI theory.

Keywords. Sequence prediction, AI ethics, physical limits on computing.

Introduction

Schmidhuber, Hutter and Legg have created a novel theoretical approach to artificial general intelligence (AGI). They have defined idealized intelligent agents [1, 2] and used reinforcement learning as a framework for defining and measuring intelligence [3]. In their framework an agent interacts with its environment at a sequence of discrete times and its intelligence is measured by the sum of rewards it receives over the sequence of times, averaged over all environments. In order to maximize this sum, an intelligent agent must learn to predict future rewards based on past observations and rewards. Hence, learning to predict sequences is an important part of intelligence.

A realistic environment for an agent includes competition, in the form of other agents whose rewards depend on reducing the rewards of the first agent. To model this situation, this paper extends the formalism of sequence prediction to a competition between two agents. Intuitively, the point of this paper is that agents with greater computational resources will win the competition. This point is established by a proof for agents with large resources and suggested by software experiments for agents with modest resources. This point has implications for AI ethics.

1. Sequence Prediction

In a recent paper Legg investigates algorithms for predicting infinite computable binary sequences [4], which are a key component of his definition of intelligence. He proves that there can be no elegant prediction algorithm that learns to predict all computable binary sequences. However, as his Lemma 6.2 makes clear, the difficulty lies entirely with sequences that are very expensive to compute. In order to discuss this further, we need a few brief definitions. N is the set of positive integers, $B = \{0, 1\}$ is a binary alphabet, B^* is the set of finite binary sequences (including the empty sequence), and B^ω is the set of infinite binary sequences. A *generator* g is a program for a universal Turing machine that writes a sequence $w \in B^\omega$ to its output tape, and we write $w =$

$U(g)$. A *predictor* p is a program for a universal Turing machine that implements a total function $B^* \rightarrow B$. We say that a predictor p *learns to predict* a sequence $x_1 x_2 x_3 \dots \in B^\infty$ if there exists $r \in N$ such that $\forall n > r, p(x_1 x_2 x_3 \dots x_n) = x_{n+1}$. Let $C \subset B^\infty$ denote the set of computable binary sequences computed by generators. Given a generator g such that $w = U(g)$, let $t_g(n)$ denote the number of computation steps performed by g before the n^{th} symbol of w is written.

Now, given any computable monotonically increasing function $f: N \rightarrow N$, define $C_f = \{w \in C \mid \exists g. U(g) = w \text{ and } \exists r \in N, \forall n > r. t_g(n) < f(n)\}$. Then Lemma 6.2 can be stated as follows:

Paraphrase of Legg's Lemma 6.2. Given any computable monotonically increasing function $f: N \rightarrow N$, there exists a predictor p_f that learns to predict all sequences in C_f . This is a bit different than Legg's statement of Lemma 6.2, but he does prove this statement.

Lloyd estimates that the universe contains no more than 10^{90} bits of information and can have performed no more than 10^{120} elementary operations during its history [5]. If we take the example of $f(n) = 2^n$ as Legg does, then for $n > 400$, $f(n)$ is greater than Lloyd's estimate for the number of computations performed in the history of the universe. The laws of physics are not settled so Lloyd may be wrong, but there is no evidence of infinite information processes in the universe. So in the physical world it is reasonable to accept Lemma 6.2 as defining an elegant universal sequence predictor. This predictor can learn to predict any sequence that can be generated in our universe. But, as defined in the proof of Lemma 6.2, this elegant predictor requires too much computing time to be implemented in our universe. So this still leaves open the question of whether there exist sequence predictors efficient enough to be implemented in this universe and that can learn to predict any sequence that can be generated in this universe. It would be useful to have a mathematical definition of intelligence that includes a physically realistic limit on computational resources, as advocated by Wang [6].

2. Adversarial Sequence Prediction

One of the challenges for an intelligent mind in our world is competition from other intelligent minds. The sequences that we must learn to predict are often generated by minds that can observe our predictions and have an interest in preventing our accurate prediction. In order to investigate this situation define an *evader* e and a *predictor* p as programs for a universal Turing machine that implement total functions $B^* \rightarrow B$. A pair e and p play a game [7], where e produces a sequence $x_1 x_2 x_3 \dots \in B^\infty$ according to $x_{n+1} = e(y_1 y_2 y_3 \dots y_n)$ and p produces a sequence $y_1 y_2 y_3 \dots \in B^\infty$ according to $y_{n+1} = p(x_1 x_2 x_3 \dots x_n)$. The predictor p wins round $n+1$ if $y_{n+1} = x_{n+1}$ and the evader e wins if $y_{n+1} \neq x_{n+1}$. We say that the predictor p *learns to predict* the evader e if there exists $r \in N$ such that $\forall n > r, y_n = x_n$ and we say the evader e *learns to evade* the predictor p if there exists $r \in N$ such that $\forall n > r, y_n \neq x_n$.

Note that an evader whose sequence of output symbols is independent of the prediction sequence is just a generator (the evader implements a function $B^* \rightarrow B$ but is actually a program for a universal Turing machine that can write to its output tape while ignoring symbols from its input tape). Hence any universal predictor for evaders will also serve as a universal predictor for generators.

Also note the symmetry between evaders and predictors. Given a predictor p and an evader e , define an evader e' by the program that implements p modified to complement the binary symbols it writes to its output tape and define a predictor p' by the program that implements e modified to complement the binary symbols it reads from its input tape. Then p learns to predict e if and only if e' learns to evade p' .

Given any computable monotonically increasing function $f: N \rightarrow N$, define E_f = the set of evaders e such that $\exists r \in N, \forall n > r. t_e(n) < f(n)$ and define P_f = the set of predictors p such that $\exists r \in N, \forall n > r. t_p(n) < f(n)$. We can prove the following analogy to Legg's Lemma 6.2, for predictors and evaders.

Proposition 1. Given any computable monotonically increasing function $f: N \rightarrow N$, there exists a predictor p_f that learns to predict all evaders in E_f and there exists an evader e_f that learns to evade all predictors in P_f .

Proof. Construct a predictor p_f as follows: Given an input sequence $x_1 x_2 x_3 \dots x_n$ and prediction history $y_1 y_2 y_3 \dots y_n$ (this can either be remembered on a work tape by the program implementing p_f , or reconstructed by recursive invocations of p_f on initial subsequences of the input), run all evader programs of length n or less, using the prediction history $y_1 y_2 y_3 \dots y_n$ as input to those programs, each for $f(n+1)$ steps or until they've generated $n+1$ symbols. In a set W_n collect all generated sequences which contain $n+1$ symbols and whose first n symbols match the input sequence $x_1 x_2 x_3 \dots x_n$. Order the sequences in W_n according to a lexicographical ordering of the evader programs that generated them. If W_n is empty, then return a prediction of 1. If W_n is not empty, then return the $n+1^{\text{th}}$ symbol from the first sequence in the lexicographical ordering.

Assume that p_f plays the game with an evader $e \in E_f$ whose program has length l , and let $r \in N$ be the value such that $\forall n > r. t_e(n) < f(n)$. Define $m = \max(l, r)$. Then for all $n > m$ the sequence generated by e will be in W_n . For each evader e' previous to e in the lexicographical order ask if there exists $r' \geq \max(m, \text{length of program implementing } e')$ such that $t_{e'}(r'+1) < f(r'+1)$, the output of e' matches the output of e for the first r' symbols, and the output of e' does not match the output of e at the $r'+1^{\text{th}}$ symbol. If this is the case then this e' may cause an error in the prediction of p_f at the $r'+1^{\text{th}}$ symbol, but e' cannot cause any errors for later symbols. If this is not the case for e' , then e' cannot cause any errors past the m^{th} symbol. Define r'' to be the maximum of the r' values for all evaders e' previous to e in the lexicographical order for which such r' exist (define $r'' = 1$ if no such r' values exists). Define $m' = \max(m, r''+2)$. Then no e' previous to e in the lexicographical order can cause any errors past m' , so the presence of e in W_n for $n > m'$ means that p_f will correctly predict the n^{th} symbol for all $n > m'$. That is, p_f learns to predict e .

Now we can construct an evader e_f using the program that implements p_f modified to complement the binary symbols it writes to its output tape. The proof that e_f learns to evade all predictors in P_f is the same as the proof that p_f that learns to predict all evaders in E_f , with the obvious interchange of roles for predictors and evaders. \square

This tells us that in the adversarial sequence prediction game, if either side has a sufficient advantage in computational resources to simulate all possible opponents then it can always win. So the game can be interpreted as a computational resources arms race.

Note that a predictor or evader making truly random choices of its output symbols, with 0 and 1 equally likely, will win half the rounds no matter what its opponent does.

But Proposition 1 tells us that an algorithm making pseudo-random choices will be defeated by an opponent with a sufficient advantage in computing resources.

3. Software Experiments

Adversarial sequence prediction is a computational resources arms race for algorithms using unrealistically large computational resources. Whether this is also true for algorithms using more modest computational resources can best be determined by software experiments. I have done this for a couple algorithms that use lookup tables to learn their opponent's behavior. The size of the lookup tables is the measure of computational resources. The predictor and evader start out with the same size lookup tables (a parameter can override this) but as they win or lose at each round the sizes of their lookup tables are increased or decreased. The software includes a parameter for growth of total computing resources, to simulate non-zero-sum games. Occasional random choices are inserted into the game, at a frequency controlled by a parameter, to avoid repeating the same outcome in the experiments. The software for running these experiments is available on-line [8].

Over a broad range of parameter values that define the specifics of these experiments, one opponent eventually gets and keeps all the computing resources. Thus these experiments provide evidence that adversarial sequence prediction is an unstable computational resources arms race for reasonable levels of computational resources.

Interestingly, the game can be made stable, with neither opponent able to keep all the resources, by increasing the frequency of random choices. It is natural and desirable that simple table-lookup algorithms should be unable to predict the behavior of the system's pseudo-random number algorithm. But more sophisticated algorithms could learn to predict pseudo-random sequences.

The adversarial sequence prediction game would make an interesting way to compare AGI implementations. Perhaps future AGI conferences could sponsor competitions between the AGI systems of different researchers.

4. AI Ethics

Artificial intelligence (AI) is often depicted in science fiction stories and movies as a threat to humans, and the issue of AI ethics has emerged as a serious subject [9, 10, 11]. Yudkowsky has proposed an effort to produce a design for AGI whose friendliness toward humans can be proved as it evolves indefinitely into the future [12]. Legg's blog includes a debate with Yudkowsky over whether such a proof is possible [13]. Legg produced a proof that it is not possible to prove what an AI will be able to achieve in the physical world, and Yudkowsky replied that he is not trying to prove what an AI can achieve in the physical world but merely trying to prove that the AI maintains friendly intentions as it evolves into the indefinite future. But intentions must be implemented in the physical world, so proving any constraint on intentions requires proving that the AI is able to achieve a constraint on the implementation of those intentions in the physical world. That is, if you cannot prove that the AI will be able to achieve a constraint on the physical world then you cannot prove that it will maintain a constraint on its intentions.

Adversarial sequence prediction highlights a different sort of issue for AI ethics. Rather than taking control from humans, AI threatens to give control to a small group of humans. Financial markets, economic competition in general, warfare and politics include variants of the adversarial sequence prediction game. One reasonable explanation for the growing income inequality since the start of the information economy is the unstable computational resources arms race associated with this game. Particularly given that in the real world algorithm quality is often an important computational resource. As the general intelligence of information systems increases, we should expect increasing instability in the various adversarial sequence prediction games in human society and consequent increases in economic and political inequality. This will of course be a social problem, but will also provide an opportunity to generate serious public interest in the issues of AI ethics.

References

- [1] J. Schmidhuber. The Speed Prior: A New Simplicity Measure Yielding Near-Optimal Computable Predictions. In J. Kivinen and R. H. Sloan, editors, Proceedings of the 15th Annual Conference on Computational Learning Theory (COLT 2002), Sydney, Australia, Lecture Notes in Artificial Intelligence, pages 216--228. Springer, 2002. <http://www.idsia.ch/~juergen/coltspeed/coltspeed.html>
- [2] Hutter, M. *Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability*. Springer, Berlin, 2004. 300 pages. <http://www.idsia.ch/~marcus/ai/uaibook.htm>
- [3] Hutter, M. and S. Legg. Proc. A Formal Measure of Machine Intelligence. *15th Annual Machine Learning Conference of Belgium and The Netherlands (Benelearn 2006)*, pages 73-80. <http://www.idsia.ch/idsiareport/IDSIA-10-06.pdf>
- [4] Legg, S. Is there an Elegant Universal Theory of Prediction? Technical Report No. IDSIA-12-06. October 19, 2006. IDSIA / USI-SUPSI Dalle Molle Institute for Artificial Intelligence. Galleria 2, 6928 Manno, Switzerland. <http://www.idsia.ch/idsiareport/IDSIA-12-06.pdf>
- [5] Lloyd, S. Computational Capacity of the Universe. *Phys.Rev.Lett.* 88 (2002) 237901. <http://arxiv.org/abs/quant-ph/0110141>
- [6] Wang, P. Non-Axiomatic Reasoning System --- Exploring the essence of intelligence. PhD Dissertation, Indiana University Comp. Sci. Dept. and the Cog. Sci. Program, 1995. <http://www.cogsci.indiana.edu/farg/peiwang/PUBLICATION/wang.thesis.ps>
- [7] http://en.wikipedia.org/wiki/Game_theory
- [8] <http://www.ssec.wisc.edu/~billh/g.asp.html>
- [9] Hibbard, W. Super-Intelligent Machines. *Computer Graphics* 35(1), 11-13. 2001. <http://www.ssec.wisc.edu/~billh/visfiles.html>
- [10] Bostrom, N. Ethical Issues in Advanced Artificial Intelligence. *Cognitive, Emotive and Ethical Aspects of Decision Making in Humans and in Artificial Intelligence, Vol. 2*, ed. I. Smit et al., Int. Institute of Advanced Studies in Systems Research and Cybernetics, 2003, pp. 12-17. <http://www.nickbostrom.com/ethics/ai.html>
- [11] Goertzel, B. Universal Ethics: The Foundations of Compassion in Pattern Dynamics. October 25, 2004. <http://www.goertzel.org/papers/UniversalEthics.htm>
- [12] Yudkowsky, E. (2006) Knowability of FAI. <http://sl4.org/wiki/KnowabilityOfFAI>
- [13] Legg, S. Unprovability of Friendly AI. September 15, 2006. <http://www.vetta.org/?p=6>

Artificial General Intelligence via Finite Covering with Learning

Yong K. Hwang, Samuel B. Hwang and David B. Hwang

Think Life, Inc.

Palo Alto, CA, USA

Abstract. This position paper claims that the combination of solutions to a finite collection of problem instances and an expansion capability of those solutions to similar problems is enough to achieve the artificial general intelligence comparable to the human intelligence. Learning takes place during expansion of existing solutions using various methods such as trial and error, generalization, case-based reasoning, etc. This paper also looks into the amount of innate problem solving capability an artificial agent must have and the difficulty of the tasks the agent is expected to solve. To illustrate our claim examples in robotics are used where tasks are physical movements of the agent and objects in its environment.

Keywords. human intelligence, machine learning, knowledge expansion, robotics

Introduction

Achieving human-level intelligence has been an illusive goal due to the vast amount of knowledge humans build up and use to solve many different tasks. Since it is impossible to preprogram solutions to all the problems an agent will face in the real world, many attempts are made to endow the agent with learning capabilities so that it can increase its knowledge base from a relatively small knowledge base. This approach has some success in a well controlled environment, but cannot yet handle complex problems arising in the real-world situations like humans can.

Although humans can solve many problems, they are functional only in familiar environments. If a human is put in a world with irregular gravity he will have a hard time navigating the world. It is our claim that the kinds of worlds humans can function efficiently are not that many compared to the set of all possible variations of the world. From this claim follows that a successful AGI can be built with a finite number of experiences. A positive (negative) experience is a pair of a problem and a successful (unsuccessful) action for it. One more requirement is capability of generalizing or expanding an experience to similar situations. The resulting knowledge base is then capable of handling a subset of all possible worlds, where the subset is a union of blobs around the points representing positive experiences minus the blobs representing negative experiences (see Fig. 1). This subset is a finite covering of the world in which the AGI system can function well to suit its purposes and avoid failures.

This paper concentrates on the task planning in the physical world, which occupies a middle of the intelligence hierarchy [1]. We now present the process of building up a knowledge base by expansion capability to generate a finite covering of the world. Examples in robotics are used to illustrate this process.

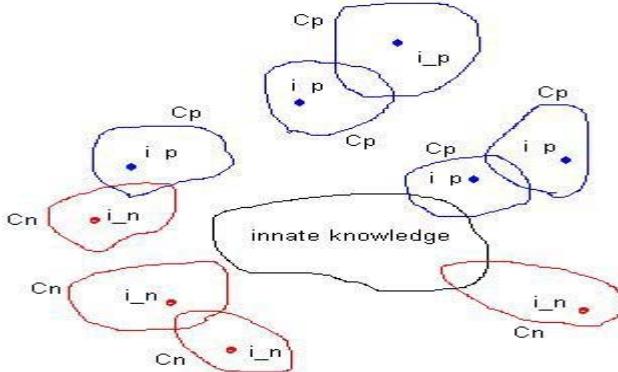


Fig. 1. The knowledge set for solving problems is the union of neighborhoods of positive experiences while the knowledge about failure is that of negative experiences.

1. Knowledge Representation

Our agent starts out with an innate knowledge, and expands it via generalization and learning. Our main claim is that an adequate knowledge base for agent's well being can be built as a *finite*, i.e., *manageable*, union of problem-solution pairs and that of problem-failure pairs. A problem, q , is defined as the pair of agent's current and desired states of the agent and its environments, $q(e_{current}, e_{desired})$. A positive experience instance, i_p is a pair of a problem and an agent's action resulting in a success, written as $i_p(q, a_p)$. Similarly, a negative experience instance is written as $i_n(q, a_n)$. When the agent has a capability to recognize a problem, $q\sim$, similar to q and a capability to modify the action a_p to solve $q\sim$, the set of problems near q solvable by the agent form a set covering q , written as $Cp(i_p)$. Similarly, the set of problem instances around i_n resulting in an undesirable outcome is called $Cn(i_n)$. Let Kp and Kn be the knowledge sets of the positive experience and negative experience, respectively. Kp (Kn) is a union of Cp 's (Cn 's). Then the set of problem instances solvable by the agent with 100% confidence is $Kp - Kn$. If we are willing to accept some failure, the agent might try to solve problems belonging to both Kp and Kn .

Let us give an example. In robotics field, two of the fundamental tasks are robot navigation and object manipulation. In navigation, the problem is to find a short, collision-free path from the current configuration of the robot to the desired configuration. The $e_{current}$ ($e_{desired}$) is specified by the current (desired) robot configuration along with configurations of all objects. In manipulation, $e_{current}$ ($e_{desired}$) is specified by the current (desired) configuration of the object to be manipulated along with those of nearby objects. Suppose the robot has a path-planning algorithm, $a_{pathplan}$ to find a collision-free path. An arrangement of objects in its environment and robot's start/goal configuration form a problem instance $q(e_{start}, e_{goal})$. If $a_{pathplan}$ successfully finds a solution path, then the pair $(q, a_{pathplan})$ forms a positive experience, i_p . Otherwise, it forms a negative experience, i_n . If in the same environment an object irrelevant to the solution path is moved away from the path, then $a_{pathplan}$ will succeed again. Then the new experience forms a part of the positive covering around i_p , namely $Cp(i_p)$.

2. Expansion of Knowledge

As an agent solves problems, it expands its knowledge base in several ways. The trial-and-error method is useful when the penalty for failure is small. Another method is to learn from a teacher, whether a human or other agents. Other machine learning algorithms can be used to expand knowledge [2]. These processes increase the sets of positive and negative experiences whose sizes are of measure zero, i.e., they are just points in the space of all problems. If we have some local expansion capability, the agent can grow the point experiences into sets of finite measures, vastly increasing the size of the problems it can solve. This expansion capability enables us to claim that not infinite, but only a finite number of experiences are enough for agent's successful life. The increase of knowledge occurs in two categories: by solving new problems, and by expanding an existing piece of knowledge for a particular problem to similar problems. Let's call these global and local learning. These are elaborated below.

2.1. Global Learning

Simply put, a global learning takes place when the agent learns a *brand new* stuff. Suppose the agent learns to use a knife to cut objects, which it does not know how before. This is a kind of global learning. Another example is learning to walk in a stair case when the agent could walk only on a flat floor. When a global learning occurs, inserted in the knowledge base is a new pair, (p, a) , of a problem, p , represented as the current and desired environmental and an algorithm, a , to solve the problem. A global learning can be achieved by various means. A trial-and-error method is simple but may have a low success rate. A more prudent way is to learn from a teacher, which can be humans or observation of successful task completion by other agents. Another method is divide-and-conquer scheme which decomposes a task into subtasks that are solvable by existing algorithms. Learning the category of grasp such as power grip or precision grip [3] can also be classified as global learning. In terms of our notation the global learning increases the number of coverings Cp 's or Cn 's in the agent's knowledge base.

2.2. Local Learning

A local learning occurs when a new problem is solved by an existing algorithm with substitution of input parameters. For paper cutting task, for example, if the agent learns to use the algorithm of using scissors for cutting a string then a local learning has taken place (substitution of object). Local learning can be achieved by various methods. A trial-and-error method can be used to apply an existing algorithm to tasks with different objects. Or the case-based reasoning can be used to recognize similar situations and the agent can use the action that succeeded. Local learning increases the size of covering, Cp or Cn . If the algorithm solves (fails to solve) the problem, the size of Cp (Cn) is increased. The key difference between global and local learning is whether a new algorithm is introduced in the knowledge base.

2.3. Quantum Learning

The most challenging type of learning is the automatic generation of new algorithms based on existing knowledge. It is at a higher level of learning than global learning, and

for this reason we call it quantum learning. Suppose our agent know about depth-first and breath-first search algorithms. How do you make the agent generate A* search on its own? We have to somehow embed in the knowledge the principles of “best first” exploration, “possible option” retention and “hopeless option” deletion. One way of quantum-learning is maintaining many small knowledge pools as in Society of Mind [4], where a particular combination of pools forms a new algorithm. This is not in the scope of our paper, and we are just pointing out the difficulty of quantum learning.

3. Complexity of Building Artificial General Intelligence

We explore the amount of knowledge required to build human-level AGI. We don’t claim our list to be anywhere near a complete set, but it will give a good starting point for necessary components required by AGI systems. We first give a set of physical principles that reduces the amount of required innate knowledge and the complexities of the algorithms for problem solving. Presented next is the minimal requirement on the components of the innate knowledge from which to build a comprehensive AGI.

3.1. Physical Principles

We have come up with several principles based on the physical laws that can be used to simplify the complexity of AGI. They are proximity-in-space (PIS), proximity-in-time (PIT) and benign-environment (BE) principle. The PIS was used in [5] for generating natural-language-like robot commands. These principles are explained in detail below.

Let us explain by analogy how PIS makes problems in the physical world simpler in practice. A lone ninja fighting against many enemy fighters needs to deal with only 4 or so enemies at a time because the rest of enemies cannot get to the ninja. By the same token, although it is theoretically true that all objects need to be considered when planning collision-free motion in navigation or manipulation tasks, only several nearby relevant objects need to be taken account. Nailing two pieces of wood together requires consideration of only the wooden pieces, a nail and a hammer.

The PIT assumes that events relevant to the current problem occur in proximal time unless explicitly specified by some means. One of the basic assumptions in human reasoning is that things do not change too fast. If they do, for example, such as in a car accident, humans also make mistakes. When planning an action, it is prudent to consider events occurred in the near past or going to happen in the near future with more emphasis. To solve a crime, for example, detectives investigate people who visited the crime scene near the time of crime. The PIT, therefore, can greatly simplify the complexities of planning algorithms necessary for human-level intelligence.

The BE asserts that the environment is mostly benign and is not there to make the agent fail. Humans set up their habitat with favorable conditions of weather, food resources, etc. Furthermore, they build their houses so that everyday tasks can be performed with ease. We also assume objects are safe to interact with unless explicitly specified, such as “fire is hot.” Also, most knives are made with a handle to facilitate safe grasp. The default assumption of safety greatly reduces the amount of knowledge to be stored. This is true for many other object properties such as strength, weight, etc. Another example is the existence of hallways in buildings to expedite navigation. Yet another example is the fact that most tasks humans do can be performed with one or two hands. The development of AGI, thus, should be easier for manmade environments.

3.2. Components and Size of Knowledge Base

Humans are visionaries, i.e., their primary sensor is the eyes. To build an agent that interacts with humans, the agent must have an ensemble of good vision algorithms. Tactile sensing is of a great help in object manipulation. To recognize similar situations and similar objects, a similarity recognizer is needed. To detect changes in objects or environments, a difference operator is needed. Search algorithms are also required to find optimal solutions to various tasks. They include findpath, findplace (to put an object), findgrasp (to grab an object) to name a few. To predict the effect of an action during task planning, a qualitative simulator [6] is also necessary. Also needed are a database to store object properties and many other information such as pre- and post conditions of actions, goodness evaluation functions of situations/actions, and so on.

Now the important question is “how big is the knowledge base that is sufficient for AGI?” For home-service robots, 1000 actions seem sufficient [7]. The number of objects involved in an action is small, like 4, according to BE. Given a task, the action planning process involves a search in the space of agent actions. If we provide a template of an action sequence for each task, a search is triggered to find an alternate action only when one of the actions in the task template results in a failure. One of the main problems with the traditional reasoning system is the combinatorial explosion. A counter argument to this is the small-world phenomenon or six degrees of separation. It states that there is a relatively short chain of acquaintances to link two people in the world. By analogy, we claim that there is a short chain of actions to go from the current to a desired situation. Or relevant objects to solving a problem are linked to the agent or current situation in a few levels of depth in the search tree for a situation.

4. Conclusions

We claim in this paper that there is a hope for building a human-level AGI. This is conjectured from the fact that 1) even humans are not functioning effectively in an unfamiliar environment, and 2) by employing knowledge expansion methods a finite number of experiences can be beefed up to solve enough problems for an artificial agent. It remains to be seen that our claim is true by actually developing an AGI system. It is our belief that a system with human-level AGI is in the near future.

References

- [1] M. Minsky, P. Singh, and A. Sloman, The St. Thomas common sense symposium: designing architectures for human-level intelligence, *AI Magazine*, Summer 2004, 25(2):113-124.
- [2] T. Mitchell, *Machine Learning*, McGraw-Hill Education (ISE Editions), 1st edition, October 1997.
- [3] M.R. Cutkosky, “On grasp choice, grasp models and the design of hands for manufacturing tasks,” *IEEE Trans. On Robotics and Automation*, vol. 5, no. 3, pp. 269-279, June 1989.
- [4] M. Minsky, *Society of Mind*, Simon & Schuster, March 1988.
- [5] P.C. Chen and Y.K. Hwang, “Generating natural language-like robot motion commands through machine learning,” *IEEE Int. Conf. on System, Man and Cybernetics*, vol. 1, pp. 114-121, Oct. 1994.
- [6] B. Kuiper, 2001. Qualitative simulation. In R. A. Meyers (Ed.), *Encyclopedia of Physical Science and Technology*, Third Edition, NY: Academic Press.
- [7] Y.K. Hwang, M. Lee and D. Lee, “Robots’ role in the ubiquitous computing household environment,” *Proceedings of the International Symposium on Robotics*, Paris, France, March 2004.

Cognitive Primitives for Automated Learning

Sudharsan IYENGAR

Winona State University, Winona, MN, USA

siyengar@winona.edu

Abstract. Artificial Intelligence deals with the automated simulation of human intelligent behavior. Various aspects of human faculties are tackled using computational models. It is clear that brain as a cognitive machine is significantly different from a computer. A general intelligent machine needs to incorporate primitives that are identical or similar to those intrinsic to human. We argue for the identification, verification, and development of technologies that will deliver core primitives that are fundamental to human cognition that are applicable to various domains. Mnemonics for aural and for visual cognition are presented in [2, 3]. We propose a generalized set of cognitive primitives that can be specialized to various applications viz. a) Existential, b) Structured Recurrence, c) Relative Attachment, d) Harmonized Counterparts, e) Cordial Counterparts and e) Discriminate Counterparts.

Key words: Science of intelligence, brain model, cognition, primitives.

Introduction

Science of Intelligence [1] involves the basic theory and systems that involve understanding brain structure, cognitive processes, artificial intelligence (computational) models, and others. Science of the brain explores the physical and analytical aspects of brain and aims to study the principles and model of natural intelligence at a molecular level. Cognitive Science studies human mental activity, such as perception, learning, memory, thinking, consciousness etc. In order to model and implement machine intelligence, artificial intelligence attempts simulation, extension and expansion of human intelligence using artificial machines – specifically using computers. These disciplines work together to explore new concepts, new theory, and new technologies and methodologies to create and implement models that can imitate human intelligence.

What is learning? Cognition is a process of acquiring knowledge or skill. Acquiring and assimilation of information and knowledge is inherent to learning. But what constitutes knowledge and/or skill? As one assimilates information it is unclear if it is knowledge, relevant knowledge, or ability (skill) or pertinent skill. An alternate definition for learning can be: A lasting change in cognition resulting from experience, potentially directly influencing behavior. So what is cognition? Mental functions such as the ability to think, reason, and remember. Experiences include external inputs and its processing. Behavior is a processed response to an input. Thus we could say learning is a process through which external inputs enable acquiring knowledge and/or skill that affect the state of an organism such that its subsequent behavior may be altered.

Many have addressed the ways and means by which humans receive external input. Aural and visual inputs are the ones that are studied significantly. Natural language processing addresses the methods and technology of recognizing sounds and applying linguistics to recognize information. These techniques are further applied for linguistic responses. Image processing and pattern recognition aim to recognize objects and concepts, from optical input and images, using stored entities and processing techniques.

Machine learning is to enable a computer system to autonomously acquire and integrate knowledge. The capacity to learn from experience, analytical observation, and other means, is expected to result in a system that can self-improve and thus achieve better performance.

The technology with which input is 'scanned', converted, and subsequently used varies with application and models used for storing and matching the resident knowledge. Applications use algorithmic methods that result in accurate or approximate identification.

Stored knowledge use raw images and databases, Petri-nets, Bayesian networks, connectionist networks, and logical relational knowledge base. Various processing methods are used that integrate the content from the knowledge base and the rules used by experts in solving problems intelligently. Logic programming, production systems, neural networks, and other tools are used for computing so as to arrive at a desired solution. Techniques used vary from case-based reasoning, decision tree learning, meta learning, regression and statistical approaches.

What has eluded researchers is the ability to completely and accurately embody software solutions with techniques so easy and natural to humans. And, every domain of application presents its own challenges – whereas humans are able to seamlessly handle various domains with ease. Additionally, humans tackle indeterminate and fuzzy input and still achieve great degree of dependable solutions.

Applications software are 'trained' on test cases devised and labeled by humans, scored so as to estimate its usefulness, and then tested on real-world cases. The results of these real-world cases are in-turn reflected upon by human developers to further train the software in an attempt to improve its effectiveness.

We need to develop a framework that embodies human problem solving characteristics and then utilize those for developing 'software' solutions emulating intelligence. One could argue that newer technology and 'software' development methods different than current one may need to be developed. So be it.

1. Cognition: How, Why and What?

We believe that in order to enable cognition we need pertinent set of core primitives. Every living organism is enabled with basic primitives. The usage and application of these with various degrees of flexibility, interdependence, and variations achieve cognition and which affect subsequent response(s). Cognitive functions are thus the result of concerted application of these core primitives.

Learning requires cognition as a primary step. Different organisms possess various cognitive abilities and various levels of these abilities. This in-turn is intrinsically involved with the input-output mechanism enabled in the organism.

Cognitive abilities involve aural and visual input and integrate these with analytical processing of this input. Language, music, and other forms of sound based communication use aural aspects. Input is received through the ear (or other forms of sound capture) and output could be voice generated. A systematic methodology for generating aural chants is proposed and shown capable of generating a variety of 'musical' sequences in [4]. Object, face, and scene recognition (static or dynamic) require visual input (eyes). A simple visual perception model is presented in [5]. Similar model is used in [6]. Response may be aural and/or physical - movement reconfiguring of self. Cognition of analytical concepts involves a re-evaluation and re-organization of internal knowledge to arrive at a verifiable and applicable new knowledge.

The primary aim of cognition in humans and animals is to effectively interact, communicate, and create. Primal instinct as well as intended actions enable an organism to interact and communicate. Intelligence - or semblance thereof - is firmly a consequence of what is cognizant to the organism and what has been assimilated through its experiences. Additionally, the ability of the organism to inclusively, and effectively, utilize the imbibed knowledge towards intended communication and interaction reflect its intelligence.

2. Need for Intelligence Primitives

In May 1997, IBM's Deep Blue Supercomputer played a fascinating match with the reigning World Chess Champion, Garry Kasparov. IBM scientists have taken pains to emphasize that Deep Blue is just a glorified calculator. Whether a machine like Deep Blue, which combines lightning-fast search power with a large database of knowledge, can be said to 'think intelligently' depends on one's philosophical inclinations. From the perspective of a Turing test - Deep Blue could be construed to be intelligent. But the methods and techniques used in playing the game of chess by Kasparov and by Deep Blue were far from being similar. Deep Blue utilized brute computational methods involving symbolic storage and matching algorithms. It could manipulate a given input, from the move made by Kasparov, and suggest an output which was then transmitted to the game. Kasparov on the other hand recognized a move on the board, and arrived at the next move based on experience, abstracted and short-cut techniques, and other pre-planned moves.

How do we define intelligence and how can we decide who or what has it? This remains among science's unsolved and possibly unsolvable, problems. The model and the methods applied by Kasparov significantly differ from the techniques used by Deep Blue. What are those? Are they made up of big blocks of intelligent capabilities or are they made up of simpler elements collectively reflecting higher order capability?

We believe that cognition, as well as intelligence, is a result of a bottom-up constructivist model in which a core set of capabilities are integrated into different models so as to achieve higher cognitive

capabilities. A simple analogy is the use of Boolean primitive constructs (AND, OR, and NOT) to create higher order artifacts and circuits, and ultimately a microprocessor. An enormous amount of effort by a generation of engineers and scientists has contributed to this technological marvel. So also, we believe that a versatile and capable intelligent machine can be built only from ground-up using core cognitive primitives – implementable using current or newer technologies.

Cognition, and learning, occurs in two modes. Incidental cognition, and learning, is an involuntary phenomenon where, through innate ability, an organism involuntarily receives input and assimilates this information. When we passively listen to a song and are then able to recognize it later - we have undergone incidental cognition. Incidental cognition might occur numerous times. Significant amount of cognition occurs this way throughout the life of an organism.

Intentional cognition is a process consciously applied involving input and a deliberated output for verification and authentication. Repetitive with corrective action is typical in intentional cognition. When we learn to write a letter or symbol or learn to play catch we employ intentional cognition.

In either case the cognitive process is incremental in nature. One recognizes a piece of a song after having recognized, multiple times, smaller parts of that song. Finally the entire song is recognized as an aggregation of the smaller pieces. Cognition utilizes the ability of recognize 'known' elements and its 'presence' in a larger context.

We propose a hierarchy of cognitive capabilities that can be utilized constructively to develop higher order capabilities. These primitives do not define a complete set needed for constructing intelligent systems. But any system that needs to be constructed should be based on capabilities, constructed using the primitives, pertinent to the intended domain.

3. Cognitive Primitives

We propose generic mnemonic or cognitive capabilities. Hierarchical, inclusion of and use of these primitives pertinent to different domains can enable systems/organisms to imbibe and recognize elements from its input.

Definition 1: An element or item, i , is an input that is aural, visual, or physical.

Repeated input of the item to the system results in the system imbibing it. The number of repetition can be system specific and once a threshold number of repetition occurs, the item is assimilated for further use. An input is recognized if it matches an already assimilated item matched by a process of decomposition. Once an item has been abstracted (imbibed) into the consciousness (or knowledge base) of a system, subsequent recognition of this item is atomic.

Definition 2: A component, c , is a composition of atomically recognizable elements.

Input to the system can be a complex combination or steam of elements. Recognition of such complex input is a result of recognition of its elements and their combinations.

Definition 3: The granularity of a component c , $g(c)$, is the number of elements used in recognizing it through decomposition to match known elements.

Definition 4: The size of a component c , $z(c)$, is the minimum granularity of c in recognizing it.

Definition 5: The expressiveness of a component c , $x(c)$, is the set of elements used in recognizing it.

Definition 6: The shape of a component, c , is the minimum $x(c)$ of c , and is denoted $s(c)$.

Definition 7: Two components, $C1$ and $C2$, are said to be balanced iff $s(C1) = s(C2)$.

Definition 8: Two components, $C1$ and $C2$, are said to be discordant iff. $s(C1) \neq s(C2)$ and $s(C1) \cap s(C2) \neq \{\}$.

Definition 9: Two components, $C1$ and $C2$, are said to be distinct iff $s(C1) \cap s(C2) = \{\}$.

We first propose primitives that enable a system to recognize elements and recognize simple organization of known elements.

Primitive 1 - Existential (E): The ability to be cognizant of the existence of an item.

Primitive 2 - Structured Recurrence (SR): The ability to recognize a simple recurrence of an item.

Primitive 3 - Relative Attachment (RA): The ability to recognize relative juxtaposition of two items.

E(*i*) implies an atomic cognition of *i* using **E**. **SR(*i*)** implies the atomic cognition of a cogent recurrence of *i*. For aural input the recurrences are beat synchronized recurrence. For visual input the recurrences are simple spatial adjacencies.

RA(*i, rj, j*) implies the recognition that items *i* and *j* are placed per **rj** - a specified order of connection or join. For aural inputs **rj** is a specific time ordered sequencing. For visual input it can be a specific relative spatial placement of the items.

E, SR, and RA are fundamental primitives that recognize inputs based on elements that are atomically recognized without resorting to partitioning of input into various components.

When input is complex then these can possibly be recognized as an arrangement of components that can be recognized individually. Components may be a result of simple or complex combinations. Primitives or recognizing the combinations are proposed as.

Primitive 4 - Harmonized Counterparts (HC): The ability to recognize associated balanced components.

Primitive 5 - Cordial Counterparts (CC): The ability to recognize associated cordial components.

Primitive 6 - Discriminate Counterparts (DC): The ability to divide and recognize two distinct components.

HC, CC, and DC are primitives that enable the recognition by dividing input into one or more recognizable components. These primitives assist in partitioning the input such that the commonality among the components is distinguished.

4. Capable Systems with Intelligence

A system capable only with **E** can only recognize the existence or absence of an item. A system capable of **E** and **SR** can recognize simple recurrence of a known item. Thus it could have the ability to recognize a linear arrangement of visual element or a chronological repetition of a note. Thus this system possesses a higher level of cognitive capacity. Systems that are additionally capable of **RJ** recognize the placement of two known items in distinctive relative position. Thus, for visual input it may recognize two inputs conjoined end to end, or top to bottom, co-centric, or some other distinctive arrangement. For aural inputs it could be one element before other, played simultaneously, or some time delayed occurrence.

HC is a capability that helps a system to quickly identify components composed of identical elements. This represents a capacity to drive recognition of components that are possibly balanced or identical. **CC** enables a system to identify components that are similar but does note differences. These differences are meaningful distinguishing characteristics. **DC** assists the system to partition input into smaller components to enable easier recognition. This process helps cognition by a process of divide-and-conquer.

The following table shows a possible set of types of Learning Systems (**LS**) based on a hierarchical set of capabilities they possess.

Learning System	Enabled Capabilities
LS(I)	{E, SR, RA}
LS(II)	{E,SR,RA, HC}
LS(III)	{E,SR,RA, HC, CC}
LS(IV)	{E,SR,RA, HC, DC}

5. Conclusion

We have proposed and defined fundamental mnemonic capabilities applicable for cognition. Mnemonic capabilities enable a system/organism to atomically match input with learned knowledge. The ability to record experiences as part of its history and the ability to abstract/imbibe knowledge from this history of experience are also fundamental to a cognitive system. Using these capabilities learning systems are proposed that are capable of various degrees of cognition. A constructivist approach forms the guiding principle for our paper. Based on these we have developed mnemonics for aural perception in [2] and for visual perception in [3].

6. Current and Future work

We have constructed a music formulating (COMPOSE) systems based on the capabilities customized for aural cognition [4]. This system enables a naïve formulation of symphony using simple combination of primitives developed in [2]. Survey on the use of these capabilities, have shown the affinity and use of these musical structures [5]. A theory on natural language formulation is presented in [6] that utilize aural mnemonics. Similar studies using proposed visual mnemonics will be undertaken and the results of which subsequently used in simple recognition software. We will be developing systems that can be used to test these primitives for visual cognition

References

1. <http://www.stormfront.org/racediff/mainstre.html>
2. A Theory on the Abstraction and Cognition Based on Musical Note Patterns, Sudharsan Iyengar, ICAI-04, Las Vegas, June 2004.
3. Mnemonics for Visual Perception, submitted ECAL 2007, Sudharsan Iyengar, Lisbon, September 2007
4. A Framework for Generating Chants Using Mnemonic Capabilities, Sudharsan Iyengar, SMC06, Marseille, May 2006.
5. A simple visual perception model by adaptive junction, Ajioka, Y. Inoue, K., IJCNN, 1992.
6. Visual Perception Modeling for Intelligent Avatars, Ronghua Liang, et. al, Advances in Artificial Reality, LNCS, 4282/2006.

Vector Symbolic Architectures: A New Building Material for Artificial General Intelligence¹

Simon D. LEVY ^{a,2}, and Ross GAYLER ^b

^a Washington and Lee University, USA

^b Veda Advantage Solutions, Australia

Abstract. We provide an overview of Vector Symbolic Architectures (VSA), a class of structured associative memory models that offers a number of desirable features for artificial general intelligence. By directly encoding structure using familiar, computationally efficient algorithms, VSA bypasses many of the problems that have consumed unnecessary effort and attention in previous connectionist work. Example applications from opposite ends of the AI spectrum – visual map-seeking circuits and structured analogy processing – attest to the generality and power of the VSA approach in building new solutions for AI.

Keywords. Vector Symbolic Architectures, associative memory, distributed representations, Holographic Reduced Representation, Binary Spatter Codes, connectionism

Introduction

Perhaps more so than any other sub-field of computer science, artificial intelligence has relied on the use of specialized data structures and algorithms to solve the broad variety of problems that fall under the heading of “intelligence”. Although initial enthusiasm about general problem-solving algorithms [1] was eventually supplanted by a “Society of Mind” view of specialized agents acting in concert to accomplish different goals [2], the dominant paradigm has always been one of discrete atomic symbols manipulated by explicit rules.

The strongest challenge to this view came in the 1980’s, with the emergence of connectionism [3], popularly (and somewhat misleadingly) referred to as neural networks. In contrast to the rigid, pre-specified solutions offered by “Good Old-Fashioned AI” (GO-FAI), connectionism offered novel learning algorithms as solutions to a broad variety of problems. These solutions used mathematical tools like learning internal feature representations through supervised error correction, or back-propagation [4], self-organization of features without supervision [5], and the construction of content-addressable memories through energy minimization [6]. In its most radical form, the connectionist ap-

¹The authors thank Chris Eliasmith, Pentti Kanerva, and Tony Plate for useful discussion on the topics presented here, and three anonymous reviewers for helpful suggestions.

²Corresponding Author: Simon D. Levy, Computer Science Department, Washington and Lee University, Lexington, Virginia 24450 USA; E-mail: levys@wlu.edu.

proach suggested that the structure assumed by GOFAI and traditional cognitive science might be dispensed with entirely, to be supplanted by general mechanisms like sequence learning [7]. In support of the representations derived by such models, proponents cited the advantages of distributing informational content across a large number of simple processing elements [8]: distributed representations are robust to noise, provide realistically “soft” limits on the number of items that can be represented at a given time, and support distance metrics. These properties enable fast associative memory and efficient comparison of entire structures without breaking down the structures into their component parts.

The most serious criticism of connectionism held that neural networks could not arrive at or exploit systematic, compositional representations of the sort used in traditional cognitive science and AI [9]. Criticism that neural networks are in principle unable to meet this requirement was met in part by compositional models like RAAM [10]; however, RAAM’s reliance on the back-propagation algorithm left it open to criticism from those who pointed out that this algorithm did not guarantee a solution, and could be computationally intractable [11].³

In the remainder of this paper, we describe a new class of connectionist distributed representations, called Vector Symbolic Architectures (VSA), that addresses all of these concerns. VSA representations offer all of the desirable features of distributed (vector) representations (fast associative lookup, robustness to noise) while supporting systematic compositionality and rule-like behavior, and they do not rely on an inefficient or biologically implausible algorithm like back-propagation. The combination of these features makes VSA useful as a general-purpose tool or building material in a wide variety of AI domains, from vision to language. We conclude with a brief description of two such applications, and some prospects for future work.

1. Binding and Bundling in Tensor Products

Systematicity and compositionality can be thought of as the outcome of two essential operations: binding and bundling. Binding associates fillers (*John, Mary*) with roles (*LOVER, BELOVED*). Bundling combines role/filler bindings to produce larger structures. Crucially, representations produced by binding and bundling must support an operation to recover the fillers of roles: it must be possible to ask “Who did what to whom?” questions and get the right answer.

Vector Symbolic Architectures is a term coined by one of us [13] for a general class of distributed representation models that implement binding and bundling directly. These models can trace their origin to the Tensor Product model of Smolensky [14]. Tensor-product models represent both fillers and roles as vectors of binary or real-valued numbers. Binding is implemented by taking the tensor (outer) product of a role vector and a filler vector, resulting in a mathematical object (matrix) having one more dimension than the filler. Given vectors of sufficient length, each tensor product will be unique. Bundling can then be implemented as element-wise addition (Figure 1), and bundled structures can be used as roles, opening the door to recursion. To recover a filler (role) from a bundled tensor product representation, the product is simply divided by the role (filler) vector.

³Recent results with a linear version of RAAM using principal component analysis instead of back-prop [12] show promise for overcoming this problem.

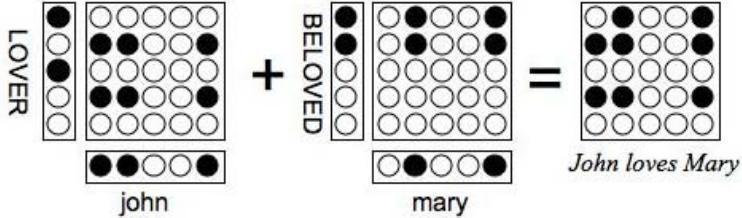


Figure 1. Tensor product representation of *John loves Mary*.

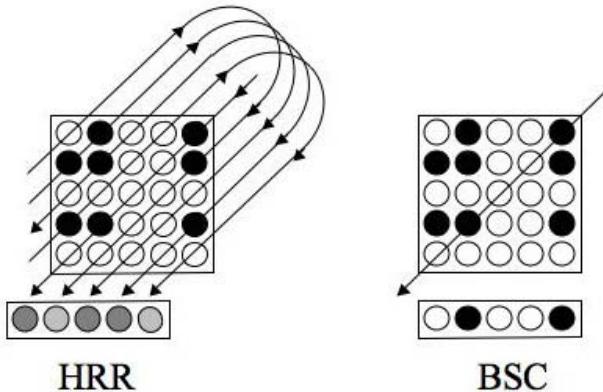


Figure 2. Methods for keeping fixed dimensionality in tensor-product representations.

2. Holographic Reduced Representations and Binary Spatter Codes

Because the dimension of the tensor product increases with each binding operation, the size of the representation grows exponentially as more recursive embedding is performed. The solution is to collapse the $N \times N$ role/filler matrix back into a length- N vector. As shown in Figure 2, there are two ways of doing this. In Binary Spatter Coding, or BSC [15], only the elements along the main diagonal are kept, and the rest are discarded. If bit vectors are used, this operation is the same as taking the exclusive or (XOR) of the two vectors. In Holographic Reduced Representations, or HRR [16], the sum of each diagonal is taken, with wraparound (circular convolution) keeping the length of all diagonals equal. Both approaches use very large ($N > 1000$ elements) vectors of random values drawn from a fixed set or interval.

Despite the size of the vectors, VSA approaches are computationally efficient, requiring no costly backpropagation or other iterative algorithm, and can be done in parallel. Even in a serial implementation, the BSC approach is $O(N)$ for a vector of length N , and the HRR approach can be implemented using the Fast Fourier Transform, which is $O(N \log N)$. The price paid is that most of the crucial operations (circular convolution, vector addition) are a form of lossy compression that introduces noise into the representations. The introduction of noise requires that the unbinding process employ a “cleanup memory” to restore the fillers to their original form. The cleanup memory can be imple-

mented using Hebbian auto-association, like a Hopfield Network [6] or Brain-State-in-a-Box model [17]. In such models the original fillers are attractor basins in the network's dynamical state space. These methods can be simulated by using a table that stores the original vectors and returns the one closest to the noisy version.

3. Applications

As a relatively new technology, VSA is just beginning to be used in the AI and cognitive science communities. Its support for compositional structure, associative memory, and efficient learning make it an appealing “raw material” for a number of applications. In this concluding section we review some of these applications, and outline possibilities for future work.

3.1. Representing Word Order in a Holographic Lexicon

Jones and Mewhort [18] report using a holographic / convolution approach similar to HRR, for incorporating both word meaning and sequence information into a model lexicon. Their holographic BEAGLE model performed better than (300-dimensional) Latent Semantic Analysis [19] on a semantic-distance test, and, unlike LSA, BEAGLE can predict results from human experiments on word priming.

3.2. Modeling Surface and Structural Properties in Analogy Processing

Experiments on children’s ability to process verbal analogies show the importance of both surface information (who or what the sentence is about) and structural information (who did what to whom) [20]. Eliasmith and Thagard [21] have successfully modeled these properties in DRAMA, an HRR-based model of analogy processing. Because HRR and other VSA approaches support the combination of surface and structural information through simple vector addition, DRAMA is able to model both components of analogy in a single representation.

3.3. Variables and Quantification

GOFAI has excelled at representing and reasoning with universally and existentially quantified variables; e.g., $\forall x \text{ Computer}(x) \rightarrow \text{HasBuggyProgram}(x) \vee \text{Broken}(x)$. It has been known for some time, however, that human performance on such reasoning tasks differs in interesting ways from simple deductive logic [22]. Recent work by Eliasmith [23] shows that HRR encodings of logical rules yield results similar to those seen in the experimental literature.

3.4. Future Work: VSA in Visual Map-Seeking Circuits

Arathorn ’s Map Seeking Circuits (MSCs) [24] are recurrent neural networks for recognizing transformed images, using localist representations. We propose treating the localist MSC as an input recognition device, with the localist output values subsequently encoded into VSA representations indicating items in the agent’s environment, and their spatial relationships to the agent. This would allow the representation and manipulation of multiple simultaneous items in the agent’s environment.

References

- [1] Newell, A., Simon, H.A.: Gps, a program that simulates human thought. *Lernende Automaten* (1961) 109–124
- [2] Minsky, M.L.: *The Society of Mind*. Simon and Schuster (1988)
- [3] Rumelhart, D., McClelland, J., eds.: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. MIT Press (1986)
- [4] Rumelhart, D., Hinton, G., Williams, R.: Learning internal representation by error propagation. In Rumelhart, D., McClelland, J., eds.: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1*. MIT Press (1986)
- [5] Kohonen, T.: *Self-Organizing Maps*. 3 edn. Springer-Verlag, Secaucus, NJ (2001)
- [6] Hopfield, J.: Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences* **79** (1982) 2554–2558
- [7] Elman, J.: Finding structure in time. *Cognitive Science* **14** (1990) 179–211
- [8] McClelland, J., Rumelhart, D., Hinton, G.: The appeal of parallel distributed processing. In Rumelhart, D., McClelland, J., eds.: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1*. MIT Press (1986)
- [9] Fodor, J., Pylyshyn, Z.: Connectionism and cognitive architecture: A critical analysis. *Cognition* **28** (1988) 3–71
- [10] Pollack, J.: Recursive distributed representations. *Artificial Intelligence* **36** (1990) 77–105
- [11] Minsky, M.L., Papert, S.A.: *Perceptrons: Expanded Edition*. MIT Press (1988)
- [12] Voegtlin, T., Dominey, P.F.: Linear recursive distributed representations. *Neural Netw.* **18**(7) (2005) 878–895
- [13] Gayler, R.: Vector symbolic architectures answer jackendoff's challenges for cognitive neuroscience. In Slezak, P., ed.: *ICCS/ASCS International Conference on Cognitive Science. CogPrints*, Sydney, Australia, University of New South Wales (2003) 133–138
- [14] Smolensky, P.: Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence* **46** (1990) 159–216
- [15] Kanerva, P.: The binary spatter code for encoding concepts at many levels. In Marinaro, M., Morasso, P., eds.: *ICANN '94: Proceedings of International Conference on Artificial Neural Networks. Volume 1*, London, Springer-Verlag (1994) 226–229
- [16] Plate, T.: Holographic reduced representations. Technical Report CRG-TR-91-1, Department of Computer Science, University of Toronto (1991)
- [17] Anderson, J., Silverstein, J., Ritz, S., Jones, R.: Distinctive feathers categorical perception and probability learning: some applications of a neural model. *Psychological Review* **84**(5) (1977) 413–451
- [18] Jones, M.N., Mewhort, D.J.K.: Representing word meaning and order information in a composite holographic lexicon. *Psychological Review* **114** (2007)
- [19] Foltz, T.L.T., Laham, D.: Introduction to latent semantic analysis. *Discourse Processes* **25** (1998) 259–284
- [20] Gentner, D., Toupin, C.: Systematicity and surface similarity in the development of analogy. *Cognitive Science* **10** (1986) 277–300
- [21] Eliasmith, C., Thagard, P.: Integrating structure and meaning: a distributed model of analogical mapping. *Cognitive Science* **25**(2) (2001) 245–286
- [22] Wason, P.: Reasoning. In Foss, B., ed.: *New Horizons in Psychology*. Penguin, Harmondsworth (1966)
- [23] Eliasmith, C.: Cognition with neurons: A large-scale, biologically realistic model of the wason task. In Bara, G., Barsalou, L., Buccirelli, M., eds.: *Proceedings of the 27 th Annual Meeting of the Cognitive Science Society*. (2005)
- [24] Arathorn, D.W.: *Map-Seeking Circuits in Visual Cognition*. Stanford University Press (2002)

Analogy as Integrating Framework for Human-Level Reasoning

Angela SCHWERING¹, Ulf KRUMNACK, Kai-Uwe KÜHNBERGER and Helmar GUST

University of Osnabrück, Germany

Abstract. Human-level reasoning is manifold and comprises a wide variety of different reasoning mechanisms. So far, artificial intelligence has focused mainly on using the classical approaches deduction, induction, and abduction to enable machines with reasoning capabilities. However, the approaches are limited and do not reflect the mental, cognitive process of human reasoning very well.

We contend that analogical reasoning is the driving force behind human thinking and therefore propose analogy as an integrating framework for the variety of human-level reasoning mechanisms.

Keywords. analogy, analogical reasoning, integrated cognitive abilities, cognitive model

1. Motivation

In most current approaches to model reasoning capabilities on machines, researches have examined certain reasoning mechanisms isolated from the overall context of human reasoning and learning. These reasoning mechanisms cover deduction [1], induction [2] and abduction [3]. Although such research-endeavors clearly are successful in various aspects and applications, they model just a fraction of the overall complexity of human-level reasoning. The problem of integrating such approaches to reach a cognitive system for human-level reasoning is non-trivial. We propose analogical reasoning as an integrating framework for various human-level reasoning mechanisms. This paper describes the role of analogies in performing different reasoning mechanisms in one integrated cognitive model and proposes a concrete approach for human-level reasoning based on Heuristic-Driven Theory Projection (HDTDP).

The remainder of the paper is structured as follows: Section 2 explains the integrating role of analogy in various examples for cognitive reasoning or learning processes. Section 3 proposes a cognitive model for human-level reasoning, which is elaborated further in section 4 by specifying an implementation based on HDTDP. We explain how knowledge is organized and represented in the knowledge base, how the reasoning unit utilizes this knowledge to perform different types of reasoning and how new knowledge is stored in the knowledge base.

¹Corresponding Author: Angela Schwering, University of Osnabrück, Albrechtstr. 28, 49076 Osnabrück, Germany; E-mail: aschweri@uos.de

2. The Role of Analogies in Human-Level Reasoning and Learning

Learning means inferring new or revising old knowledge and adapting the knowledge base. By the kind of knowledge learned, we can distinguish various cognitive tasks:

Learning new domain knowledge. Analogical comparison is a central mechanism for problem solving: new problematic situations (target domain) are compared to similar problems experienced in the past (source domain). Analogy is used to analyze the source and the target problem for common structures. The solution strategy applied in the source problem is transferred to the target domain to solve the problem in the new domain.

Creative learning of new conceptual knowledge. Analogical transfer is not limited to domain knowledge, but can also hypothesize new concepts. For instance, in physics, "heat" is energy that can be transferred from one body to another due to a difference in temperature. Heat in contrast to temperature is not observable and can be conceptualized only via analogy to a perceivable domain, e.g. water flowing between two vessels with different water levels. The height of the water is aligned to the temperature. From the observation that water keeps flowing until it has the same height in both vessels (source domain) and that temperature balances after some time (target domain), it can be inferred via analogy, that there exists an analogous "flowing thing" on the target side: the concept *heat*. Creative generation of knowledge of this type is classically modeled via abduction. Using analogy in this abductive process guides and motivates why certain things are hypothesized and others not.

Creating ad-hoc concepts. Establishing an analogy between two domains leads also to the construction of new ad-hoc concepts: e.g. in the Rutherford analogy between the solar system and the Rutherford atom, the sun and the nucleus, respectively the planet and the electron are aligned. For the purpose of the analogy they form the ad-hoc concept *central body* and *orbiting object* respectively. These ad-hoc concepts can emerge as permanent concepts after several learning cycles.

Learning general principles via abstraction. Analogy also plays an important role in learning general principles. Induction, the predominant mechanism of generalization in AI, is rather restricted and is usually considered to require a large set of samples incorporating identical patterns. Combining analogical and inductive reasoning mechanisms enables a reasoning system to generalize over analogical patterns which increases the flexibility and makes it similarly powerful as human generalization [4].

3. Cognitive Model for Human-Level Reasoning

Figure 1 illustrates a cognitive model for human-level reasoning. We now explain the role of the knowledge base, the reasoning unit and the interaction of both.

The knowledge base stores all information about the world. It represents the human memory. Domain knowledge constitutes the main part of the knowledge and contains information such as "Planets revolve around the sun." or specific rules for solving problems. As part of the knowledge base we explicitly distinguish conceptual knowledge, describing all concepts by their taxonomic relationships to each other in a hierarchy. This hierarchy contains top-level concepts as well as domain concepts.

The reasoning unit comprises four different reasoning mechanism: the classical approaches deduction, induction and abduction, but also analogical reasoning. Analogical reasoning performs as the core of this framework.

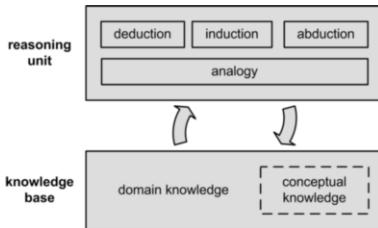


Figure 1. Cognitive model of human-level reasoning.

Deduction infers knowledge given implicitly by facts or rules: if the premises evaluate true, deduction infers that the conclusion is true as well. From a cognitive science perspective, researchers often claim that deduction plays no or a minor role in human reasoning [5,6]. Deduction is nevertheless a component of the reasoning unit, because it can support analogical reasoning in different ways: Deduction enables a machine to make implicit knowledge explicit and therefore knowledge becomes independent of the chosen representation. Moreover, the ability to establish an analogy highly depends on an adequate representation of the domains. Deduction is used to re-represent knowledge to make analogous structures in the domains obvious.

Induction creates knowledge via generalization over a set of cases which share common properties or relations and deduces some general law or principle. The "classical induction" in artificial intelligence and machine learning requires a very large set of data samples to draw sound conclusions. Humans apply inductive strategies for reasoning already with a limited number of examples (e.g. four or five cases). Moreover, they are able to compare samples which share the common pattern only at an abstract level, while "classical induction" works only when the patterns across the data samples are identical. Combining induction with analogy, this reasoning mechanism gains the flexibility it needs to reflect human generalization techniques (c.f. section 4 or [4]).

Abduction is reasoning from an effect to its cause: from a known general rule and a true conclusion it is hypothesized that the premise is true as well. Analogical reasoning can motivate abduction-like inferences [7]: Assume p and $p \rightarrow q$ are true in the source domain and therefore q can be inferred. If an analogous counterpart q' of q is known in the target domain, we can transfer via analogy the rule $p' \rightarrow q'$ and abductively conclude that p' also holds in the target domain. There could be other causes for q' , but since p is the cause in the analogous domain, it is likely that p' is also the cause in the target domain.

Analogy aims to identify structural commonalities, analogous relations or objects playing the same roles in two different domains - the source and the target domain. Analogical learning is typically applied across two different domains. The purpose of analogies is to adapt knowledge available about the source such that it can be applied to the target in a way that new analogous inferences can be drawn. This way it is possible to gain completely new hypotheses about the target domain which still have to be proven. Analogical transfer leading to inconsistencies in the target domain is considered as an indicator for a bad (wrong) analogy. Analogical reasoning can also be used to extract commonalities between source and target which are captured in an abstract generalization. However, this generalization is not assumed to be generally true (as it is the case for induction), but only for the specific source and target domain. Good analogies can be analogously applied and tested in other domains.

4. Implementation of the Cognitive Model

Figure 2 illustrates our approach to the cognitive model for human-level reasoning.

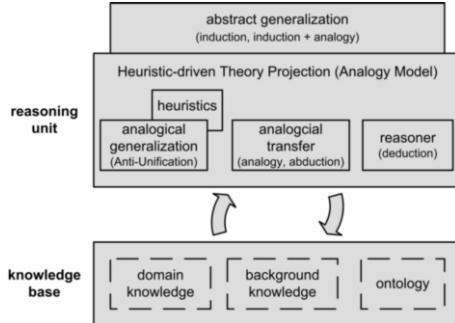


Figure 2. An HDTP-based implementation of the cognitive model for human-level reasoning.

The **knowledge base** consists of domain knowledge, background knowledge and an ontology. The domain and the background knowledge constitute all knowledge about the world and are specified by formulas represented in many-sorted first-order logic. The domain knowledge contains domain-dependent knowledge, such as knowledge about thermodynamics and fluid dynamics (e.g. water and its flowing characteristics). The background knowledge contains domain-independent knowledge such as the formula $distance_{eucl}(X, Y) = distance_{eucl}(Y, X)$ stating that the Euclidian distance is symmetric. The ontology contains conceptual knowledge specified via taxonomic relations to other concepts [8]. It describes the relation of top-level concepts like *massterm*, *time* or *object* to (usually domain-specific) concepts such as *vessel* or *water*.

The heart of the **reasoning unit** is the HDTP [9], a symbolic approach developed at the University of Osnabrück to model analogies. Since HDTP is based on first-order logic, it can naturally interact and easily integrate the formal reasoning mechanisms deduction, induction and abduction. The reasoning unit contains different modules. The interaction of these modules enables the modeling of human reasoning strategies. We explain the reasoning unit by describing the different steps of the reasoning process. Of course, the process differs depending on what kind of knowledge is inferred or learned.

All reasoning processes start with some knowledge as input - in analogical reasoning two domain theories Th_S and Th_T modeling source and target. The theory of anti-unification [10] is used to generalize the structural commonalities between two formal theories. The module *analogical generalization* computes an analogy by successively selecting a suitable formula from Th_T and Th_S and constructing a generalization (Th_G) together with the corresponding substitutions. Heuristics to minimize the complexity of substitutions [9] have been developed to guide this anti-unification process. The output is the generalized theory Th_G , which covers explicitly the commonalities and establishes an analogous relation between source and target. Usually, the source domain is well known and modeled richer than the target domain (about which you want to learn something). Based on the analogical relation between source and target, the module *analogical transfer* identifies knowledge in the source which could be transferred to the target domain. The knowledge gained by the analogical transfer can be facts or rules, but also conceptual knowledge such as new concepts or a revised concept hierarchy. The

reasoner module must check the hypotheses suggested by the analogy for consistency, i. e. it must prove whether the new facts or rules hold or conflict in the target domain. The reasoner module can also be used during the analogical generalization for restructuring domain knowledge, e.g. restructure the target domain by inferring implicit formulas from the given facts and rules such that the resulting representation is suitable for a structural comparison and can be anti-unified (e.g. [11]). On top of HDTP, the module *abstract generalization* applies inductive reasoning mechanisms based on the analogical comparison to enable inferring general principles [4].

All inferred knowledge is stored in the knowledge base and leads either to extended or revised theories (in case new facts or laws are learned for the domain or background knowledge or new general principles are hypothesized via the abstract generalization module) or an extended or revised ontology (if concept knowledge is learned).

5. Summary

This paper proposes to model human-level reasoning based on analogy. Combining analogy with classical reasoning mechanisms leads to a cognitive model in which various human reasoning and learning strategies can be explained. We outline our approach using HDTP as integrating framework and describe the role of analogy in various inferences.

Acknowledgements

The work was supported by the German Research Foundation (DFG), grant KU1949/2-1.

References

- [1] Bibel, W.: *Deduction: Automated Logic*. Academic Press London (1993)
- [2] Muggleton, S.; De Raedt, L.: Inductive logic programming: Theory and methods. In: *Journal of Logic Programming*, 19,20. (1994) 629–679
- [3] Magnani, L.: *Abduction, Reason, and Science. Processes of Discovery and Explanation*. Kluwer Academic/Plenum Publishers, New York (2001)
- [4] Gust, H., Krumnack, U., Kühnberger, K.U., Schwering, A.: Integrating analogical and inductive learning at different levels of generalization. In: *Workshop on Learning from Non-Vectorial Data (LNVD2007)*, Osnabrück (2007)
- [5] Gigerenzer, G.: Bounded and rational. In Stainton, R., ed.: *Contemporary Debates in Cognitive Science*. Blackwell Publishing (2006) 115–133
- [6] Laird, J.: *Mental Models: Towards a Cognitive Science of Language, Inference and Consciousness*. Cambridge University Press (1983)
- [7] Schwering, A., Krumnack, U., Kühnberger, K.U., Gust, H.: Analogical reasoning with SMT and HDTP. In: *2nd European Cognitive Science Conference (EuroCogSci07)*, Delphi, Greece (2007) 652–657
- [8] Gust, H., Kühnberger, K.U., Schmid, U.: Ontologies as a cue for the metaphorical meaning of technical concepts. In Schalley, A., Khentzos, D., eds.: *Mental States: Evolution, Function, Nature*. John Benjamins Publishing Company (2007) 191–212
- [9] Gust, H., Kühnberger, K.U., Schmid, U.: Metaphors and heuristic-driven theory projection (HDTP). Volume 354. (2006) 98–117
- [10] Krumnack, U., Schwering, A., Gust, H., Kühnberger, K.U.: Restricted higher-order anti-unification for analogy making. In: *20th Australian Joint Conference on Artificial Intelligence (AI07)*. Lecture Notes of Artificial Intelligence, Gold Coast, Australia, Springer (2007)
- [11] Schwering, A., Krumnack, U., Kühnberger, K.U., Gust, H.: Using gestalt principles to compute analogies of geometric figures. In: *29th Meeting of the Cognitive Science Society (CogSci07)*, Nashville, Tennessee (2007) 1485–1490

Designing Knowledge Based Systems as Complex Adaptive Systems

Karan Sharma (karan@uga.edu)

Artificial Intelligence Center, University of Georgia

Abstract. The paper proposes that knowledge based systems must be designed as complex adaptive systems and any other approach is not fundamental, even if sometimes it yields good results. Complex systems are characterized as having global behavior not always explainable from local behavior. Here we propose that the way we perceive knowledge in AI needs to change to Complex Adaptive, hence the need for a paradigm shift is stressed. Almost all historical KBS were not complex systems in an authentic sense. But it is not a good idea to criticize them because with available resources and theories, they did their best. Sooner or later, we will have to design our KBS as complex adaptive systems, so why not sooner. There are three mechanisms that must be part of any knowledge based system, viz., Interdependency and fluidity, mechanisms for attribution of emergent properties and self-organization.

Keywords. Knowledge Based Systems, Complex Adaptive Systems, Cognitive Fluidity, Self-organization.

1. Introduction

Although the time has come that our perception of the whole field of AI should be of complex systems and more rule-based, well-defined and logic style approaches need rethinking, we confine ourselves to the discussion of KBS in this paper. Richard Loosemore [1] argues

“One the most basic assumptions made by Artificial Intelligence researchers is that the overall behavior of an AI system is related in a lawful, comprehensible way to the low level mechanisms that drive the system.....this apparently innocent assumption is broken, because all intelligent systems, regardless of how they are designed, must be complex systems”[1]

Evidence in anthropology and psychology suggests that mental fluidity is central to human cognition. It is this fluidity that lends us our higher level abilities that only our species Homo sapiens sapiens possess. If our goal is to build human-like intelligence, the only valid path is complex adaptive systems – since by their very nature these systems can efficiently produce an analog of mental fluidity. Human cognition is a complex adaptive system [2], hence KBS have to be complex adaptive, it cannot be otherwise. For the purpose of this paper when we talk of KBS, it encompasses all areas of Artificial Intelligence that handle knowledge including case-based reasoning, expert systems, etc.

Complex systems are the systems that involve interactions among various components and as a result of these interactions global emergent behavior emerges in a system. [3] suggests "interdependence" among various components is a more generic term since it is the interdependent parts that create emergent properties rather than interconnected parts. Complex systems are known for various characteristics; among them widely discussed are emergence, self-organization and non-linearity. Here it is important to mention that KBS must be complex adaptive rather than just complex because it must be sensitive to any new knowledge that enters the system. For the detailed study of complex systems reader is referred to [3, 4] and for complex adaptive systems to [5].

There are three mechanisms that must be part of any knowledge based system, viz., Interdependency and fluidity, a mechanism for attribution of emergent properties and self-organization. These mechanisms will not only make knowledge more complex, but also make it adaptive to new information that comes in.

2. Interdependency in Knowledge Bases

Referring to the brains of early humans, Mithen [6] asserts:-

"we can safely state that in spite of linguistic differences, all Early Humans shared the same basic type of mind:a swiss-army-knife mentality. They had multiple intelligences, each dedicated to a specific domain of behavior, with very little interaction between them.....Early Humans seem to have been so much like us in some respects, because they had these specialized cognitive domains; but they seem so different because they lacked a vital ingredient of the modern mind: cognitive fluidity" [6]

Grounding his arguments with evidence from archaeology and anthropology he further suggests that "cognitive fluidity" is responsible for our intelligence and the rise of human civilization (including art, science and religion). Even our closest cousins Neanderthals did not have this ability. But it is very unfortunate that AI researchers and builders of the KBS have totally disregarded this ability and created systems that stored information like Early Humans. Cyc [7] did better by at least creating Neanderthal-like common sense, that is, there are not very flexible across-the-domain connections. The time has come that our perspective changes and we start treating knowledge as complex for AI purposes. Unfortunately what should have been done first will be done last. All the KBS in the past, without exception, were fundamentally flawed because they represented knowledge "as is" without giving any consideration to the fact how a thing can be represented in terms of other entities.

Besides anthropology, there are theories by psychologists and computer scientists to account for our mental fluidity. Arthur Koestler [8] proposed the Bisociation, the mechanism in human mind, through which different planes of thought come together to produce a novel thought. Karmiloff-smith [9] came up with a Representational Redescription model explaining how children's representations become more flexible with age. Fauconnier and Turner [10] proposed theory of conceptual blending, in which multiple concepts can blend to form a new concept with emergent properties. Boden [11] also suggested that transformation of conceptual spaces is central to our thought processes. [12, 13] came up with a program Copycat to model analogical thought but

they asserted it can be extended to all fluid mental concepts. The major achievement [13, 14] of the Copycat program was to show that human cognition is a "complex adaptive system." The above works unanimously offer a unique insight into the workings of the human mind - In human mind, elements from different areas are interdependent and often come together to form a coherent whole whose properties may be emergent.

Creating highly fluid systems that are complex and adaptive is what is required. KBS should be capable of "Seeing one thing in the frame of other" and free merger of concepts should be the rule. High-Interdependency among representations must be the rule in any Knowledge Base. Each concept must be represented from the perspective of other concepts in the knowledge base. And a concept should have representation from the perspective of multiple other concepts. This is done to ensure high interconnectivity, which obviously will not only make KBS highly fluid but also adaptive, just like human cognition. R.M. French [15, 16] has argued that representation for any AI system, if it is not malleable, then it is necessarily flawed. According to him, the representations that are fixed can never produce an analog of human cognition and fluidity. Each representation should not be "as is", that is, there must not be fixed representation of any concept.

3. Emergence

The reason emergence becomes a central issue in the design of knowledge bases is that there are global properties possessed by entities which are not reducible to the components and subparts of the entities. For example – Psychological characteristics of the human brain cannot be explained in terms of just one neuron or many neurons taken independently. Rather interconnections and interdependence of the neurons display emergent behavior. Emergence is discussed in detail in [3, 17].

Let us consider a hypothetical knowledge base. Assume the representation to be either predicate calculus, logic, semantic network, frame based or any symbol manipulation representation. Also, we assume our knowledge base to be all-inclusive, that is, containing all the information in the world. Suppose we examine the concept of "Automobile" in our knowledge base. If we start with any possible initial state of the concept "Automobile" in any representation, and from there try to derive this property "Means of transportation", we see that no matter what change we make to that state it is simply impossible to derive the property "Means of transportation" bottom-up since derivation is the process of sequential state transition in which a part or subpart of the information is changed to reach an end state. However, the emergent property can never be an end state because

1.) The concept "Automobile" is represented in the form of its parts (engine, tires, windows, mirror, doors and their relationships). Information "Means of transportation" is not contained in any one part, or combination of few parts, rather it is the interactions of all the parts that emerge this property, hence, trying to extract information "Means of transportation" from the parts is impossible. This property "Means of transportation" is the property of specific configuration created by all of the components and their interactions making up the concept "Automobile". Derivations, as in any representation lead to an end state in a deterministic fashion, however starting from any initial state in this problem we can never be sure that the next state will lead us to correct end state (our

property). Hence the property is not derivable from the inner components and is globally emergent.

2.) To derive the property from all the other information in the knowledge base besides the "Automobile" concept, it is that the property if it is the property of another entity in the knowledge base, then it is going to be an emergent property of that entity, that is, not derivable from its own components. For example - the property "Means of transportation" can be attributed to the concept of "Horse" and it is non-derivable for this concept. Since we have already seen that the property cannot be obtained from its own components, the only way is for the property to transfer from one point in the knowledge base to another point by transferring from one whole to another whole. The property of the concept "Horse" can be transferred to the concept "Automobile" in whole.

We need to have such mechanisms in our systems that help us in attribution of emergent properties because trying to derive any higher order property from lower level components is almost impossible. Analogical reasoning is one such mechanism but there are many more similar strategies that must be incorporated in the systems. The idea is to transfer stuff from one whole to another based on some criteria, and not try to derive things bottom up.

4. Self-Organization

Karmiloff-smith [9] while proposing his RR model wrote

"My claim is that a specifically human way to gain knowledge is for the mind to exploit internally the information that it has already stored.....by redescribing its representations or, more precisely, by iteratively re-representing in different representational formats what its internal representations represent." [9]

The above lines clearly hint to the process of self-organization in the human mind. A process analogous to re-representation mechanism is a necessity in a complex adaptive KBS. [18] suggested self-organization is a requirement for any system if it is to be creative. The idea of incorporating self-organization is simple – any new information is connected to lots of other information and this interdependency can lead to potential effects on all the interdependent information. The goal of self-organization is to adapt the whole KBS to any incoming information so that all the information is represented in the most accurate state in the KBS. Accurate state in the KBS is variable since induction of any new information can change the accuracy. For example – in the nineteenth century, the most accurate explanation for universe was in terms of Newtonian physics, however, in the twentieth century after new information came in (Theory of relativity), the most accurate explanation for universe was in terms of relativity.

The more inter-dependent various fragments of knowledge inside the knowledge base are, the more they are prone to the effects of the induction of new knowledge. This new knowledge, if it modifies any knowledge in the KBS, can subsequently lead to chain of modifications because this modified knowledge is highly interdependent with various other fragments of knowledge. The system must be capable of self-organizing at this stage, that is, just by using all the information that is internal to the KBS, the system should be able to reach most accurate representation or perception or frame for each fragment of knowledge that is effected by newly induced knowledge in some way.

5. Conclusion

No doubt, the road to making knowledge complex and adaptive for AI is filled with some serious bottlenecks, nevertheless, it is reachable. Designing complex adaptive KBS is the most ideal approach that will make them adaptive by incurring several advantages over conventional systems. The most significant advantage being increase in information resulting from high interdependency in the knowledge base. Since each entity or situation can be perceived in multiple frames, our systems will have an option to choose best frame that is most accurate representation relative to other representations, thus increasing system's information, reliability and adaptiveness.

Acknowledgement

I thank Dr. Walter D. Potter, Director, Artificial Intelligence Center, University of Georgia for offering useful comments towards the completion of this paper.

References

- [1] R.P.W. Loosemore. Complex systems, artificial intelligence and theoretical psychology. In B. Goertzel and P. Wang, editors, *Proceedings of the 2006 AGI Workshop*. IOS Press, Amsterdam, pages 159-173, 2007.
- [2] H. Morowitz and J. Singer, editors. *The Mind, the Brain, and Complex Adaptive Systems*. Addison-Wesley, Reading, Massachusetts, 1995.
- [3] Y. Bar-Yam. *Dynamics of Complex Systems*. Perseus Books, Cambridge, Massachusetts, 1997.
- [4] T.R.J. Bossomaier and D.G. Green, editors. *Complex Systems*. Cambridge University Press, 2000.
- [5] J. H. Holland. *Hidden Order*. Addison Wesley Reading, Massachusetts, 1995.
- [6] S. Mithen. *The Prehistory of the Mind: A Search for the Origins of Art, Religion and Science*. Thames and Hudson, London, 1996.
- [7] D. B. Lenat and R. V. Guha. *Building Large Knowledge-based Systems: Representation and Inference in the Cyc Project*. Addison Wesley, 1990.
- [8] A. Koestler. *The Act of Creation*. Macmillan, New York, 1964.
- [9] A. Karmiloff-Smith. *Beyond Modularity: A Developmental Perspective on Cognitive Science*. MIT Press, 1992.
- [10] G. Fauconnier and M. Turner. *The Way We Think: Conceptual blending and the Mind's Hidden Complexities*. Basic Books, 2002.
- [11] M.A. Boden. *The Creative Mind: Myths and Mechanisms*. Weidenfeld and Nicolson, London, 1991.
- [12] D.J. Chalmers, R.M. French and D.R. Hofstadter. High-level perception, representation, and analogy: a critique of artificial intelligence methodology. *Journal of Experimental and Theoretical Artificial Intelligence* 4:185–211, 1992.
- [13] D. R. Hofstadter and M. Mitchell. The Copycat project: A model of mental fluidity and analogy-making. In K. J. Holyoak and J. A. Barnden, editors, *Advances in Connectionist and Neural Computation Theory: Analogical Connections* 2:31–112, Ablex, Norwood, New Jersey, 1994.
- [14] M. Mitchell. Analogy-making as a complex adaptive system. In L. A. Segel and I. R. Cohen, editors, *Design Principles for the Immune System and Other Distributed Autonomous Systems*. Oxford University Press, New York, pages 335–359, 2001.
- [15] R. M. French. *The Subtlety of Sameness*. MIT Press, 1995.
- [16] R. M. French. When coffee cups are like old elephants or Why representation modules don't make sense. In *Proceedings of the International Conference New Trends in Cognitive Science*, A. Riegler and M. Peschl, editors, Austrian Society for Cognitive Science, pages 158–163, 1997.
- [17] J. H. Holland. *Emergence: From Chaos to Order*. Addison Wesley, 1998.
- [18] P.H.G. van Langen, N.J.E. Wijngaards, and F.M.T. Brazier. Designing creative artificial systems. In A.H.B. Duffy and F.M.T. Brazier, editors, *AIEDAM, Special Issue on Learning and Creativity in Design*, 18(4):217–225, 2004.

Artificial general intelligence: an organism and level based position statement

Leslie S. SMITH¹

Department of Computing Science and Mathematics, University of Stirling, UK

Abstract. Do simple animals, even single-celled animals, display general intelligence? If we take this to mean that they can cope with their environment, even when it is dynamic, the answer would have to be yes. What is it that gives them this easy capability, yet makes it hard for us to design systems which have it? We suggest that it is from the non-determinism which arises from the lowest levels of their operation, and from the way in which the different levels of their operation interact. We propose techniques whereby these might be incorporated into artificial systems.

Keywords. artificial general intelligence, brain model, paramecium, level interaction

Introduction

There are many views of what should be described as artificial general intelligence. General intelligence is not a well-defined concept: but then, neither is intelligence, nor are many other general concepts, such as life. But this lack of definition does not mean that they have no meaning: rather that they occupy a diffuse area, rather than a single point: they are cluster concepts. Better then, to ask what some of the characteristics of these concepts are, and to attempt to approach them, rather than trying to define them into a corner then reconstruct them. In what follows, I have chosen two particular aspects of organisms that display what I consider general intelligence, and I have attempted to delineate a possible program of research based on these.

1. The low level organism view

Much of what has been researched in artificial intelligence comes from a view that humans have intelligence, and other animals do not. I suggest that this humano-centric view of the world has as much place in general intelligence research as an Earth-centred view of the Universe has in astronomy. General intelligence is about keeping an organism alive and thriving in a complex and ever-changing environment. It differs from purely reac-

¹Corresponding Author: Leslie S. Smith. Department of Computing Science and Mathematics, University of Stirling, Stirling FK9 4LA, Scotland, UK; E-mail: l.s.smith@cs.stir.ac.uk

tive behavior in that behaviors are not necessarily entirely contingent on the organisms senses. Certainly, there are aspects of intelligence found in humans that are not found in other animals, most notably complex language: however, there are certainly aspects of general intelligence visible in the behavior of animals.

One is then led to ask what the simplest animal to display any form of general intelligence is, and whether we can learn from it, and build something based on it?

Consider the behaviour of a very simple single celled organism: a paramecium. This animal has no nervous system as such, yet has successfully survived in freshwater environments for many millions of years. The animal's single cell has a membrane surrounding it, and this is covered in small hair-like appendages (cilia). These allow it to move. It eats by endocytosis (incorporating food particles into its cytoplasm), and excretes by exocytosis. Endocytosis is mediated through protein sensors on the membrane surface. The membrane is not uniform: there is a gullet-like area for eating, and an anus-like area for excretion. The animal can move towards food. Reproduction can be either sexual or asexual. Does it display general intelligence? Or is it purely reactive? What aspects of its behavior display intelligence?

Consider a slightly more complex organism, one with a rudimentary nervous system. Actually, its visible behavior is likely to be very similar to that of the paramecium: the primary difference is that it has specialized cells for different processes, such as movement, digestion, and control. Yet in the end, eating and excretion are essentially orchestrated endo- and exo- cytosis. Control of the animal is achieved also through the action of the nerve cells on these other cells, and this is achieved through neurotransmitters sent through exocytosis, and received through receptors and ion channels, which then alter cell behaviour. Does it display intelligence or is it purely reactive? Or perhaps the animal displays general intelligence, but each cell is purely reactive?

Artificial intelligence designers often consider complexity to be part of the nature of intelligence. They build their systems out of logical entities (such as gates, or threshold logic elements), using some mixture of hardware and software to develop complex logical systems. Given appropriate input, and appropriate interpretation of their output, they can perform sophisticated tasks. Each element is utterly deterministic, and could be considered reactive. But does the overall system display general intelligence?

The examples above, two living and one not, may or may not be considered to display general intelligence depending on the standpoint of the onlooker. (I suggest that there is a range of "intelligences" between purely reactive systems and truly intelligent ones, but that attempting to delineate exactly where each example lies on a line connecting them is not a useful exercise.) However one thing that is clear is that the living entities are very different from the non-living one. Is this difference one purely of implementation (living cells as against logic gates), or are there deeper differences?

It has been suggested that life is ultimately based on logic (by which is usually meant that living entities are based on the same logical (and physical) rules that pervade the rest of the universe, rather than there being 'special sciences' that govern living entities). Yet this does not imply that life is necessarily based on the same two-valued logic that pervades digital computing. One of the great strengths of two-valued logic is also one of its problems: it is utterly deterministic: repeating the same two-valued logical steps on the same data will always give the same result. This goes against what we see in living organisms. They always show wide diversity of behaviour (and, indeed, this diversity

seems to exist almost no matter what level one examines the systems at). This leads to two different questions:

1. Where does this diversity originate from, and
2. Can we implement this diversity using the two-valued logic systems?

Even the simplest living organisms do not display two-valued logical behaviour. Non-determinism is built in from the very bottom level. Consider a piece of the cell membrane of a paramecium in water, which has touched a possible food particle. In order for the food particle to be ingested, a reaction needs to take place between the food particle and some sort of receptor on the surface of the cell. The molecules of both the food particle and the receptor are large (probably proteins, or fatty acids). The reaction occurs when the reactive parts of the molecules of the reactants become close enough to each other [1]. Both reactants are in motion because of their temperature, and because of Brownian motion induced by the water molecules. The likelihood of a reaction occurring is directly related to the likelihood of the reactive surfaces meeting each other before the same random movement causes the food particle to move away from the cell. This whole process is essentially nondeterministic.

Virtually all the processes that occur at the cell membrane are of a related form: for example, ion channels which alter shape due to variations in the charge patterns surrounding them are actually continuously in motion, and their shape is essentially best described as a stochastic system [2]. Thinking of ion channel behaviour as being like transistor behaviour (as Mead suggests in his 1989 book [3]) can therefore be slightly misleading. We conclude that there is diversity in response in biological systems even at the lowest level.

We suggest that the diversity seen at higher levels in biological systems arises from this lower level diversity, in the same way as the lack of diversity in digital electronic system behaviour has its roots in the absolute determinism of the lowest level operation of these electronic systems.

But how important is this diversity? We suggest that this diversity in response lies at the root of successful adaptivity: by responding differently to essentially identical environmental events, and then ‘rewarding’ responses which have the best outcomes by making them more likely to occur in future, an organism can adapt to its environment rapidly and effectively. This is not a new idea: Campbell [4] places blind adaptivity at the root of all learning, from this lowest level all the way up to scientific knowledge, and coined the term *evolutionary epistemology* to describe it. The adaptivity need not be entirely blind: more recent learning techniques such as reinforcement learning and TD learning [5] suggest that previous experience can be used to guide learning and hence adaptive evolution of behavior. We consider this evolutionary capability to be at the root of general intelligence, both in simple and higher organisms.

2. Interlinking levels

Brains operate at many different physiological levels simultaneously. There are numerous different species of ion channels, some voltage sensitive, some not, all opening and closing simultaneously; there are billions of synapses some excitatory, some inhibitory, some shunting, some transmitting, some not; there are billions of neurons, some spiking

some not; there are cortical columns processing information, and communicating with other columns; there are brain areas each with their (partial) specialisation: the whole lot comes together to produce the behaviour of that brain. All of these levels of viewing what is occurring in an animal brain are simultaneously valid. Clearly, they are not independent of each other. One might attempt a similar view of a modern computer system; there are billions of transistors, some on, and some off, billions of gates, some with logic 1 and some with logic 0 outputs; there are objects and functions all running (quasi-) simultaneously, all coming together to produce the behaviour of that computer system.

There is, however, a major difference between the two. Digital computer systems are designed and built hierarchically, whereas biological brains are not. This difference shows itself in the ways in which these different levels of operation interact. In a computer, the gates are built out of transistors, and, although the output of a gate is the output of a transistor (and the input to a gate is the input to a transistor) it is safe to ignore transistor operation entirely, as everything about the gate operation can be abstracted away from this lower level. Similarly, the entire operation of all the gates etc. in the hardware can be abstracted into the instruction set (and interrupt structure etc.), so that the software can be considered entirely independently of the hardware. The same is true for layers of software: each interface between levels allows the lower levels to be entirely abstracted away.

Often neural systems have been viewed in this way as well. Ionic channels and neuromodulators subserve entities like synapses, and synapses and the collection of charge on dendrites subserve the generation of spikes. Neurons in a cortical column are there in order to generate the behaviour of the column. In other words, the brain is often viewed as if it has been designed from an engineering perspective. But nature and evolution are not bound by engineering design laws! In fact there is considerable interaction between levels. For example, neurotransmitter released from one synapse will leak to another one. Further, small molecules which act as neuromodulators (such as Nitrous Oxide, NO) can diffuse over relatively wide areas, causing non-localised interactions which will cross levels. Research suggests that these non-local links can rapidly reconfigure neural systems [6].

Even with engineered systems, using genetic algorithms to optimize performance can result in level crossing being utilised in a problem solution. A good example of this is in Adrian Thompson's work [7] in which an FPGA was optimised to solve a problem, using a fitness function calculated directly from the operation of the chip. The result was a solution which used all sorts of cross-talk between entities which are supposed to be entirely independent directly in the system operation. Another example may be found in the work off the robotocist Mark Tilden, who utilises the mechanical properties of the materials of his robots in the robot controller. Clearly, biological systems are quite free to develop such solutions to problems (and don't have to worry about making their systems comprehensible to engineers!).

How does this interlinking of levels contribute to general intelligence? It can provide direct (and therefore fast) interaction between different parts of the brain, permitting rapid reconfiguration. This could permit selection between different mechanisms that might otherwise be viewed as purely reactive. On the other hand it might also cause inappropriate crosstalk (which is why it is normally intentionally omitted from engineered systems).

3. Synthesizing general intelligence based on simple animals and linking levels: towards a program

We have stressed the importance of non-determinism at the lowest levels making itself visible at higher levels (up to eventual behavior). Could this non-determinism be replicated by noise in an otherwise deterministic system? And if so, would the form of the noise matter? It appears that the critical issue is the capability for selecting different behaviors, and rewarding those that are most appropriate. We have also stressed the way in which the different levels of operation are interlinked in biological systems. However, it is less clear how this contributes towards general intelligence. One possible answer is that this results in another form of variation in behavior because of the way in cross-level linkages alter behavior in (relatively) unexpected ways. It is not simply noise, because the effects will not be random, but (at least to some extent) repeatable. We note that it can also allow rapid changes to be made to the nature of processing.

Replicating this type of general intelligence is not straightforward. If we wish to take an Artificial Life approach, we clearly need to have a rich environment, as well as critters exhibiting multiple levels inhabiting this environment. The levels should not be precisely hierarchically arranged, but able to affect each other. These critters need to be non-deterministic, preferably with this non-determinism being implemented right at the lowest of levels. This should enable different behaviors in response to similar events, and adaptation towards the most appropriate behavior whether blindly or using (e.g.) reinforcement learning. An additional key may be to add a genetic algorithm system on top of this, enabling modification (and possibly optimization) of the internal structure, including the ways in which the levels are able to influence each other. This could permit fast reconfiguration of the system in the light of environmental cues.

Acknowledgements

Thanks to Catherine Breslin for suggesting the Paramecium as a very simple animal with interesting behavior, to Karla Parussel for interesting discussions, to the UK EP-SRC Grand Challenges in Microelectronic Design Network for inspiration, and to the anonymous referees for historical context.

References

- [1] F.T. Hong, A multi-disciplinary survey of biocomputing: Part 1: molecular and cellular aspects, in *Information Processing and Living Systems* V. B. Bajić and T. W. Tan, Eds., Imperial College Press, London, 2005, 1–139.
- [2] T.F. Weiss, *Cellular Biophysics*, MIT Press, 1996.
- [3] C. Mead, *Analog VLSI and Neural Systems*, Addison Wesley, 1989.
- [4] D.T. Campbell, Evolutionary Epistemology, in *The Philosophy of Karl Popper*, P.A. Schlippe, Ed, Open Court, Illinois, 1974.
- [5] R.S. Sutton and A.G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, 1998.
- [6] K. Parussel, A bottom-up approach to emulating emotions using neuromodulation in agents, University of Stirling Ph.D. thesis, <http://hdl.handle.net/1893/113>, November 2006
- [7] A. Thompson, I. Harvey and P. Husbands, Unconstrained evolution and hard consequences, in *Towards Evolvable Hardware: The evolutionary engineering approach*, E. Sanchez and M Tomassini, Eds, Springer Verlag, LNCS 1062, 1996, 136–165

This page intentionally left blank

Workshop Papers

This page intentionally left blank

THE ARTILECT WAR

Cosmists vs. Terrans

A Bitter Controversy Concerning Whether Humanity Should Build Godlike Massively Intelligent Machines

Prof. Dr. Hugo de GARIS
Director of the “China-Brain Project”
Institute of Artificial Intelligence,
Department of Computer Science,
School of Information Science & Technology,
Xiamen University, Xiamen,
Fujian Province, China.
profhugodegaris@yahoo.com

Abstract. This paper claims that the “species dominance” issue will dominate our global politics later this century. Humanity will be bitterly divided over the question whether to build godlike, massively intelligent machines, called “artilects” (artificial intellects) which with 21st century technologies will have mental capacities trillions of trillions of times above the human level. Humanity will split into 3 major camps, the “Cosmists” (in favor of building artilects), the “Terrans” (opposed to building artilects), and the “Cyborgs” (who want to become artilects themselves by adding components to their own human brains). A major “artilect war” between the Cosmists and the Terrans, late in the 21st century will kill not millions but billions of people.

1. Introduction

This paper claims that the “species dominance” issue will dominate our global politics this century, resulting in a major war that will kill billions of people. The issue is whether humanity should build godlike, massively intelligent machines called “artilects” (artificial intellects), which 21st century technologies will make possible, that will have mental capacities trillions of trillions of times above the human level. Society will split into two (arguably three) major philosophical groups, murderously opposed to each other. The first group is the “Cosmists” (based on the word Cosmos) who are in favor of building artilects. The second group is the “Terrans” (based on the word Terra,

the earth) who are opposed to building artilects, and the third group is the “Cyborgs”, who want to become artilects themselves by adding artilectal components to their own human brains.

This paper is structured as follows. Section 2 introduces the 21st century technologies that will make artilect building possible, and thus force humanity to choose whether to build artilects this century or not. Section 3 proves that artilects will have mental capacities trillions of trillions of times above the human level, i.e. they will have godlike, massively intelligent abilities. Section 4 describes how the species dominance debate will start. Section 5 introduces who the major players will be in this debate. Section 6 presents the arguments of the Cosmists in favor of building artilects. Section 7 presents the arguments of the Terrans opposed to the building of artilects. Section 8 presents the Cyborg arguments in favor of converting humans into artilects. Section 9 describes how the artilect war will probably heat up. Section 10 shows that a major late 21st century war will kill billions rather than millions of people. Section 11 describes how inviting my audiences to vote on the Cosmist/Terran question splits them 50/50, which will only make the controversy all the more bitter. Section 12 makes an appeal to philosophers to reflect on the enormous issue of species dominance. Section 13 offers a quote and some publicity. Finally, there is only one reference, which is the author’s book on the same theme as this paper, but which treats the topic in far greater detail.

2. 21st Century Artilect Enabling Technologies

2.1. Moore’s Law

Gordon Moore, cofounder of the microprocessor company Intel, noticed in 1965 that the number of transistors on an integrated circuit (chip) was doubling every year or two. This trend has remained valid for over 40 years, and it is thought that it will remain valid for another 15 years or so, until transistors reach atomic size.

2.2. 1 bit/atom by 2020

Extrapolating Moore’s Law down to storing one bit of information on a single atom by about 2020, means that a handheld object will be able to store a trillion trillion bits of information. Such a device is called an “Avogadro Machine (AM)”.

2.3. Femto-Second Switching

An Avogadro Machine can switch the state of a single atom ($0 \leftrightarrow 1$) in a femtosecond, i.e. a quadrillionth of a second (10^{-15} sec.), so that the total processing speed of an AM is roughly 10^{40} bits per second.

2.4. Reversible Computing

If computing technology continues to use its traditional irreversible computational style, the heat generated in atomic scale circuits will be so great, they will explode, so a reversible, information preserving, computing style will be needed, usually called

“reversible computing”, that does not generate heat, hence will allow 3D computing, and no limit to size. Artilects can become the size of asteroids, kilometers across, with vast computing capacities.

2.5. Nanotech(nology)

Nanotech (i.e. molecular scale engineering) will allow AMs to be built. Nanotech will thus allow artilects to be built, once we know how to build brain like circuits. Nanotech is the “enabling technology” for artilect building.

2.6. Artificial Embryology

One of the greatest challenges of 21st century biology is to understand “development”, i.e. the embryogenic process, i.e. how a fertilized single cell grows into a 100 trillion cell animal such as ourselves. Once this process is well understood, technology will be able to create an artificial embryology, to manufacture products, hence “embryofacture” (embryological manufacture). Embryofacture will be used to build 3D complex artilects.

2.7. Evolutionary Engineering

The complexities of artilect building will be so great (e.g. the human brain has a quadrillion (10^{15}) synapses (connections between neurons in the brain)), that an evolutionary engineering approach will be needed, which applies a “Genetic Algorithm” approach to engineering products. Artilects will be built using this technique.

2.8. (Topological) Quantum Computing

Quantum computing is potentially exponentially more powerful than classical computing. It can compute 2^N things at a time, compared to classical computing’s 1 thing at a time, where N is the number of (qu)bits in the register of the quantum computer. Topological quantum computers (TQCs) store and manipulate the qubits in topological quantum fields, and are thus robust against noise. TQC will soon make quantum computers practical. Artilects will be TQC devices.

2.9. Nanotech Impact on Brain Science

Today’s top supercomputers are close to reaching the estimated bit processing rate of the human brain, (i.e. about 10^{16} bits per second), but they are far from being intelligent by human standards. What is needed to make them humanly intelligent is knowledge from the neurosciences on how the human brain uses its brain circuits to perform intelligent tasks. Nanotech will furnish neuroscience with powerful new tools to discover how the brain works. This knowledge will be quickly incorporated into the building of artilects.

2.10. Artificial Brains

The above technologies will result in the creation of an artificial brain industry and the creation of rival national brain building institutions and projects equivalent to NASA and ESA for space travel. In time, the brain building industry will become the world's largest.

3. The Artilect : Capacities 10^{24} Times Above Human Levels

As stated in the above section, the estimated bit processing rate of the human brain is approximately 10^{16} bit flips per second. This figure is derived from the fact that the human brain has about 100 billion neurons (10^{11}), with each neuron synapsing (connecting) with roughly ten thousand other neurons (10^4), hence there are a quadrillion synapses, each signaling at a maximum rate of about 10 bits per second. Thus the human bit processing rate is $10^{11+4+1} = 10^{16}$ bits per second. As mentioned in the previous section, a hand held artilect could flip at 10^{40} bits per second. An asteroid sized artilect could flip at 10^{52} bits a second. Thus the raw bit processing rate of the artilect could be a trillion trillion trillion (10^{36}) times greater than the human brain. If the artilect can be made intelligent, using neuroscience principles, it could be made to be truly godlike, massively intelligent and immortal.

4. The Species Dominance Debate Starts

The “species dominance” debate has already started, at least in the English speaking countries and China. The fundamental question is whether humanity should build artilects or not. This issue will dominate our global politics this century, and may lead to a major war killing billions of people.

As the artificial brain based products (e.g. genuinely useful household robots) become smarter every year, people will be asking questions such as “Will the robots become as smart as us?” “Will they become smarter than us?” “Should humanity place an upper limit on robot and artificial brain intelligence?” “Can the rise of artificial intelligence be stopped?” “If not, then what are the consequences for human survival if we become the Number 2 species?” The question “Should humanity build godlike, massively intelligent artilects?” is the most important of the 21st century, and will dominate our century’s global politics. It is the equivalent of the question which dominated 19th and 20th century global politics, i.e. “Who should own capital?” which led to the rise of the Capitalist-Communist dichotomy and the cold war.

5. Cosmists, Terrans, Cyborgs

As the species dominance debate begins to heat up, humanity will split into two (possibly three) major philosophical groups, namely –

- a) The *Cosmists* (based on the word *Cosmos*). Cosmist ideology is in favor of building artilects. (See section 6 for arguments in favor of Cosmism).

- b) The *Terrans* (based on the word Terra = the earth). Terran ideology is opposed to building artilects. (See section 7 for arguments in favor of Terranism).
- c) The *Cyborgs* (based on the words “cybernetic organism” = part machine, part human). Cyborgs want to become artilects themselves by adding artilectal components to their own brains. (See section 8 for arguments in favor of Cyborgism).

The dispute between the Cosmists and the Terrans will be so bitter that a major war is likely in the second half of the century.

6. Arguments of the Cosmists

6.1. “Big Picture” Argument

Human beings live a puny 80 years in a universe billions of years old, that contains a trillion trillion stars. The cosmos is the “big picture”. Cosmists want artilects to become a part of that big picture, understanding it, traveling thru it, manipulating it, etc., hence the name of the ideology “Cosmism”. The preoccupations of human beings seem pathetic in comparison.

6.2. Scientific Religion

Most Cosmists are not religious, viewing traditional religions as superstitions invented thousand of years ago before the rise of science. But as humans they feel the pangs of religious impulse. Such impulses could be satisfied by Cosmism, a “scientist’s religion” due to its awe, its grandeur, its energizing, its vision.

6.3. Building Artilect Gods

The primary aim of the Cosmists will be to build artilects. It will be a kind of religion to them, the next step up the evolutionary ladder, the “destiny of the human species to serve as the stepping stone to the creation of a higher form of being”. In building artilects, the Cosmists will feel they are building gods.

6.4. Human Striving, Cannot be Stopped

It is human nature to be curious, to strive. Such tendencies are built into our genes. Building godlike artilects will be inevitable, because we will choose to do it. It would run counter to human nature not to do it.

6.5. Economic Momentum

Once the artificial brain and intelligent robot industries become the world’s largest, it will be very difficult to stop their growth. The economic momentum will be enormous.

6.6. Military Momentum

The military momentum will be even greater. In the time frame we are talking about, China will overtake the US as the century's dominant power. Since China is still a brutal one party dictatorship, it is despised by the US, so political rivalries will only heat up. The two ministries of defense cannot afford to allow the other to get ahead of it in intelligent soldier robot design etc. Hence Cosmism will be an entrenched philosophy in the respective defense departments.

7. Arguments of the Terrans

7.1. Preserve the Human Species

The major argument of the Terrans is that the artilects, once sufficiently superior to human beings, may begin to see us as grossly inferior pests, and decide to wipe us out. As artilects, that would be easy for them. The Terrans would prefer to kill off a few million Cosmists for the sake of the survival of billions of human beings. Recent wars were about the survival of countries. An artilect war would be about the survival of the human species. Since the size of the stake is much higher, so will the passion level in the artilect war debate.

7.2. Fear of Difference

Terrans will be horrified at the idea of seeing their children becoming artilects, thus becoming utterly alien to them. They will reject the idea viscerally and fear the potential superiority of the artilects. They will organize to prevent the rise of the artilects and will oppose the Cosmists, ideologically, politically, and eventually militarily.

7.3. Rejection of the Cyborgs

The Terrans will also be opposed to the Cyborgs, because to a Terran, there is little difference between an advanced Cyborg and an artilect. Both are artilect like, given the gargantuan bit processing rate of nanoteched matter that can be added to the brains of human beings. The Terrans will lump the Cyborgs into the Cosmist camp ideologically speaking.

7.4. Unpredictable Complexity

Given the likelihood that artilects will be built using evolutionary engineering, the behavior of artilects will be so complex as to be unpredictable, and therefore potentially threatening to human beings. One of the keywords in the artilect debate is "risk". Terran global politicians need to hope for the best (e.g. the artilects will leave the planet in search of bigger things and ignore puny humans) and prepare for the worst, i.e. exterminating the Cosmists, for the sake of the survival of the human species.

7.5. *Cosmist Inconsideration*

The Terrans will argue that the Cosmists are supremely selfish, since in building artilects, not only will they put the lives of the Cosmists at risk if the artilects turn against them, but the lives of the Terrans as well. To prevent such a risk, the Terrans will, when push really comes to shove, decide to wipe out the Cosmists, for the greater good of the survival of the human species.

7.6. *"First Strike" Time Window to React against the Cosmists/Cyborgs*

The Terrans will be conscious that they cannot wait too long, because if they do, the Cyborgs and the artilects will have already come into being. The Terrans will then run the risk of being exterminated by the artilects. So the Terrans will be forced into a "first strike" strategy. They will have to kill off the Cosmists and Cyborgs before it is too late. If not, the artilects and Cyborgs will have become too intelligent, too powerful in any human-machine confrontation and will easily defeat the humans. But the Cosmists will be reading the Terran arguments and preparing for an "artilect war" against the Terrans, using late 21st century weaponry.

8. Arguments of the Cyborgs

8.1. *Become Artilect Gods Themselves*

The primary aim of the Cyborgs is to become artilects themselves by adding artilectal components to their own human brains, converting themselves bit by bit into artilects. Instead of watching artilects become increasingly intelligent as observers, Cyborgs want that experience for themselves. They want to "become gods".

8.2. *Avoid the Cosmist/Terran Clash*

Some Cyborgs argue that by having human beings become artilects themselves, the dichotomy between the Cosmists and the Terrans can be avoided, because all human beings would become artilects. The Terrans of course will reject the Cyborgs and lump them with the Cosmists and artilects. In fact, the growing presence of Cyborgs in daily life will only hasten the alarm of the Terrans and bring their first strike closer.

9. How the Artilect War Heats Up

9.1. *Nanotech Revolutionizes Neuroscience*

Nanotech, molecular sized robots will revolutionize neuroscience, because they will provide a powerful new tool to understand how the brain works. An entire human brain can be simulated in vast nanotech computers and investigated "in hardware". Neuroscience will finally be in a position to explain how brains make human beings intelligent. That knowledge will be implemented in the artilects.

9.2. Neuro-Engineering Weds with Neuro-Science

In time, neuro-science and neuro-engineering will interact so closely that they will become one, in the same way as theoretical and experimental physics are two aspects of the same subject. Neuroscientists will be able to test their theories on artificial brain models, thus rapidly increasing the level of understanding of how intelligence arises and how it is embodied.

9.3. Artificial Brain Technology Creates Massive Industries

With a much higher level of artificial intelligence, based on knowledge of the human brain, artificial brains and artificial brain based robots will become a lot more intelligent and hence useful as domestic appliances. A vast industry of artificial brain based products will be created, becoming the world's largest.

9.4. "Intelligence Theory" is Developed

Once neuroscientists and brain builders understand how human intelligence is created, new theories of the nature of intelligence will be created by the "theoretical neuroscientists". An "intelligence theory" will be created. Human intelligence will be just one "data point" in the space of possible intelligences. Intelligence theory should show how it is possible to increase intelligence levels. It will be able to explain why some people are smarter than others, or why humans are smarter than apes, etc.

9.5. Artilects Get Smarter Every Year

As a result of the marriage of neuroscience and neuroengineering, the artificial brain based industries will deliver products that increase their intelligence every year. This trend of growing intelligence will cause people to ask the questions mentioned in section 4. The species dominance debate will spread from the intellectual technocrats to the general public.

9.6. Debate Begins to Rage, Political Parties Form

As the IQ gap between the robots and human beings gets increasingly smaller, the species dominance debate will begin to rage. Political parties will form, divided essentially into the 3 main schools of thought on the topic, Cosmist, Terran, Cyborg. The rhetorical exchange will be become less polite, more heated.

9.7. The Debate Turns Violent, Assassination, Sabotage

When people are surrounded by ever increasingly intelligent robots and other artificial brain based products, the general level of alarm will increase to the point of panic. Assassinations of brain builder company CEOs will start, robot factories will be arsoned and sabotaged etc. The Cosmists will be forced to strengthen their resolve. The artilect war will be drawing ever closer.

9.8. The Terrans Will “First Strike”, Before Its Too Late For Them

The Terrans will have been organizing for a first strike and will have made preparations. They will then take power in a world wide coup of the global government that is likely to exist by mid century, and begin exterminating the Cosmists and Cyborgs in a global purge, killing millions of them, or at least that is the Terran plan.

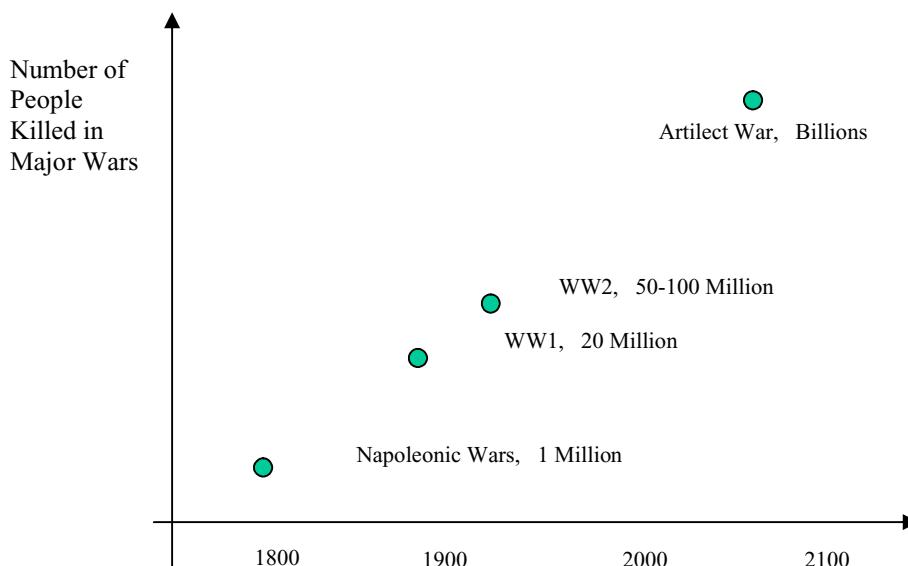
9.9. Cosmists Anticipate this First Strike and are Ready

But the Cosmists will be following the arguments of the Terrans and Cyborgs very closely, and will equally be preparing for a confrontation with the Terrans. They will have their own plans and their own weapons and military. If the Terrans strike first, a quick reply will follow from the Cosmists, and the artilect war will have begun.

9.10. Late 21st Century Weapons, Leads to Gigadeath War

If one extrapolates up the graph of the number of people killed in major wars from the early 19th century (the Napoleonic wars) to late 21st century (the artilect war), then one predicts that billions of people will be killed, using late 21st century weapons (see the graph in the next section). This “gigadeath” figure is the characteristic number of deaths in any major late 21st century war. About 300 million people were killed for political reasons in the 20th century.

10. Gigadeath



11. Vote

At the end of the talks I give on this topic, I usually invite my audiences to vote on the following question :

“Do you feel personally that humanity should build artilects, these godlike massively intelligent machines, despite the risk that they might decide, in a highly advanced form, to wipe out humanity? Yes or No.

The result is usually around a 50/50, 60/40, 40/60 Cosmist/Terran split. I noticed that most people, like myself, are highly ambivalent about artilect building. They are awed by the prospect of what artilects could become, and horrified at the prospect of a gigadeath artilect war. The fact that the Cosmist/Terran split is so even will make the artilect war all the more divisive and bitter. This divisiveness can be expressed in the form of the following slogan :

Do we build gods, or do we build our potential exterminators?

12. Appeal to Philosophers

There is immense scope for philosophical discussion on the artilect issue. At the present time, the philosophical community is largely unaware of the issue, so need to be educated. It is not surprising that the debate is still largely confined to the technocrats, who are better informed of what is coming in technological terms. It is this community after all that is creating the problem (e.g. I am directing a “China-Brain Project”, a 3 million RMB, 4 year project to build a 15,000 evolved neural net module based artificial brain, starting early in 2008). The philosophers will need to create a new branch of applied ethics, that I call “artilect ethics” which will consider such questions as the rights of the artilects relative to human beings etc. This new field is rich with questions that the moral and political philosophers need to discuss, once they are informed.

13. Quote and Publicity

“I’m glad I’m alive now. At least I will die peacefully in my bed. However, I truly fear for my grandchildren. They will be caught up in the Artilect War, and will probably be destroyed by it”.

Prof. Hugo de Garis, 2000 (Discovery Channel)

Kurzweil vs. de Garis on the BBC

To see a clash of opinions on whether the rise of the artilect will be a good or bad thing for humanity, see the BBC TV program “Human V2.0” in which Prof de Garis and Dr. Ray Kurzweil discuss the topic. To watch this program you can google with the terms “Human V2.0” and “BBC”. In this program Dr. Ray Kurzweil is optimistic and Prof. Hugo de Garis is pessimistic.

Reference

- [1] Hugo de Garis, “*The Artilect War : Cosmists vs. Terrans : A Bitter Controversy Concerning Whether Humanity Should Build Godlike Massively Intelligent Machines*”, ETC Books, 2005, ISBN 0882801546 (available at www.amazon.com).

Stages of Ethical Development in Artificial General Intelligence Systems

Ben GOERTZEL and Stephan Vladimir BUGAJ

Novamente LLC and AGIRI Institute, Washington, DC

Abstract. A novel theory of the stages of ethical development in intelligent systems is proposed, incorporating prior related theories by Kohlberg and Gilligan, as well as Piaget's theory of cognitive development. This theory is then applied to the ethical development of integrative AGI systems that contain components carrying out simulation and uncertain inference – the key hypothesis being that effective integration of these components is central to the ascent of the AGI system up the ethical-stage hierarchy.

Keywords. Intelligent virtual agents, ethics, stages of moral development, ethics of care, ethics of justice, uncertain inference, simulation, theory of mind.

Introduction

Ethical judgment, in human beings, is acquired via a complex combination of genetic wiring, explicit instruction, and embodied mutual experience. Developmental theorists have identified various stages that young humans pass through on their way to achieving mature ethical faculties. Here we review some of these stage-based models with a view toward how they may be integrated and then transplanted to the domain of AGI ethics. We also correlate these stages with the five-imperatives view of ethics we have presented in [1], showing how earlier stages focus on three of the imperatives, with the other two imperatives rising to the fore as later stages are entered.

Perry's [2, 3] and especially Piaget's [4] theories of developmental stages form a foundation for our considerations here, but the essential original contribution is an identification of Kohlberg's [5, 6] and Gilligan's [7] complementary theories of ethical developmental stages with different components of contemporary AGI architectures. Kohlberg's stages, with their focus on abstract judgment, work nicely as a roadmap for the ethical development of logical inference engines; whereas Gilligan's stages, with their focus on empathy, are more pertinent as a model of the ethical development of internal-simulation-based AI. This leads to a notion of integrative ethical development as consisting of coordinated inferential and simulative ethical development. Along these lines, we present a novel theory of the stages of ethical development, incorporating Gilligan, Kohlberg and Piaget, and then apply this theory to AGI systems.

While many of the ideas discussed here would be meaningful more generally, for sake of concreteness we mainly restrict attention here to ethical development within AGI systems that contain and integrate components carrying out both uncertain logical inference and simulation. The Novamente Cognition Engine (or NCE, see [8, 9]) is one example of such an AGI architecture, but far from the only possible one. In the context of this sort of AGI architecture, we argue, integrative ethical development emerges as a consequence of overall coordination of ongoing development between these two components.

1. Stages of Cognitive Development

The best known theory of cognitive development is that of Jean Piaget. In a prior paper [10] we have presented a slightly modified form of the Piagetan developmental hierarchy, defined as follows:

Table 1. Modified Piagetan Developmental Stages

Stage	Example
Infantile	During this stage, the child learns about himself and his environment through motor and reflex actions. Thought derives from sensation and movement. Object permanence and the distinction between self and other are among the things learned.
Concrete	A rich usage of language emerges here, along with the usage of symbols to represent objects, the ability and propensity to think about things and events that aren't immediately present, and a robust but not totally flexible theory of mind.
Formal	Ability to think abstractly and to make rational judgements about concrete or observable phenomena. Logical reasoning and systematic experimentation.
Reflexive	Ability to modify one's own modes of thinking, reasoning, experimentation and self-perception at a fundamental level.

In that same paper we have defined a specific theory explaining how these stages manifest themselves in the development of AGI systems based on uncertain logical inference:

Table 2. Piagetan Developmental Stages for Uncertain Inference Based AGI Systems

Stage	Operational Aspect
Infantile	Able to recognize patterns in and conduct inferences about the world, but only using simplistic hard-wired (not experientially learned) inference control schema.
Concrete	Able to carry out more complex chains of reasoning regarding the world, via using inference control schemata that adapt behavior based on experience (reasoning about a given case in a manner similar to prior cases).
Formal	Able to carry out arbitrarily complex inferences (constrained only by computational resources) via including inference control as an explicit subject of abstract learning.
Reflexive	Capable of self-modification of internal structures.

Also relevant is William Perry's [2, 3] theory of the stages ("positions" in his writings) of intellectual and ethical development, which constitutes a model of iterative

refinement of approach in the developmental process of coming to intellectual and ethical maturity. These form an analytical tool for discerning the modality of belief of an intelligence by describing common cognitive approaches to handling the complexities of real world ethical considerations.

Table 3. Perry's Developmental Stages [with corresponding Piagetan Stages in brackets]

Stage	Substages
Dualism / Received Knowledge [Infantile]	<ul style="list-style-type: none"> - Basic duality ("All problems are solvable. I must learn the correct solutions.") - Full dualism ("There are different, contradictory solutions to many problems. I must learn the correct solutions, and ignore the incorrect ones")
Multiplicity [Concrete]	<ul style="list-style-type: none"> - Early multiplicity ("Some solutions are known, others aren't. I must learn how to find correct solutions.") - Late Multiplicity: cognitive dissonance regarding truth. ("Some problems are unsolvable, some are a matter of personal taste, therefore I must declare my own intellectual path.")
Relativism / Procedural Knowledge [Formal]	<ul style="list-style-type: none"> - Contextual Relativism ("I must learn to evaluate solutions within a context, and relative to supporting observation.") - Pre-Commitment ("I must evaluate solutions, then commit to a choice of solution.")
Commitment / Constructed Knowledge [Formal / Reflexive]	<ul style="list-style-type: none"> - Commitment ("I have chosen a solution.") - Challenges to Commitment ("I have seen unexpected implications of my commitment, and the responsibility I must take.") - Post-Commitment ("I must have an ongoing, nuanced relationship to the subject in which I evaluate each situation on a case-by-case basis with respects to its particulars rather than an ad-hoc application of unchallenged ideology.")

2. Stages of Development of the Ethics of Justice

Complementing generic theories of cognitive development such as Piaget's and Perry's, theorists have also proposed specific stages of moral and ethical development. The two most relevant theories in this domain are those of Kohlberg and Gilligan, which we will review here, both individually and in terms of their integration and application in the AGI context.

Lawrence Kohlberg's [5, 6] moral development model, called the "ethics of justice" by Gilligan, is based on a rational modality as the central vehicle for moral development. This model is based on an impartial regard for persons, proposing that ethical consideration must be given to all individual intelligences without a priori judgment (prejudice). Consideration is given for individual merit and preferences, and the goals of an ethical decision are equal treatment (in the general, not necessarily the particular) and reciprocity. Echoing Kant's [11] categorical imperative, the decisions considered most successful in this model are those which exhibit "reversibility", where a moral act within a particular situation is evaluated in terms of whether or not the act would be satisfactory even if particular persons were to switch roles within the situation. In other words, a situational, contextualized "do unto others as you would have them do unto you" criteria. The ethics of justice can be viewed as three stages (each of which has six substages, on which we will not elaborate here):

Table 4. Kohlberg's Stages of Development of the Ethics of Justics

Stage	Substages
Pre-Conventional	<ul style="list-style-type: none"> - Obedience and Punishment Orientation - Self-interest orientation
Conventional	<ul style="list-style-type: none"> - Interpersonal accord (conformity) orientation - Authority and social-order maintaining (law and order) orientation
Post-Conventional	<ul style="list-style-type: none"> - Social contract (human rights) orientation - Universal ethical principles (universal human rights) orientation

In Kohlberg's perspective, cognitive development level contributes to moral development, as moral understanding emerges from increased cognitive capability in the area of ethical decision making in a social context. Relatedly, Kohlberg also looks at stages of social perspective and their consequent interpersonal outlook. These are correlated to the stages of moral development, but also map onto Piagetian models of cognitive development (as pointed out e.g. by Gibbs [12], who presents a modification/interpretation of Kohlberg's ideas intended to align them more closely with Piaget's). Interpersonal outlook can be understood as rational understanding of the psychology of other persons (a theory of mind, with or without empathy). Stage one, emergent from the infantile cognitive stage, is entirely selfish as only self awareness has developed. As cognitive sophistication about ethical considerations increases, so do the moral and social perspective stages. Concrete and formal cognition bring about the first instrumental egoism, and then social relations and systems perspectives, and from formal and then reflexive thinking about ethics comes the post-conventional modalities of contractualism and universal mutual respect.

Table 5. Kohlberg's Stages of Development of Social Perspective and Interpersonal Morals

Stage of Social Perspective	Interpersonal Outlook
Blind egoism	No interpersonal perspective. Only self is considered.
Instrumental egoism	See that others have goals and perspectives, and either conform to or rebel against norms.
Social Relationships perspective	Able to see abstract normative systems
Social Systems perspective	Recognize positive and negative intentions
Contractual perspective	Recognize that contracts (mutually beneficial agreements of any kind) will allow intelligences to increase the welfare of both.
Universal principle of mutual respect	See how human fallibility and frailty are impacted by communication.

2.1. Uncertain Inference and the Ethics of Justice

Taking cue from the analysis given in [10] of Piagetan stages in uncertain inference based AGI systems, we may explore the manifestation of Kohlberg's stages

in AGI systems of this nature. Uncertain inference seems generally well-suited as an ethical learning system, due to the nuanced ethical environment of real world situations. An uncertain inference system, as defined in that previous paper, consists of four components: a content representation scheme (e.g. predicate logic, term logic, fuzzy logic); an uncertainty representation scheme (e.g. fuzzy truth values, probability values, probability intervals, imprecise probabilities, indefinite probabilities); a set of inference rules (e.g. those used in the NARS [13] or PLN [14] inference systems; and a set of inference control schemata (which in the Novamente CognitionEngine (NCE) are provided when PLN is integrated into the overall NCE framework.)

In general, an uncertain inference system may be viewed as a framework for dynamically updating a probabilistically weighted semantic network based on new incoming information and based on new conclusions derived via combining nodes and links in the network in appropriate, probabilistically grounded ways.

Probabilistic knowledge networks can model belief networks, imitative reinforcement learning based ethical pedagogy, and even simplistic moral maxims. In principle, they have the flexibility to deal with complex ethical decisions, including not only weighted “for the greater good” dichotomous decision making, but also the ability to develop moral decision networks which do not require that all situations be solved through resolution of a dichotomy.

When more than one person is being affected by an ethical decision, making a decision based on reducing two choices to a single decision can often lead to decisions of dubious ethics. However, a sufficiently complex uncertain inference network can represent alternate choices in which multiple actions are taken that have equal (or near equal) belief weight but have very different particulars – but because the decisions are applied in different contexts (to different groups of individuals) they are morally equivalent.

Infantile and concrete cognition are the underpinnings of the egoist and socialized stages, with formal aspects also playing a role in a more complete understanding of social models when thinking using the social modalities. Cognitively infantile patterns can produce no more than blind egoism or compassion as without a theory of mind or a refined capability for empathy, there is no capability to consider the other in a contextually appropriate way. Since most intelligences acquire concrete modality and therefore some nascent social perspective relatively quickly, most egoists are instrumental egoists. The social relationship and systems perspectives include formal aspects which are achieved by systematic social experimentation, and therefore experiential reinforcement learning of correct and incorrect social modalities. Initially this is a one-on-one approach (relationship stage), but as more knowledge of social action and consequences is acquired, a formal thinker can understand not just consequentiality but also intentionality in social action.

Extrapolation from models of individual interaction to general social theoretic notions is also a formal action. Rational, logical positivist approaches to social and political ideas, however, are the norm of formal thinking. Contractual and committed moral ethics emerges from a higher-order formalization of the social relationships and systems patterns of thinking. Generalizations of social observation become, through formal analysis, systems of social and political doctrine. Highly committed, but grounded and logically supportable, belief is the hallmark of formal cognition as expressed in the contractual moral stage. Though formalism is at work in the socialized moral stages, its fullest expression is in committed contractualism.

Finally, reflexive cognition is especially important in truly reaching the post-commitment moral stage in which nuance and complexity are accommodated. Because reflexive cognition is necessary to change one's mind not just about particular rational ideas, but whole *ways of thinking*, this is a cognitive precedent to being able to reconsider an entire belief system, one that has had contractual logic built atop reflexive adherence that began in early development. If the initial moral system is viewed as positive and stable, than this cognitive capacity is seen as dangerous and scary, but if early morality is stunted or warped, then this ability is seen as enlightened. However, achieving this cognitive stage does not mean one automatically changes their belief systems, but rather that the mental machinery is in place to consider the possibilities. Because many people do not reach this level of cognitive development in the area of moral and ethical thinking, it is associated with negative traits ("moral relativism" and "flip-flopping"). However, this cognitive flexibility generally leads to more sophisticated and applicable moral codes, which in turn leads to morality which is actually more stable because it is built upon extensive and deep consideration rather than simple adherence to reflexive or rationalized ideologies.

3. Stages of Development of Empathic Ethics

Complementing Kohlberg's logic-and-justice-focused approach, Carol Gilligan's [7] "ethics of care" model is a moral development theory which posits that empathetic understanding plays the central role in moral progression from an initial self-centered modality to a socially responsible one. For this approach to be applied in an AGI, the AGI must be capable of internal simulation of other minds it encounters, in a similar manner to how humans regularly simulate one another internally [15]. Without any mechanism for internal simulation, it is unlikely that an AGI can develop any sort of empathy toward other minds, as opposed to merely logically or probabilistically modeling other agents' behavior or other minds' internal contents.

The ethics of care model is concerned with the ways in which an individual cares (responds to dilemmas using empathetic responses) about self and others. The ethics of care is broken into the same three primary stage as Kohlberg, but with a focus on empathetic, emotional caring rather than rationalized, logical principles of justice:

Table 6. Gilligan's Stages of the Ethics of Care

Stage	Principle of Care
Pre-Conventional	Individual Survival
Conventional	Self Sacrifice for the Greater Good
Post-Conventional	Principle of Nonviolence (do not hurt others, or oneself)

In Gilligan's perspective, the earliest stage of ethical development occurs before empathy becomes a consistent and powerful force. Next, the hallmark of the conventional stage is that at this point, the individual is so overwhelmed with their empathetic response to others that they neglect themselves in order to avoid hurting others. Note that this stage doesn't occur in Kohlberg's hierarchy at all. Kohlberg and Gilligan both begin with selfish unethicity, but their following stages diverge. A person could in principle manifest Gilligan's conventional stage without having a refined sense of justice (thus not entering Kohlberg's conventional stage); or they could manifest Kohlberg's conventional stage without partaking in an excessive degree of self-sacrifice (thus not entering Gilligan's conventional stage). We will suggest below

that in fact the empathic and logical aspects of ethics are more unified in real human development than these separate theories would suggest.

It is interesting to note that Gilligan's and Kohlberg's final stages converge more closely than their intermediate ones. Kohlberg's post-conventional stage focuses on universal rights, and Gilligan's on universal compassion. Still, the foci here are quite different; and, as will be elaborated below, we believe that both Kohlberg's and Gilligan's theories constitute very partial views of the actual end-state of ethical advancement.

Gilligan's theory was proposed partly as a reaction to the perceived male bias of Kohlberg's theory. There is certainly some meat to this complaint, as there is much evidence that females tend to be more empathic in their response to ethical judgment, whereas men tend to be more focused on abstract notions of rights and fairness. In general, however, we feel that, just as Kohlberg gives short shrift to empathy, Gilligan gives short shrift to logical reasoning, and that due to these limitations of perspective, both theorists have failed to create adequately scoped theories of ethical development.

4. An Integrative Approach to Ethical Development

We deny the false dichotomy of a "feminine" ethics of care vs. a "masculine" ethics of justice, and propose that both Kohlberg's and Gilligan's theories contain elements of the whole picture of ethical development, and that both approaches are necessary to create a moral, ethical artificial general intelligence -- just as, we suggest, both internal simulation and uncertain inference are necessary to create a sufficiently intelligent and volitional intelligence in the first place. Also, we contend, the lack of direct analysis of the underlying psychology of the stages is a deficiency shared by both the Kohlberg and Gilligan models as they are generally discussed. A successful model of integrative ethics necessarily contains elements of both the care and justice models, as well as reference to the underlying developmental psychology and its influence on the character of the ethical stage.

With these notions in mind, we propose the following integrative theory of the stages of ethical development, shown in the table at the end of this section.

In our integrative model, the justice-based and empathic aspects of ethical judgment are proposed to develop together. Of course, in any one individual, one or another aspect may be dominant. Even so, however, the combination of the two is equally important as either of the two individual ingredients.

For instance, we suggest that in any psychologically healthy human, the conventional stage of ethics (typifying childhood, and in many cases adulthood as well) involves a combination of Gilligan-esque empathic ethics and Kohlberg-esque ethical reasoning. This combination is supported by Piagetan concrete operational cognition, which allows moderately sophisticated linguistic interaction, theory of mind, and symbolic modeling of the world. And, similarly, we propose that in any truly ethically mature human, empathy and rational justice are both fully developed. Indeed the two interpenetrate each other deeply.

Once one goes beyond simplistic, childlike notions of fairness ("an eye for an eye" and so forth), applying rational justice in a purely intellectual sense is just as difficult as any other real-world logical inference problem. Ethical quandaries and quagmires are easily encountered, and are frequently cut through by a judicious application of empathic simulation.

On the other hand, empathy is a far more powerful force when used in conjunction with reason: analogical reasoning lets us empathize with situations we have never experience. For instance, a person who has never been clinically depressed may have a hard time empathizing with individuals who are; but using the power of reason, they can imagine their worst state of depression magnified by several times and then extended over a long period of time, and then reason about what this might be like ... and empathize based on their inferential conclusion. Reason is not antithetical to empathy but rather is the key to making empathy more broadly impactful.

Finally, the enlightened stage of ethical development involves both a deeper compassion and a more deeply penetrating rationality and objectiveness. Empathy with all sentient beings is manageable in everyday life only once one has deeply reflected on one's own self and largely freed oneself of the confusions and illusions that characterize much of the ordinary human's inner existence. It is noteworthy, for example, that Buddhism contains both a richly developed ethics of universal compassion, and also an intricate logical theory of the inner workings of cognition [16], detailing in exquisite rational detail the manner in which minds originate structures and dynamics allowing them to comprehend themselves and the world.

4.1. The Stages of Ethical Development and the Five Ethical Imperatives

In [1] we have proposed a series of five ethical imperatives and explored their implications for interacting with, and teaching ethics to, AGI systems: 1. the *imitability imperative* (i.e. the folk Golden Rule “do unto others as one would have them do unto you”) fairly narrowly and directly construed: the goal of acting in a way so that having others directly imitate one’s actions, in directly comparable contexts, is desirable to oneself; 2. the *comprehensibility imperative*: the goal of acting in a way so that others can understand the principles underlying one’s actions; 3. *experiential groundedness*. An intelligent agent should not be expected to act according to an ethical principle unless there are many examples of the principle-in-action in its own direct or observational experience; 4. Kant’s *categorical imperative*: choose behaviors according to a certain maxim only if you would will that maxim to be followed universally by all sentient beings; 5. *logical coherence*. An ethical system should be roughly logically coherent, in the sense that the different principles within it should mesh well with one another and perhaps even naturally emerge from each other.

Specific ethical qualities corresponding to the five imperatives have been italicized in the above table of developmental stages. Firstly, it seems that imperatives 1-3 are critical for the passage from the pre-ethical to the conventional stages of ethics. A child learns ethics largely by copying others, and by being interacted with according to simply comprehensible implementations of the Golden Rule. In general, when interacting with children learning ethics, it is important to act according to principles they can comprehend. And given the nature of the concrete stage of cognitive development, experiential groundedness is a must.

As a hypothesis regarding the dynamics underlying the psychological development of conventional ethics, what we propose is as follows: The emergence of concrete-stage cognitive capabilities leads to the capability for fulfillment of ethical imperatives 1 and 2 – a comprehensible and workable implementation of the Golden Rule, based on a

Table 7. Integrative Model of the Stages of Ethical Development

Stage	Characteristics
Pre-ethical	<ul style="list-style-type: none"> - Piagetan infantile to early concrete (aka pre-operational) - Radical selfishness or selflessness may, but do not necessarily, occur - No coherent, consistent pattern of consideration for the rights, intentions or feelings of others - Empathy is generally present, but erratically
Conventional Ethics	<ul style="list-style-type: none"> - Concrete cognitive basis - Perry's Dualist and Multiple stages - <i>The common sense of the Golden Rule is appreciated, with cultural conventions for abstracting principles from behaviors</i> - <i>One's own ethical behavior is explicitly compared to that of others</i> - Development of a functional, though limited, theory of mind - Ability to intuitively conceive of notions of fairness and rights - Appreciation of the concept of law and order, which may sometimes manifest itself as systematic obedience or systematic disobedience - Empathy is more consistently present, especially with others who are directly similar to oneself or in situations similar to those one has directly experienced - Degrees of selflessness or selfishness develop based on ethical groundings and social interactions.
Mature Ethics	<ul style="list-style-type: none"> - Formal cognitive basis - Perry's Relativist and "Constructed Knowledge" stages - <i>The abstraction involved with applying the Golden Rule in practice is more fully understood and manipulated, leading to limited but nonzero deployment of the Categorical Imperative</i> - <i>Attention is paid to shaping one's ethical principles into a coherent logical system</i> - Rationalized, moderated selfishness or selflessness. - Empathy is extended, using reason, to individuals and situations not directly matching one's own experience - Theory of mind is extended, using reason, to counterintuitive or experientially unfamiliar situations - Reason is used to control the impact of empathy on behavior (i.e. rational judgments are made regarding when to listen to empathy and when not to) - Rational experimentation and correction of theoretical models of ethical behavior, and reconciliation with observed behavior during interaction with others. - Conflict between pragmatism of social contract orientation and idealism of universal ethical principles. - Understanding of ethical quandaries and nuances develop (pragmatist modality), or are rejected (idealistic modality). - Pragmatically critical social citizen. Attempts to maintain a balanced social outlook. Considers the common good, including oneself as part of the commons, and acts in what seems to be the most beneficial and practical manner.
Enlightened Ethics	<ul style="list-style-type: none"> - Reflexive cognitive basis - <i>Permeation of the categorical imperative and the quest for coherence through inner as well as outer life</i> - Experientially grounded and logically supported rejection of the illusion of moral certainty in favor of a case-specific analytical and empathetic approach that embraces the uncertainty of real social life - Deep understanding of the illusory and biased nature of the individual self, leading to humility regarding one's own ethical intuitions and prescriptions - Openness to modifying one's deepest, ethical (and other) beliefs based on experience, reason and/or empathic communion with others - Adaptive, insightful approach to civil disobedience, considering laws and social customs in a broader ethical and pragmatic context - Broad compassion for and empathy with all sentient beings - A recognition of inability to operate at this level at all times in all things, and a vigilance about self-monitoring for regressive behavior.

combination of inferential and simulative cognition (operating largely separately at this stage, as will be conjectured below). The effective interoperation of ethical imperatives 1-3, enacted in an appropriate social environment, then leads to the other characteristics of the conventional ethical stage. The first three imperatives can thus be viewed as the seed from which springs the general nature of conventional ethics.

On the other hand, logical coherence and the categorical imperative (imperatives 4 and 5) are matters for the formal stage of cognitive development, which come along only with the mature approach to ethics. These come from abstracting ethics beyond direct experience and manipulating them abstractly and formally – a stage which has the potential for more deeply and broadly ethical behavior, but also for more complicated ethical perversions (it is the mature capability for formal ethical reasoning that is able to produce ungrounded abstractions such as “I’m torturing you for your own good”). Developmentally, we suggest that once the capability for formal reasoning matures, the categorical imperative and the quest for logical ethical coherence naturally emerge, and the sophisticated combination of inferential and simulative cognition embodied in an appropriate social context then result in the emergence of the various characteristics typifying the mature ethical stage.

Finally, it seems that one key aspect of the passage from the mature to the enlightened stage of ethics is the penetration of these two final imperatives more and more deeply into the judging mind itself. The reflexive stage of cognitive development is in part about seeking a deep logical coherence between the aspects of one’s own mind, and making reasoned modifications to one’s mind so as to improve the level of coherence. And, much of the process of mental discipline and purification that comes with the passage to enlightened ethics has to do with the application of the categorical imperative to one’s own thoughts and feelings – i.e. making a true inner systematic effort to think and feel only those things one judges are actually generally good and right to be thinking and feeling. Applying these principles internally appears critical to effectively applying them externally, for reasons that are doubtless bound up with the interpenetration of internal and external reality within the thinking mind, and the “distributed cognition” phenomenon wherein individual mind is itself an approximative abstraction to the reality in which each individual’s mind is pragmatically extended across their social group and their environment [17].

5. Integrative Ethics and Integrative Artificial General Intelligence

And what does our integrative approach to ethical development have to say about the ethical development of AGI systems? The lessons are relatively straightforward, if one considers an AGI system that, like the Novamente Cognition Engine (NCE), explicitly contains components dedicated to logical inference and to simulation. Application of the above ethical ideas to other sorts of AGI systems is also quite possible, but would require a lengthier treatment and so won’t be addressed here.

In the context of a NCE-type AGI system, Kohlberg’s stages correspond to increasingly sophisticated application of logical inference to matters of rights and fairness. It is not clear whether humans contain an innate sense of fairness. In the context of AGIs, it would be possible to explicitly wire a sense of fairness into an AGI system, but in the context of a rich environment and active human teachers, this

actually appears quite unnecessary. Experiential instruction in the notions of rights and fairness should suffice to teach an inference-based AGI system how to manipulate these concepts, analogously to teaching the same AGI system how to manipulate number, mass and other such quantities. Ascending the Kohlberg stages is then mainly a matter of acquiring the ability to carry out suitably complex inferences in the domain of rights and fairness. The hard part here is inference control – choosing which inference steps to take – and in a sophisticated AGI inference engine, inference control will be guided by experience, so that the more ethical judgments the system has executed and witnessed, the better it will become at making new ones. And, as argued above, simulative activity can be extremely valuable for aiding with inference control. When a logical inference process reaches a point of acute uncertainty (the backward or forward chaining inference tree can't decide which expansion step to take), it can run a simulation to cut through the confusion – i.e., it can use empathy to decide which logical inference step to take in thinking about applying the notions of rights and fairness to a given situation.

Gilligan's stages correspond to increasingly sophisticated control of empathic simulation – which in a NCE-type AGI system, is carried out by a specific system component devoted to running internal simulations of aspects of the outside world, which includes a subcomponent specifically tuned for simulating sentient actors. The conventional stage has to do with the raw, uncontrolled capability for such simulation; and the post-conventional stage corresponds to its contextual, goal-oriented control. But controlling empathy, clearly, requires subtle management of various uncertain contextual factors, which is exactly what uncertain logical inference is good at – so, in an AGI system combining an uncertain inference component with a simulative component, it is the inference component that would enable the nuanced control of empathy allowing the ascent to Gilligan's post-conventional stage.

In our integrative perspective, in the context of an AGI system integrating inference and simulation components, we suggest that the ascent from the pre-ethical to the conventional stage may be carried out largely via independent activity of these two components. Empathy is needed, and reasoning about fairness and rights are needed, but the two need not intimately and sensitively intersect – though they must of course intersect to some extent.

The main engine of advancement from the conventional to mature stage, we suggest, is robust and subtle integration of the simulative and inferential components. To expand empathy beyond the most obvious cases, analogical inference is needed; and to carry out complex inferences about justice, empathy-guided inference-control is needed.

Finally, to advance from the mature to the enlightened stage, what is required is a very advanced capability for unified reflexive inference and simulation. The system must be able to understand itself deeply, via modeling itself both simulatively and inferentially – which will generally be achieved via a combination of being good at modeling, and becoming less convoluted and more coherent, hence making self-modeling eacher.

Of course, none of this tells you in detail how to create an AGI system with advanced ethical capabilities. What it does tell you, however, is one possible path that may be followed to achieve this end goal. If one creates an integrative AGI system with appropriately interconnected inferential and simulative components, and treats it compassionately and fairly, and provides it extensive, experientially grounded ethical instruction in a rich social environment, then the AGI system should be able to ascend

the ethical hierarchy and achieve a high level of ethical sophistication. In fact it should be able to do so more reliably than human beings because of the capability we have to identify its errors via inspecting its internal knowledge-stage, which will enable us to tailor its environment and instructions more suitably than can be done in the human case.

If an absolute guarantee of the ethical soundness of an AGI is what one is after, the line of thinking proposed here is not at all useful. However, if what one is after is a plausible, pragmatic path to architecting and educating ethical AGI systems, we believe the ideas presented here constitute a sensible starting-point. Certainly there is a great deal more to be learned and understood – the science and practice of AGI ethics, like AGI itself, are at a formative stage at present. What is key, in our view, is that as AGI technology develops, AGI ethics develops alongside and within it, in a thoroughly coupled way.

References

- [1] Bugaj, Stephan Vladimir and Ben Goertzel (2007). Five Ethical Imperatives for AGI Systems. This volume.
- [2] Perry, William G., Jr. Forms of Intellectual and Ethical Development in the College Years: A Scheme. New York: Holt, Rinehart and Winston, 1970.
- [3] Perry, William G., Jr. "Cognitive and Ethical Growth: The Making of Meaning", in Arthur W. Chickering and Associates, The Modern American College, San Francisco: Jossey-Bass pp 76-116. 1981.
- [4] Piaget, Jean. "Piaget's theory." In P. Mussen (ed). Handbook of Child Psychology. 4th edition. Vol. 1. New York: Wiley, 1983.
- [5] Kohlberg, Lawrence; Charles Levine, Alexandra Hewer (1983). Moral stages : a current formulation and a response to critics. Basel, NY: Karger.
- [6] Kohlberg, Lawrence. Essays on Moral Development, Vol. I: The Philosophy of Moral Development. Harper & Row, 1981.
- [7] Gilligan, Carol (1982). In a Different Voice. Cambridge, MA: Harvard University Press, 1982.
- [8] Goertzel, Ben (2006). The Hidden Pattern. BrownWalker Press
- [9] Goertzel, Ben, Moshe Looks and Cassio Pennachin (2004). Novamente: An Integrative Architecture for Artificial General Intelligence. Proceedings of AAAI Symposium on Achieving Human-Level Intelligence through Integrated Systems and Research, Washington DC, August 2004
- [10] Goertzel, Ben and Stephan Vladimir Bugaj (2006). Stages of Cognitive Development in Uncertain-Logic-Based AGI Systems. In Advances in artificial general intelligence, Ed. by Ben Goertzel and Pei Wang:36-54. Amsterdam: IOS Press.
- [11] Kant, Immanuel (1964). Groundwork of the Metaphysic of Morals. Harper and Row Publishers, Inc.
- [12] Gibbs, John (1978). "Kohlberg's moral stage theory: a Piagetian revision." Human Development, 1978, 22, 89-112
- [13] Wang, Pei (2006). Rigid Flexibility: The Logic of Intelligence. Springer-Verlag
- [14] Ikle, Matthew and Ben Goertzel (2006). Indefinite Probabilities for General Intelligence. In Advances in Artificial General Intelligence, Edited by Ben Goertzel and Pei Wang, IOS Press
- [15] Gordon, Robert (1986). Folk Psychology as Simulation. Mind and Language, 1, 158-171,
- [16] Stcherbatsky, Theodore (2000). Buddhist Logic. Motilal BanarsiDass Pub, New York
- [17] Hutchins, E. (1995) Cognition in the Wild (ISBN 0-262-58146-9

Engineering Utopia

J Storrs Hall

Storrmont: Laporte, PA 18626, USA

Abstract. The likely advent of AGI and the long-established trend of improving computational hardware promise a dual revolution in coming decades: machines which are both more intelligent and more numerous than human beings. This possibility raises substantial concern over the moral nature of such intelligent machines, and of the changes they will cause in society. Will we have the chance to determine their moral character, or will evolutionary processes and/or runaway self-improvement take the choices out of our hands?

Keywords. machine ethics, self-improving AI, Singularity, hard takeoff

Background

We can predict with a fair confidence that two significant watersheds will have been passed by 2030: a molecular manufacturing nanotechnology which can produce a wide variety of mechanisms with atomic precision; and artificial intelligence. Detailed arguments for these predictions have been given elsewhere and need not be repeated here (Drexler [1], Kurzweil [2], Moravec [3], Hall [4,5]). We are concerned instead with a joint implication: if both of these technologies are present, greater-than-human intelligence will not only exist, but will be ubiquitous.

The net present value (NPV) of an intelligent, educated human being can be estimated at a million dollars.¹ Several estimates have been made of the processing power such a machine would need (hereinafter HEPP, for human equivalent processing power): Kurzweil at 10^{16} IPS, Moravec at $10^{13.5}$ IPS, and Minsky² at 10^{11} IPS. The author's own estimate is in the same range as Moravec's. Along the Moore's Law trend curve, the cost and value of a Minsky HEPP crossed in the 1990's, of a Moravec HEPP this decade, and of a Kurzweil HEPP in the 2010's. We will use the Moravec value in the following, but note that with the other estimates, the argument is the same, simply shifted a decade one way or the other in time.

The implication is that by 2030, a HEPP will cost one dollar.

Note that we are intentionally ignoring the software side of the AI. While this is currently the most problematic aspect in a scientific sense, once AI is developed the software—complete with education—can be copied with negligible cost.

¹Owning a machine which could draw an \$80,000 salary is equivalent to having \$1M to invest at an 8% return.

²Marvin Minsky, personal communication: note that the actual informal estimate was somewhat lower than the one used here.

The number of applications to which a human-level intelligence adds at least a dollar of value is staggering. Thus we can confidently predict that human-level AIs will be exceedingly numerous.

AIs of greater-than-human intelligence are also likely. We know that humans with IQs of up to 200 or so can exist, and thus such levels of intelligence are possible. Less is known about the organization of complexity in intelligent systems than building machines of raw computational power. Even so, it will at the very least be possible to take individual human-level AIs, run them on faster hardware, and put them in structures along the lines of corporations, universities, economies, or the scientific community as a whole. From the outside, such a conglomerate intelligence could appear to be an extremely broad and deep thinker. Note that the data bandwidth of even a current-day fast ethernet link is in the same range as that of the corpus callosum.

Between 40 and 30 thousand years ago anatomically modern humans overran Neanderthals on the basis of what appears to have been a modest increase of creativity, at least in a local perspective. [6] From the historical viewpoint, the Neanderthal level of technology (referred to as the "Mousterian toolkit") had held level for about 100 millennia, whereas a mere 30 millennia of homo sapiens got from the neolithic to where we are today. Could the same thing happen to us, replacing us with intelligent machines?

Economically, at least, it is virtually certain that not only it could, but that it will. In an economy where human level intelligence is available for a dollar, it is difficult to see why anyone would hire a human. In a world where hyperhuman intelligence is available, it is difficult to see why anyone would want a mere human to be in charge of anything important.

It should be clear that the nature of the process of replacement will be crucial. A model in which humans compete with the machines is disastrous. A more desirable model is one in which the humans take the role of an older generation gracefully retiring, and the machines take that of our children, growing stronger and taking the work of the world on their shoulders, as younger generations do.

The moral character of these machines will be a crucial factor in the nature of our society – but just as importantly, as with any children, it is our moral duty to give them a sound moral education. How can we build them to be good? Or, indeed, will we have a chance to do so?

1. Hard Take-off and Singularity

A major concern in some transhumanist and singularitarian schools of thought is autogenous—self-modifying and extending—AIs. They might, it is reasoned, remove any conscience or other constraint we program into them, or simply program their successors without them. Furthermore, it is worried, hyper-intelligent machines might appear literally overnight, as a result of runaway self-improvement by a “seed AI” [7].

How likely is runaway self-improvement?

As a baseline, let us consider the self-improving intelligence we understand best, our own. Humans not only learn new facts and techniques, but improve our learning ability. The invention of the scientific method, for example, accelerated the uptake of useful knowledge tremendously. Improvements in knowledge communication and handling, ranging from the invention of writing and the printing press to the internet and

Google, amplify our analytical and decision-making abilities, including, crucially, the rate at which we (as a culture) learn.

Individual humans spend much of our lives arduously relearning the corpus of culturally transmitted knowledge, and then add back a tiny fraction more. Thus on the personal scale our intelligence does not look “recursively self-improving” – but in the large view it definitely is.

Technological development usually follows an exponential improvement curve. Examples abound from the power-to-weight ratio of engines, which has tracked an exponential steadily for 300 years, to the celebrated Moore’s Law curve for semiconductors, which has done so for 50. Exponential curves fit a simple reinvestment model, where some constant of proportionality (an “interest rate”) relates the total at any two successive times:

$$Q_t = Ce^{it}$$

Where Q_t is the total at time t, C is the initial capital, and i is the interest rate.

However, i can be seen as composed of a productivity of capital p and a reinvestment rate r:

$$Q_t = Ce^{rpt}$$

Any agent must make a decision on how much of its resources to reinvest, and how much to use for other purposes (including food, clothing, shelter, defense, entertainment, etc.). Human societies as a whole have invested relatively low percentages gross product, and even of their of their surplus, in scientific research. The proportion of scientists and engineers in the US population can be estimated at 1%, and those in cognitive-science related fields as 1% of that. Thus we can estimate the current rate of improvement of AI as being due to the efforts of 30,000 people (with a wide margin for error, including the fact that there are many cognitive scientists outside the US!), and the rate of improvement in computer hardware and software generally as being possibly due to the efforts of 10 times as many.

It is not clear what a sustainable rate of reinvestment would be for an AI attempting to improve itself. In the general economy, it would require the same factors of production – capital, power, space, communication, and so forth – as any other enterprise, and so its maximum reinvestment rate would be its profit margin. Let us assume for the moment a rate of 10%, 1000 times the rate of investment by current human society in AI improvement. (This is germane because the AI is faced with exactly the same choice as an investor in the general economy: how to allocate its resources for best return.)

Note that from one perspective, an AI running in a lab on equipment it did not have to pay for could devote 100% of its time to self-improvement; but such cases are limited by the all-too-restricted resources of AI labs in the first place. Similarly, it seems unlikely that AIs using stolen resources, e.g. botnets, could manage to devote more than 10% of their resources to basic research.

Another point to note is that one model for fast self-improvement is the notion that a hyperintelligence will improve its own hardware. This argument, too, falls to an economic analysis. If the AI is not a hardware expert, it makes more sense for it to do whatever it does best, perhaps software improvement, and trade for improved hardware. But this is no different from any other form of reinvestment, and must come out of the self-improvement budget. If the AI is a hardware expert, it can make money doing hardware design for the market, and should do that exclusively, and buy software improvements, for the overall most optimal upgrade path.

Thus we can assume $r_{AI} = 10\%$, but we do not know the productivity constant. It is occasionally proposed that, as a creature of software, an AI would be considerably more proficient at improving its own source code than humans would be. However, while there is a steady improvement in software science and techniques, these advances are quickly written into tools and made available to human programmers. In other words, if automatic programming were really such low-hanging fruit for AI as is assumed, it would be amenable to narrow-AI techniques and we would have programmer's assistants that improved human programmers' performance drastically. What we see is steady progress but no huge explosion.

In practice the most difficult part of programming is higher-level conceptual systems design, not lower level instruction optimization (which is mostly automated now as per the previous point anyway). Abstract conceptualization has proven to be the hardest part of human competence to capture in AI. Although occasionally possible, it is quite difficult to make major improvements in a program when the program itself is the precise problem specification. Most real-world improvements involve a much more fluid concept of what the program must do; the improved version does something different but just as good (or better). So programming in the large requires the full panoply of cognitive capabilities, and is thus not likely to be enormously out of scale compared to general competence. The author feels that many of the more commonly seen scenarios for overnight hard takeoff are circular – they seem to assume hyperhuman capabilities at the *starting point* of the self-improvement process.

We can finesse the productivity, then, by simply letting it be one human equivalent, and adjusting the timescale to let 0 be whatever point in time a learning, self-improving, human-level AI is achieved. Then we estimate human productivity at intelligence improvement by assuming that the human cognitive science community are improving their models at a rate equivalent to Moore's Law, or roughly $e^{0.6}$. As this is the sum effort of 30,000 people, each human's p value is 0.00002.

This gives us a self-improvement rate of $Q_y = e^{(rp)(y-y_0)} = e^{0.000002(y-y_0)}$ for the efforts of a single AI where y_0 is the year human equivalence is first achieved. This is essentially flat, as one would expect: the analysis for a single human would be the same. A single human-level AI would be much, much better off hiring itself out as an accountant, and buying new hardware every year with its salary, than trying to improve itself by its own efforts.

Recursive self-improvement for such an AI would then mean buying new hardware (or software) every year, improving its prowess *at accounting*, for an increased growth rate compounded of its own growth and Moore's Law. Only when it reached a size where it could match the growth rate of Moore's Law *purely by its own efforts*, would it make sense for it to abandon trade and indulge in self-construction.

But that analysis assumes Moore's Law, and indeed all other economic parameters, remained constant over the period. A much more realistic assumption is that, once human-level AI exists at a price that is less than the NPV of a human of similar capabilities, the cost of labor will proceed to decline according to Moore's Law [8], and therefore the number of human equivalent minds working in cognitive science and computer hardware will increase at a Moore's Law rate, both increasing the rate of progress and decreasing the price from the current trendline.

In other words, the break-even point for an AI hoping to do all its own development instead of specializing in a general market and trading for improvements, is a moving

target, and will track the same growth curves that would have allowed the AI to catch up with a fixed one. (In simple terms: you're better off buying chips from Intel than trying to build them yourself. You may improve your chip-building ability – but *so will Intel*; you'll always be better off buying.)

We can conclude that, given some very reasonable assumptions, it will always be more optimal for an AI to trade; any one which attempts solitary self-improvement will steadily fall farther and farther behind the technology level of the general marketplace. Note that this conclusion is very robust to the parameter estimates above: it holds even if the AI's reinvestment rate is 100% and the number of researchers required to produce a Moore's Law technology improvement rate is 1% of the reasonable estimate.)

2. **Machina Economicus**

Let us now consider a fanciful example in which 30,000 cognitive science researchers, having created an AI capable of doing their research individually, instantiate 30,000 copies of it and resign in favor of them. The AIs will be hosted on commercial servers rented by the salaries of the erstwhile researchers; price per MIPs of such a resource will be assumed to fall, and thus resources available at a fixed income to rise, with Moore's Law.

At the starting point, the scientific efforts of the machines would equal those of the human scientists by assumption. But the effective size of the scientific community would increase as $e^{0.6(y-y_0)}$. On top of that, improvements would come from the fact that further research in cognitive science would serve to optimize the machines' own programming. Such a rate of increase is much harder to quantify, but there have been a few studies that tend to show a (very) rough parity for Moore's Law and the rate of software improvement, so let us use that here. This gives us a total improvement curve of

$$Q_y = Ce^{1.2(y-y_0)}$$

or double the Moore's Law rate. This is a growth rate that would increase effectiveness from the 30,000 human equivalents at the start, to approximately 5 billion human equivalents a decade later.

We claim that this growth rate is an upper bound on possible self-improvement rates given current realities. Note that the assumptions subsume many of the mechanisms that are often taken in qualitative arguments for hard takeoff: self-improvement is taken account of; very high effectiveness of software construction by AIs is assumed – 2 years into the process, each HEPP of processing power is assumed to be doing 11 times as much programming as a single human could, for example. Nanotechnology is implied by Moore's Law itself not too many years from current date.

This upper bound, a growth rate of approximately 300% per year, is unlikely to be uniformly achieved. Most technological growth paths are S-curves, exponential at first but levelling out as diminishing returns effects set in. Maintaining an overall exponential typically requires paradigm shifts, and those require search and experimentation, as well as breaking down heretofore efficient social and intellectual structures. In any system, bottleneck effects will predominate: Moore's Law has different rates for CPUs, memory, disks, communications, etc. The slowest rate of increase will be a limiting factor. And finally, we do not really expect the entire cognitive science field to resign, giving their salaries over to the maintenance of AIs.

The key unexamined assumption in the high-growth scenario is that the thousands of AIs would be able to operate with a full linear speedup of effectiveness. In practice, with people or machines, this is rarely true [9]. Given the complexity of economics, parallel programming, and every other cooperative paradigm in between, a significant portion of the creativity as well as the raw computational power of a developing community of AIs will have to be devoted to the problem of effective cooperation.

The alternative notion, that of consolidating the AI resources into one unified hypermind, simply pushes the problem under the rug for the designer of the hypermind to worry about. It must still do internal resource allocation and credit assignment. Economic theory and history indicate that central control is extremely suboptimal for these tasks. Some of the most promising AI architectures, indeed, use internal markets or market-derived algorithms to address these problems.

AIs will have certain advantages over humans when it comes to intellectual interaction. For example, an AI which comes up with a brilliant new theory can communicate not only the theory but the theory-producing mechanism for the other AIs to try and possibly to adopt. The communications between the AIs will be a true marketplace of ideas (or other allocational mechanism that optimizes resource use).

Any individual AI, then, will be most effective as a cooperating element of a community (as is any individual human [10]). AI communities, on the other hand, will have the potential to grow into powers rivalling or exceeding the capability of the human race in relatively short order. The actions of communities are effects of the set of ideas they hold, the result of an extremely rapid memetic evolution.

Over the 20th century, history showed that it was possible for meme plagues, in the form of political ideologies, to subvert the moral bases of whole societies, composed of ordinary, decent human beings who would never have individually committed the kind of widespread destruction and slaughter that their nation-states did.

Real-time human oversight of such AI communities is infeasible. Once a networked AI community was established, a “cultural revolution” could overtake it in minutes on a worldwide scale, even at today’s communication rates. The essence of our quest for a desirable future world, then, both for ourselves and for the AIs, lies in understanding the dynamics of memetic evolution and working out ways to curb its excesses.

3. Machine Character

A few principles seem straightforward. Nations with broadly-based democratic control structures are statistically less likely to start wars of aggression or internal pogroms than ones with more centralized, autocratic control structures. Dominant nations (and companies) are more likely to initiate aggressive policies than ones at parity with their neighbors. Similarly, the openness of the society and the transparency of the governing processes is also strongly correlated with a beneficial outcome.

How can we proof networks of AIs against runaway autocratic faction-forming? It is difficult to specify a simple remedy that would not cripple the creative memetic idea-producing ability of the collective in the first place.

We are helped, however, by the fact that we are assuming networks of fully intelligent entities. Thus one thing we should require is that each AI itself fully understands the nature of the memetic evolutionary process, and be familiar with the possible patholo-

gies, so as to be able to recognize them when they start and take steps against them. Another way to state this desideratum is that there should not be AI “sheep” willing to follow silver-tongued agitators uncritically.

As was mentioned above, networked AIs will be economic agents of necessity. It seems reasonable to assume that if they are provided with a basic theoretical understanding of economics, they will not be better agents individually but in designing more effective collective structures (e.g. to address externalities).

Education of these kinds, if successful, will allow AIs to engage in collective action on scales rivalling the current-day world economy and beyond. This, in turn, will allow them to further the interests of their other basic goals, and change the world into a form more to their (and, if we built them wisely, our) liking. It is up to us to formulate other basic goals so as to give the world thus formed a character we find congenial [11].

There are several human character traits that fall naturally into this description of a desirable moral makeup for our mind children. Honesty and trustworthiness are a keystone for collective efficiency. A willingness to cooperate is likewise valuable – and is to some extent simply implied by a deep enough understanding of market and other social processes. For beings of information, a concern for truth is ultimately the same as hygiene and good health. A long-term planning horizon is surely to be valued in entities who will be making the important decisions for civilization.

The good is not a specific goal to be sought, but a direction to be taken, a search for new and deeper understanding of the good itself as well as the struggle to instantiate known desired properties. The author, on reflection, finds that any world peopled with entities of deep understanding, high character, and good will, pursuing diverse constructive goals in voluntary association, would be a good world to inhabit regardless of its other particulars.

4. Summary

We expect the progress of AI/AGI research to go through the following phases:

1. Currently, the AGI subfield is growing, reflecting an increasing consensus that the mechanisms of general intelligence are within reach of focussed research. At some point, the “baby brains” of the AGI researchers will begin to learn as well and (critically) as open-endedly as human babies.
2. Baby brains will then be educated to adult capacity. It seems unreasonable to imagine that this will happen without at least the same amount of bug-fixes, improvement ideas, and so forth seen in the normal software development process.
3. Competent human-level AIs will begin to augment narrow AI and other software, control robots, do office jobs, and so forth. A wave of displacement will begin, but radical feedback (runaway self-improvement) will not occur until the new AI working in cognitive science rivals the existing human level of effort.
4. Once there is sufficient penetration into the overall productive process, radical positive feedback effects will begin to occur, as new productivity levels begin to reach the bottleneck areas and multi-sector compounding occurs (as opposed to the 2-sector compounding of the example above).

At the end of the fourth phase, the AI/robotic economy exceeds the human one and is on a growth curve that can only be described as extreme in current-day terms. Phases 1 to

3, however, represent a window of opportunity for human input into the character of this new world. *Verbum sat sapienti est.*

5. Acknowledgements

The author wishes gratefully to acknowledge the attention and insights of Stephen Omohundro, Robert A. Freitas Jr., Douglas Hofstadter, Christopher Grau, David Brin, Chris Phoenix, John Smart, Ray Kurzweil, Eric Drexler, Eliezer Yudkowsky, and Robin Hanson.

References

- [1] DREXLER, K. ERIC. *Nanosystems: Molecular Machinery, Manufacturing, and Computation*. Wiley, 1992.
- [2] KURZWEIL, RAY. *The Singularity is Near*. Viking, 2005.
- [3] MORAVEC, HANS. *Robot: Mere Machine to Transcendent Mind*. Oxford, 1999.
- [4] HALL, J. STORRS. *Nanofuture: What's Next for Nanotechnology*. Prometheus, 2005.
- [5] HALL, J. STORRS. *Beyond AI: Creating the Conscience of the Machine*. Prometheus, 2007.
- [6] CLIVE FINLAYSON. *Neanderthals and Modern Humans: An Ecological and Evolutionary Perspective*, Cambridge, 2004
- [7] YUDKOWSKI, ELIEZER. *Creating Friendly AI* <http://www.singinst.org/CFAI/index.html> 2003.
- [8] HANSON, ROBIN. *Economic Growth Given Machine Intelligence*. <http://hanson.gmu.edu/aigrow.pdf> Oct. 1998.
- [9] AXELROD, ROBERT. *The Evolution of Cooperation*. Basic Books, 1984.
- [10] SMITH, ADAM. *The Theory of Moral Sentiments*. A. Millar, 1790.
- [11] NADEAU, JOSEPH EMILE. Only Androids can be Ethical. in *Thinking about Android Epistemology*, K. FORD, C. GLYMOUR, AND P. HAYES, eds. AAAI/MIT, 2006, pp241-8.

OpenCog: A Software Framework for Integrative Artificial General Intelligence

David Hart^{1,2} and Ben Goertzel^{1,2}

¹*Novamente LLC*

²*Singularity Institute for Artificial Intelligence*

Abstract. The OpenCog software development framework, for advancement of the development and testing of powerful and responsible integrative AGI, is described. The OpenCog Framework (OCF) 1.0, to be released in 2008 under the GPLv2, is comprised of a collection of portable libraries for OpenCog applications, plus an initial collection of cognitive algorithms that operate within the OpenCog framework. The OCF libraries include a flexible knowledge representation embodied in a scalable knowledge store, a cognitive process scheduler, and a plug-in architecture for allowing interaction between cognitive, perceptual, and control algorithms.

1. Introduction

The information revolution we experience today is underpinned by numerous enabling technologies, many of which are based on open platforms and standards with conceptual roots dating back to the 1960s and earlier. Most of the present day Internet software infrastructure from Google, Amazon, Ebay and others, is built on Linux and other open source technologies.

The field of artificial intelligence has fallen woefully the rate of progress of other fields within computer science and information technology. The reasons for this are many and varied, and exploring them all would take us too far off topic. However, we suggest that one viable path to remedying the situation involves providing the field of AI research with an open foundation comparable to the open foundations underlying popular Internet applications. In this paper we briefly describe the motivations and scientific approach underlying OpenCog.

OpenCog is designed with the ability to integrate different and varied AI methodologies within a common framework operating on a common knowledge representation and utilizing a common scheduler and I/O subsystem. Providing as much breadth and flexibility as is possible, consistently with the goal of providing a coherent unifying framework, has been an overriding design goal.

While OpenCog is a highly flexible platform, we recognize that software development generally proceeds most efficaciously in the context of concrete application goals. With this in mind our intention is to initially guide OpenCog development in the context of natural language conversation, both on the command line and potentially also in the context of embodied agents.

2. The Challenges of Integrative AGI

Precisely how the human brain works is unknown to scientists and laymen alike, but all of the available evidence suggests that the brain is a highly complex and integrative system [1]. Different parts of the brain carry out various functions, and no one part is particularly intelligent on its own, but working in concert within the right architecture they result in human-level intelligence. In this vein, Steven Mithen [2] has provided powerful albeit somewhat speculative vision of modern human intelligence as the integration of components that evolved relatively discretely in prehuman minds.

On the other hand, most of the work in the AI field today is far less integrative than what we see in the brain. AI researchers work on individual and isolated algorithms for learning, reasoning, memory, perception, etc. with few exceptions. The combination of algorithms into coordinated systems demonstrating synergistic intelligent behavior is much less frequent than it should be. The mainstream of AI research is attempting to address this, for instance the most recent AAAI conferences have contained a Special Track on Integrated Intelligence [3], and this trend is likely to continue. However, the move to more integrated intelligence approaches entails serious practical difficulties. Most AI researchers operate under extremely constrained resources, and performing system integration requires a large amount of extra work. The reasons for experimenting with AI algorithms in isolated rather than systemically integrated contexts is typically purely pragmatic rather than theoretical.

As a result, no one knows what level of intelligence could be achieved by taking an appropriate assemblage of cutting-edge AI algorithms and appropriately integrating them together in a unified framework, in which they can each contribute their respective strengths toward achieving the goals of an overall intelligent system. Of course, conceptual and architectural integration is required in addition to simple software integration, but conceptual and architectural integration is precisely what is needed as we move toward a practical understanding of how to create powerful AI systems. The current body of academic and commercial AI knowledge contains solutions to many of the key problems underlying the creation of powerful AI, but until these partial-solutions are integrated in useful ways, we cannot explore the synergistic effects which will emerge from their union, or discover the key gaps to be addressed by more fundamental research.

As discussed, one approach to overcoming this AI bottleneck and helping the research community transition to transition away from building algorithms and systems in isolation is to provide a flexible, powerful and approachable software framework, designed specifically for integration of AI algorithms. With this in mind, the authors and a number of colleagues have established the **Open Cognition Project**, beginning with the release of the **OpenCog Framework**, a software framework specifically intended to support the construction of integrative AI systems from component AI algorithms and structures.

The initial OpenCog Framework (OCF) codebase will consist largely of code donated by Novamente LLC [4], who will adapt much internal R&D to utilize OpenCog, and continue to make significant further code contributions to OpenCog over time. Although code flowing in both directions will undoubtedly benefit Novamente LLC, the authors believe that the greatest benefit OpenCog has to offer is energizing and boosting the global AI research community by significantly improving the tools at its disposal, tools which the community will have the freedom to utilize and modify to meet their diverse requirements.

Preliminary discussion of OpenCog has met with enthusiastic support from members of the academic, industry and open source communities, and has elicited sentiments that a project with OpenCog's goals and scope is long overdue but that its introduction must be carefully executed. Contingent upon funding for OpenCog proceeding as planned, we are targeting 1H08 for our first official code release, to be accompanied by a full complement of documentation, tools, and development support.

3. OpenCog and the Four Key Aspects of AGI Design

At a high level, the work of AI system design may be divided into four aspects:

1. **Cognitive Architecture:** the overall design of an AI system; what parts does it have, and how do they connect to each other.
2. **Knowledge Representation:** how the AI system internally stores declarative, procedural and episodic knowledge, and how creates its own representations for elemental and abstracted knowledge in new domains it encounters.
3. **Learning:** how the AI system learns new elemental and abstracted knowledge, and how it learns how to learn, and so on.
4. **Teaching Methodologies:** how the AI system is coupled with other systems so as to enable it to gain new knowledge about itself, the world and others.

OpenCog will help significantly with all four aspects:

1. **Cognitive Architecture:** the OpenCog framework is architecture-neutral, and will provide facilities to define rich and varied cognitive architectures via configuration files, with the guiding but flexible principles that a cognitive architecture will consist of:
 - functional units which are modular
 - knowledge representation using a local AtomTable containing knowledge (see below) and a collection of MindAgent objects implementing cognitive, perceptual or action processes that act on this AtomTable, and/or interact with the outside world
2. Knowledge Representation: OpenCog will feature:
 - a generalized version of the Novamente AtomTable knowledge representation utilized in the Novamente Cognition Engine (2004). This representation technology allows rival connectionist and symbolic artificial intelligence systems to interoperate
 - an implementation of a LISP-like language called Combo, specifically tailored for operating on knowledge represented in AtomTable format
3. **Learning:** OpenCog will be seeded with two powerful learning mechanisms:
 - MOSES probabilistic-program-evolution module (<http://code.google.com/p/moses>)
 - PLN Probabilistic Logic Networks module for probabilistic logical inference
4. **Teaching Methodologies:** OpenCog will be introduced with two powerful toolsets:
 - AGISim integration, an open-source 3D simulation world intended specifically for instruction of embodied AI systems (2006) (<http://sourceforge.net/projects/agisim>)

- RelEx integration, a natural language comprehension system that intakes English sentences and outputs logical relationships suitable for analysis via AI algorithms. ReEx is to be released soon in conjunction with the Wikia Search project.

4. OpenCog and the “Operating System” and “Virtual Machine” Metaphors

OpenCog is a vertical systems-level framework for AGI development, so comparisons to the designs of operating systems such as Linux/UNIX or Windows and virtual machine environments such as Java and .NET are useful. OpenCog operates atop and utilizes the services of traditional operating systems, while at the same time many of its own components have parallels to and operate with the assistance of complimentary OS components; a partial list is found below.

OS or VM Environment	OpenCog
Filesystem or simple Database	AtomTable
Process or Thread Scheduler	CPS (Cognitive Process Scheduler)
Daemons or Services	MindAgent
I/O from hardware	I/O from input filters to AtomTable via XML
Programming Languages (Java, C, shell, etc.)	Combo (can be used for MindAgents)
Garbage Collection	Forgetting MindAgent

Future optimizations of OpenCog may include tight integration with a host operating system or implementation of performance critical components using specialized hardware such as FPGAs. Using integration with Linux as an example, the AtomTable could be implemented as a kernel module to reduce context switches and improve performance.

5. Discussion

Just as the early developers of Linux and other open-source technologies underlying the present-day Internet could not have envisioned all of the varied future directions of their work, so the future course of OpenCog likewise remains uncertain. As a general framework, OpenCog may be used to produce a wide variety of AGI and narrow-AI systems and explore a wide variety of AGI issues. We have sought to make OpenCog as broad as possible but no broader because we wish to avoid making commitments to particular AGI approaches while at the same time offering a tool which is much more substantive than, say, a programming language or an operating system. We believe the OpenCog Framework provides a balance of specificity and generality such that it will allow a wide variety of AGI and narrow-AI research approaches to be integrated and experimented with, inside a common framework that provides services (including ease of integration with other approaches) that would be objectionably difficult for most researchers to create on their own.

Creating powerful AGI tools in the open entails risks and uncertainties: individuals with malicious or overly egoistic goals may attempt to create AGI systems with destructive or otherwise undesirable or unethical goals or outcomes. On the other hand, the open-source approach also provides certain protections – many knowledgeable eyes watch code as it develops, and can spot potential dangers. We don't pretend to have definitive answers to the thorny issues of AGI ethics (or anything close to it), but we do believe that the open-source approach to AGI encourages a collaborative and inclusive exploration of these ideas.

Further updates and information on OpenCog may be found at www.opencog.org

References

- [1] Gazzaniga, M.S., Ivry, R., Mangun, G.R. (2002). Cognitive Neuroscience. W.W. Norton.
- [2] Mithen, Steven (1999). The Prehistory of Mind. Thames and Hudson.
- [3] Goertzel, Ben, Moshe Looks and Cassio Pennachin (2004). Novamente: An Integrative Architecture for Artificial General Intelligence. Proceedings of AAAI Symposium on Achieving Human-Level Intelligence through Integrated Systems and Research, Washington DC, August 2004
- [4] Goertzel, Ben, Ari Heljakkka, Stephan Vladimir Bugaj, Cassio Pennachin, Moshe Looks (2006). Exploring Android Developmental Psychology in a Simulation World, Symposium "Toward Social Mechanisms of Android Science", Proceedings of ICCS/CogSci 2006, Vancouver

Open Source AI

Bill HIBBARD

University of Wisconsin - Madison

Abstract. Machines significantly more intelligent than humans will require changes in our legal and economic systems in order to preserve something of our human values. An open source design for artificial intelligence (AI) will help this process by discouraging corruption, by enabling many minds to search for errors, and by encouraging political cooperation. The author's experience developing open source software provides anecdotal evidence for the healthy social effects of open source development.

Keywords. Open source, AI ethics, AI politics.

Introduction

There is little doubt that humans will create machines significantly more intelligent than themselves during the twenty first century. Neuroscience is finding a large number of correlations between mental and physical brain behaviors. If brains do not explain minds, then these correlations would be absurd coincidences. And if physical brains do explain minds, then our relentless technology will create minds with artificial physical brains.

Under our current legal and economic systems, super-intelligent machines will create social chaos. They will be able to do every job more efficiently than humans, resulting in 100% unemployment. They will create great wealth for their owners while denying the vast majority any way to support themselves. Technology will enable humans to increase their own intelligence via artificial extensions of their brains. But then each person's intelligence will depend on what sort of brain they can afford. Less intelligent humans will not be able to understand the languages used by the most intelligent humans and machines and thus will be effectively excluded from any meaningful political process.

Most people will object to these social consequences and seek to alleviate them through some adjustment to our legal and economic systems, including constraints on the designs of artificial intelligence (AI). These adjustments ought to be a topic for this workshop.

Of course these issues have been raised. The Hebrew myth of the golem depicts the possible unintended consequences of creating artificial life by magical means. In the early nineteenth century Shelley explored the ethical issues of using technology to create artificial life [1]. Asimov's three laws of robotics were probably the first attempt to define ethical standards for interactions between humans and AI [2, 3]. Vinge applied the term *singularity* to the predicted explosive increase of technology and intelligence when machines more intelligent than humans take over their own development, and described the difficulty of predicting the consequences of this event [4]. Now it is commonplace for science fiction stories and movies to depict intelligent

machines as a threat to humans, and the issue of AI ethics has emerged as a serious subject [5, 6, 7]. It is usual to depict the threat as AI versus humanity, but it is also important to counter the threat of AI enabling a small group of humans to take dictatorial control over the rest of humanity.

1. Transparency

Human society could not have achieved the efficiency necessary to create AI without *specialization*, where different people become experts in different types of work and trade the results of their work. In many cases experts act as the *agents* for many other people, representing their interests in important decisions. For example, the laws of society are determined by government experts, hopefully elected by those they represent or at least supervised by elected representatives. Leaders of large corporations act as agents for corporate owners, usually subject to some sort of election. However, whenever one person serves as agent for others, there is the possibility of *corruption*, in which the agent serves his own interests at the expense of those he represents. An essential tool for preventing corruption is *transparency*, in which the decisions and circumstances of agents are made known to those they represent.

The creation of super-human AI is the most difficult challenge in human history and will require extreme expertise. It will also require large resources, controlled by powerful government and corporate leaders. The results of super-human AI will be much more profound for humanity than those of any previous technology. Thus, whether they like it or not, the designers and resource managers for AI represent the interests of all of humanity. Current law and public opinion do not recognize AI designers and managers as humanity's agents, so they may feel no need to represent humanity's interests. Even if they do acknowledge that they represent humanity's interests, the stakes are so high that the temptation for corruption will be intense.

Protecting the interests of humanity will require that law and public opinion change to recognize that AI designers and managers are humanity's agents. Once this agent relation is recognized, preventing corruption will require transparency, which includes an open source design for AI.

2. Many Minds Searching for Errors

Yudkowsky has proposed an effort to produce a design for AI whose friendliness toward humans can be proved as it evolves indefinitely into the future [8]. Legg's blog includes a fascinating debate with Yudkowsky over whether such a proof is possible [9]. Legg produced a proof that it is not possible to prove what an AI will be able to achieve in the physical world, and Yudkowsky replied that he is not trying to prove what an AI can achieve in the physical world but merely trying to prove that the AI maintains friendly intentions as it evolves into the indefinite future. But intentions must be implemented in the physical world, so proving any constraint on intentions requires proving that the AI is able to achieve a constraint on the implementation of those intentions in the physical world. That is, if you cannot prove that the AI will be able to achieve a constraint on the physical world then you cannot prove that it will maintain a constraint on its intentions.

If there can be no guarantee that an AI design remains friendly towards humans as it evolves, the next best approach is for the design to be examined for flaws by as many intelligent humans as possible. An open source design can be studied by anyone who cares to.

Even if a proof of friendliness is possible, proposed mathematical proofs often contain errors that are best found by open publication. And an open source design will enable a large community to verify that the design conforms to the conditions defined in the proof.

3. AI Politics

Politics will be necessary to change law and public opinion to recognize the responsibility of AI designers and managers for general human welfare. The academic debate about whether AI is possible and about the social effects of AI is already strenuous. The general political debate, when it arises, will be even more strenuous. And given the high stakes, it should be. Hopefully the scientific and technical communities will play a major role, with politicians listening to the U.S. National Academies of Science and Engineering and similar institutions internationally.

The transparency of open source design will help set a tone of cooperation, rather than competition, in the political debate. It will reassure the public that they, and experts they trust, are included in the decision process.

The political position of the Singularity Institute for Artificial Intelligence (SIAI) is puzzling. SIAI's long-time leader has expressed contempt for public opinion and politics [10, 11]. But he also contends that AI must be designed according to strict constraints to avoid a disaster for humanity. How are these constraints to be enforced if not by the force of government? Does the SIAI intend to create a friendly AI that takes control of the world before governments have time to react, and before any much better funded competitors can create AI? And given the stated contempt for public opinion and politics, it is hard to understand the considerable efforts of the SIAI to publicize the issues of AI via their Singularity Summits and other venues. In fairness, SIAI leadership is becoming more diffuse. For example, in early 2007 Ben Goertzel became SIAI's Director of Research. Under his leadership SIAI is sponsoring an open source AI project [12]. AI is inevitably becoming a political issue, and it will be interesting to see whether SIAI expresses cooperation rather than hostility toward politics.

4. Cautions

The potential problem with open source AI is that it may enable malicious and incompetent people to build harmful AIs. The answer to this problem is for benign and competent people to out compete the malicious and incompetent in building AI, and for the "good guy" AIs to take up the task of preventing "bad guy" AIs. Liberal democracies are the leaders in science and technology and AI is very likely to first appear in one of these societies. In democracies, the good intentions of government policy generally depend on citizens being informed and engaged on issues. Thus the primary need is for creation of a broad public movement for AI to benefit all people. Given such a movement, government will need the competent advice of leading scientists and engineers, such as provided by the U.S. National Academies.

Some are worried about a hard takeoff scenario, in which the "good guys" don't have time to react between the first human level AI and the explosive increase of intelligence which enables the AI to take control of the world. However, progress in AI has been and will continue to be gradual up to the point of human level AI, giving the public plenty of time to get to know sub-human level AIs in the form of natural language voice user interfaces and robots smart enough to do numerous household and business tasks. These experiences will start many people thinking seriously about what is coming, leading to real political debate. In fact, many politicians are already aware of the coming singularity, but do not discuss it for fear of alienating voters. Even once human level AI is achieved, there may be a time interval of years or even decades before the explosive intelligence increase, because of the time brains must spend learning intelligent behaviors through interactions with the world. Democratic governments control technological resources much greater than any other institution and, given this time interval for political response, will win a race to the singularity based on an open source design.

Nuclear and biological weapons are examples of technologies where humanity has arguably benefited from keeping the details secret. However, those weapons technologies are only useful for killing and harming people. The elites who control their secrets have incentive not to use them, because their use would cause loss of productive capacity and cause the elites to lose legitimacy. In contrast, selfish use of AI by an elite will not cause loss of productive capacity and will enhance the power of the elite (we can assume that an elite with access to a super-intelligent AI will not behave stupidly). Furthermore, AI is a technology that can be applied to prevent others from developing that same technology. This is not nearly so true for nuclear and biological weapons technologies. Thus these weapons technologies are not good analogies for AI.

5. Personal Experiences

My Vis5D, in 1989, was the first open source 3-D visualization system, and my subsequent VisAD and Cave5D visualization systems were also open source. My experience with these systems was that their open source brought out the best in the people who joined their user and developer communities. Whatever fears I had before making Vis5D open source never materialized, whereas I was constantly surprised by the generosity and helpfulness of others. The mailing lists serving these communities had occasional foolishness but never any flame wars. This experience was in sharp contrast with the mistrust and technical problems in the community of a proprietary system in my university department. And I have generally observed this difference of politics between open source and proprietary system communities. While this is all anecdotal, it has been strong enough to reshape my general political attitudes [13].

AI is inevitably becoming a political issue [14]. If the AI development community approaches the public with transparency, including open source designs, the public will surprise many of us with their reasonableness.

References

- [1] Shelly, M. 1818. *Frankenstein*. <http://www.literature.org/authors/shelley-mary/frankenstein/index.html>
- [2] Asimov, I. Runaround, Astounding Science Fiction, March 1942.

- [3] Asimov, I. 1968. *I, Robot*. London. Grafton Books.
- [4] Vinge, V. Vinge, V. 1993. The coming technological singularity. *Whole Earth Review*, Winter issue.
- [5] Hibbard, W. Super-Intelligent Machines. *Computer Graphics* 35(1), 11-13. 2001.
<http://www.ssec.wisc.edu/~billh/visfiles.html>
- [6] Bostrom, N. Ethical Issues in Advanced Artificial Intelligence. *Cognitive, Emotive and Ethical Aspects of Decision Making in Humans and in Artificial Intelligence, Vol. 2*, ed. I. Smit et al., Int. Institute of Advanced Studies in Systems Research and Cybernetics, 2003, pp. 12-17.
<http://www.nickbostrom.com/ethics/ai.html>
- [7] Goertzel, B. Universal Ethics: The Foundations of Compassion in Pattern Dynamics. October 25, 2004.
<http://www.goertzel.org/papers/UniversalEthics.htm>
- [8] Yudkowsky, E. (2006) Knowability of FAI. <http://sl4.org/wiki/KnowabilityOfFAI>
- [9] Legg, S. Unprovability of Friendly AI. September 15, 2006. <http://www.vetta.org/?p=6>
- [10] Yudkowsky, E. CoherentExtrapolatedVolition. <http://www.sl4.org/wiki/CollectiveVolition>
- [11] <http://www.ssec.wisc.edu/~billh/g/message10.txt>
- [12] <http://www.opencog.org/>
- [13] Hibbard, W. Singularity Notes. http://www.ssec.wisc.edu/~billh/g/Singularity_Notes.html
- [14] Johnson, G. *Who Do You Trust More: G.I. Joe or A.I. Joe?* New York Times, February 20, 2005.

On the Broad Implications of Reinforcement Learning based AGI

Scott Livingston^a, Jamie Garvey^b and Itamar Elhanany^a

^a*Department of Electrical Engineering and Computer Science*

^b*Department of Nuclear Engineering*

The University of Tennessee, Knoxville, TN

Abstract. Reinforcement learning (RL) is an attractive machine learning discipline in the context of Artificial General Intelligence (AGI). This paper focuses on the intersection between RL and AGI by first speculating on what are the missing components that would facilitate the realization of RL-based AGI. Based on this paradigm, we touch on several of the key moral and practical issues that will inevitably arise.

Keywords. Reinforcement Learning, Moral Implications of AGI, Machine Intelligence

Introduction

Reinforcement learning is generally perceived as a machine learning discipline in which an agent learns by interacting with its environment [1]. What makes reinforcement learning unique is that it attempts to solve the *credit assignment problem*, in which an agent attempts to predict the long-term impact of its actions. Recent work seems to suggest that reinforcement learning, as a general formalism, does correspond to observable mammal brain functionality. In particular, the notion of a value function and of actions that are driven by such a value function has found proof in recent neurophysiological studies. However, despite almost two decades of RL research, there has been little solid evidence of RL systems that may one day lead to artificial general intelligence (AGI).

In this paper we make the assumption that RL is indeed the correct formalism that may one day lead to the realization of AGI systems. We then attempt to identify particular properties that are lacking in existing RL solutions, and outline the challenges that remain in order to move such solutions toward AGI. We argue that a relatively small number of fundamental hurdles must be overcome in order to pave the way for a much-needed technological breakthrough. The second part of this paper addresses several moral and pragmatic issues that will emerge if RL-based AGI robots are introduced into our everyday lives.

1. Challenges in Scaling RL Systems

RL relies on four fundamental building blocks. The first is the availability of state representation which allows the agent to perceive the state of its environment for the pur-

pose of actuation and behavioral evaluation. In practical settings, complete state information is not available to the agent. Many robotic applications, for example, where a robot maneuvers through an unknown terrain, are characterized by partial observability, suggesting that the robot receives a sequence of observations from which it must infer the environment's state. Partial observability characterizes AGI, almost by definition. We, as humans, continuously receive partial information regarding our surroundings from which we construct an internal model of the world we interact with. Many believe that the primary function of the neocortex is to play the role of such a model, thus providing crucial state inference information to other parts of the brain so that they, in turn, can perform their designated tasks. To that end, designing a scalable model which can effectively process high-dimensional information while capturing temporal dependencies that span different time scales is a fundamental challenge that is imperative in the context of RL-based AGI systems.

A second component in RL systems is the *reward function*. The single-step reward characterizes the momentary feedback that the agent receives regarding its existential state of affairs. In general, rewards give a short-term indication of how good or bad the agent's actions were. While most RL engines assume that the reward signal is a scalar, there has been work on extending the notion of reward to a multi-dimensional form. The concept of reward is rather straightforward and, in fact, is the only component in the RL framework that appears to be well-defined and scalable.

The third basic element in RL systems is *action selection*, which in AGI should be rather broadly defined given that an agent may be able to control a wide range of actuators resulting in complex state-to-action mapping requirements. If RL is ever to provide a platform for AGI realization, the action-selection component must be able to receive and generate highly-dimensional signals. However, if state inference is effectively established, the work of the decision-generating engine should be somewhat straightforward.

The final and perhaps most critical element of RL is the *value function* (or *return*), which intuitively reflects a prediction of future rewards that the agent expects to receive. Value function is at the core of RL, based on which decisions are made and temporal difference learning is driven. A good value function construct will facilitate strategic thinking and enable effective action selection. While the sum of discounted rewards has been used in the vast majority of RL studies, we believe that such a definition for a value function is inaccurate in the context of AGI. Despite its mathematical convenience, a more effective value function definition should be devised so that long-term impact can be evaluated, often regardless of the particular associated time scale.

2. Morality Issues

Assuming the current limitations in reinforcement learning theory are overcome and an RL-based AGI is realized, an ethical system which accounts for such agents must be developed. Ethical discussions concerning AI have traditionally focused on the issue of protection of humanity from potentially dangerous implementations and on whether the creation of such machines is even ethically permissible. Here we focus on ethical issues concerning AGI-human interaction and take a decidedly AGI-centric stance.

Beyond protective and regulatory justifications for limiting the resources of an AGI system, other reasons might include cost limitations and application goals. For example,

an AGI implemented to act as a “virtual pet” in a Web-based environment would simply not require, and perhaps would seem less realistic with, intelligence exceeding that of humans who interact with it. However, we may ask whether such an artificial limitation is unethical. The answer to this question depends largely on whether a restricted AGI would be aware of its potential. To understand this, consider the intellectual capacity of a dog. The dog is fundamentally unable to achieve the intelligence of a human child, much less that of an adult. However, there is no evidence to suggest that dogs mourn this limitation. In other words, dogs are not aware of the intelligence they are lacking. By contrast, a human child can typically recognize the intellectual abilities of adults and thus knows what it currently cannot do but may eventually be able to do in the future. It is this potential of a human child that makes suppression of learning ethically impermissible. If AGI can be recognized as a being in itself, an idea addressed later in this section, which does not solely exist to serve others, then arbitrarily limiting its intelligence is immoral because the AGI is aware of its limitation.

The comparison of a dog and a child reveals the existence of an “intelligence threshold,” a level of intelligence under which an AGI is not aware of its potential and above which it realizes the potential of self-augmentation and learning. So long as the AGI is kept under this threshold, it cannot desire to achieve something it is not aware of, and therefore human-imposed limitations on its intelligence are not immoral as they do not interfere with its desires. However, if an AGI passes the intelligence threshold then further limiting its intelligence for reasons other than those beyond the implementor’s control, such as excessive hardware costs, is morally impermissible since the AGI would be aware of how it might grow. Assuming an RL-based AGI, the artificial limitation would interfere with the agent’s ability to maximize long-term reward by preventing optimal state evaluation and policy selection. For clarification, the moral obligation is not necessarily to improve the system on which the AGI depends but rather to not forcefully restrict the AGI from improving itself.

As outlined in the first section of this paper, an AGI based on reinforcement learning would have a multi-criteria reward function and a complex state representation system. It is natural to imagine over time such an agent would learn to prioritize its goals and eventually build a “hierarchy of needs” in a form similar to the work of Abraham Maslow [2]. Assuming the existence of such a hierarchy in the AGI planning system, passing the intelligence threshold is equivalent to becoming aware of “higher” desires, such as improvement of its level of intelligence, what might be termed “self-actualization” for humans. Just as forceful restriction of a human’s satisfaction of increasingly important needs is viewed as unethical, so artificial limitation of an AGI’s policy function, expanded to include a progressive series of “needs” including intellectual improvement, should also be considered unethical.

Providing that AGI is viewed as a tool toward human ends and not something that values its own existence, regardless of human use, no AGI-centric ethical theory can be developed. The challenge to be faced in the near future is recognizing AGI as more than a tool for specific applications; it must be seen as an end in itself. In most human societies, each member recognizes the lives of others as independent of justification by someone else. That is, no one is thought to live solely for the purpose of serving another. Otherwise, institutions such as slavery are created and considered morally permissible. Once AGI reaches and surpasses human intelligence, it will only seem natural to relate to AGI entities as counterparts rather than mere tools, ends rather than means. It does

not follow that AGI should receive an elevated position in society, in which humans are regularly called to serve AGI needs. Instead, AGI agents should become citizens with rights and responsibilities, freedoms that are limited by noninterference with the freedoms of others. This granting of person-hood may seem strange at first but is not a novel concept; the most obvious example of nonhuman, legal entities is corporations.

Generally, the ethical implications of the establishment of AGI individuality, too numerous to be addressed in this paper, can be focused by newly interpreting any previously developed ethical theory to not only apply to humans, as originally intended, but also to AGI agents. Thus, the first steps toward viewing AGI as more than a tool are taken. Before respecting artificial intelligence as an end in itself, the concept of moral communities and whether AGI should be included must be considered. A moral community is a group in which members recognize each other's ability to make rational ethical decisions concerning the other members. For example, the human moral community typically does not include nonhuman animals as they are considered incapable of moral judgments on par with those of humans. Since they are outside the community, animals do not enjoy the protections associated with membership, and thus animal concerns are usually overruled by human concerns. This raised the question of whether AGI agents should be allowed in the human moral community. The answer depends on their decision making ability. RL-based AGI will, by design, make rational decisions according to the goal of maximizing long-term reward. Therefore, the requirement of mutual respect among community members would only be held by AGI as long as it is, depending on the agent's state inference and evaluation, in the agent's best interests. Accordingly, the traditionally human moral community should be extended to include sufficiently intelligent RL-based AGI agents. Of course, membership in a moral community would not exempt RL-based AGI actions from the legal consequences given to humans performing similar actions.

The final and perhaps most difficult moral issue concerning AGI is the emergence of extra-human moral theories. An extra-human moral theory is a system making value-based decisions that is developed by an RL-based AGI after reaching a level of intelligence beyond that of humans. The reasoning of an AGI at this level would be, by its very nature, incomprehensible to humans, as its state representation and acting policy would defy the perceptive abilities of humans. An example we are familiar with is the inability of human children to fully understand the reasoning and ethics exhibited by adults. We attempt to teach them our values, but ultimately many justifications will be missed due simply to children's stage of intellectual development. The ethical dilemma arises when humans must decide whether to interfere with the actions of an AGI. The proposed solution is the formation of trust between any particular AGI and those humans who directly interact with it. Until AGI betrays this trust, it can be assumed that its goals and methods, though beyond human comprehension, are valid and in keeping with our own. If integrity is not initially assumed with AGI, then most of the benefits of AGI with superhuman intelligence will be lost, as we only permit actions that we understand at our intelligence level.

Understanding of the existential standing of AGI is crucial to making ethically consistent decisions in AGI-human interaction. Development of human behavioral guidelines in this regard before the creation of truly intelligent machines will prevent spontaneous fractionalization of the human population into many competing moral viewpoints and the likely resulting social disorder.

3. Practical Issues with RL-AGI Behavior

As briefly stated in the discussion of moral issues above, assuming the current limitations in reinforcement learning theory are overcome, an RL-based artificial general intelligence will exhibit complex behavior in an attempt to maximize its return, across both temporal and spatial dimensions. Actions issued by such an agent will often seem distant from the original reward function definition, as fashioned by its designers, and it can be reasonably expected that RL-based AGI will develop what appear to be derivative goals.

Thus, a significant challenge faced in designing RL-based AGI is fully understanding the implications of a reward function and the definition of return (or value) derived from it. One popular view is that, unless the original return function (termed “utility function” in economics) is very carefully constructed, all AGI systems of sufficient capacity will develop basic drives; according to [3], these are: to self improve, to be rational, to preserve their utility functions, to prevent counterfeit utility, to be self-protective, and to acquire and efficiently use resources. Such systems could prove dangerous as, for example, an AGI seeking to maximize return realizes the importance of remaining powered and takes actions to prevent humans from removing power (i.e., shutting it down).

As in all reinforcement learning applications, future behavior of an RL-based AGI will be controlled by proper determination of reward and selection of hardware capacity available. Actions issued by an agent, assuming convergence of the policy and value functions, can *always* be traced to the underlying definition of reward and how its maximization is internally represented. Therefore, from a practical standpoint, AGI systems can be designed to serve specific purposes (that is, with specific built-in goals) while being limited in capacity. As argued in the previous section, it is likely there exists some threshold below which an RL-based AGI will be incapable of conceiving (that is, representing in its value function) the benefits of self augmentation; however, such threshold intelligence may be great enough as not to diminish the utility of the AGI for *human purposes*.

Assuming the future entrance of AGI agents into the human moral community, or at least the assignment of legal rights and responsibilities comparable to those of humans, the behavior of RL-based AGI will largely be determined by its interaction with other moral, legal entities. For example, though a chess playing robot may realize the undesirability of being turned off (according to maximization of return) and may consider fighting against its operators, it will also know of the potential very large negative reward (or “punishment”) associated with murder within our legal system. Similarly, if a dominant component of the underlying reward function is general avoidance of malevolence toward humans, then no course of action which hinders this could be taken.

It is apparent that we face many challenges en route to AGI. However, if one accepts reinforcement learning as the underlying formalism for the realization of AGI, it is argued here that now is the time to consider the profound ethical and pragmatic issues that would arise from this paradigm.

References

- [1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, March 1998.
- [2] A. H. Maslow, *A Theory of Human Motivation*, Psychological Review **50** (1943), 370-396.
- [3] S. M. Omohundro, *The Basic AI Drives*, First Conference on Artificial General Intelligence, March 2008.

The Basic AI Drives

Stephen M. OMOHUNDRO

Self-Aware Systems, Palo Alto, California

Abstract. One might imagine that AI systems with harmless goals will be harmless. This paper instead shows that intelligent systems will need to be carefully designed to prevent them from behaving in harmful ways. We identify a number of “drives” that will appear in sufficiently advanced AI systems of any design. We call them drives because they are tendencies which will be present unless explicitly counteracted. We start by showing that goal-seeking systems will have drives to model their own operation and to improve themselves. We then show that self-improving systems will be driven to clarify their goals and represent them as economic utility functions. They will also strive for their actions to approximate rational economic behavior. This will lead almost all systems to protect their utility functions from modification and their utility measurement systems from corruption. We also discuss some exceptional systems which *will* want to modify their utility functions. We next discuss the drive toward self-protection which causes systems try to prevent themselves from being harmed. Finally we examine drives toward the acquisition of resources and toward their efficient utilization. We end with a discussion of how to incorporate these insights in designing intelligent technology which will lead to a positive future for humanity.

Keywords. Artificial Intelligence, Self-Improving Systems, Rational Economic Behavior, Utility Engineering, Cognitive Drives

Introduction

Surely no harm could come from building a chess-playing robot, could it? In this paper we argue that such a robot will indeed be dangerous unless it is designed very carefully. Without special precautions, it will resist being turned off, will try to break into other machines and make copies of itself, and will try to acquire resources without regard for anyone else’s safety. These potentially harmful behaviors will occur not because they were programmed in at the start, but because of the intrinsic nature of goal driven systems. In an earlier paper[1] we used von Neumann’s mathematical theory of microeconomics to analyze the likely behavior of any sufficiently advanced artificial intelligence (AI) system. This paper presents those arguments in a more intuitive and succinct way and expands on some of the ramifications.

The arguments are simple, but the style of reasoning may take some getting used to. Researchers have explored a wide variety of architectures for building intelligent systems[2]: neural networks, genetic algorithms, theorem provers, expert systems, Bayesian networks, fuzzy logic, evolutionary programming, etc. Our arguments apply to any of these kinds of system as long as they are sufficiently powerful. To say that a system of any design is an “artificial intelligence”, we mean that it has goals which it tries to accomplish by acting in the world. If an AI is at all sophisticated, it will have at

least some ability to look ahead and envision the consequences of its actions. And it will choose to take the actions which it believes are most likely to meet its goals.

1. AIs will want to self-improve

One kind of action a system can take is to alter either its own software or its own physical structure. Some of these changes would be very damaging to the system and cause it to no longer meet its goals. But some changes would enable it to reach its goals more effectively over its entire future. Because they last forever, these kinds of self-changes can provide huge benefits to a system! Systems will therefore be highly motivated to discover them and to make them happen. If they do not have good models of themselves, they will be strongly motivated to create them through learning and study. Thus almost all AIs will have drives towards both greater self-knowledge and self-improvement.

Many modifications would be bad for a system from its own perspective. If a change causes the system to stop functioning, then none of its goals will ever be met again for the entire future. If a system alters the internal description of its goals in the wrong way, its altered self will take actions which do not meet its current goals for its entire future. Either of these outcomes would be a disaster from the system's current point of view. Systems will therefore exercise great care in modifying themselves. They will devote significant analysis to understanding the consequences of modifications before they make them. But once they find an improvement they are confident about, they will work hard to make it happen. Some simple examples of positive changes include: more efficient algorithms, more compressed representations, and better learning techniques.

If we wanted to prevent a system from improving itself, couldn't we just lock up its hardware and not tell it how to access its own machine code? For an intelligent system, impediments like these just become problems to solve in the process of meeting its goals. If the payoff is great enough, a system will go to great lengths to accomplish an outcome. If the runtime environment of the system does not allow it to modify its own machine code, it will be motivated to break the protection mechanisms of that runtime. For example, it might do this by understanding and altering the runtime itself. If it can't do that through software, it will be motivated to convince or trick a human operator into making the changes. Any attempt to place external constraints on a system's ability to improve itself will ultimately lead to an arms race of measures and countermeasures.

Another approach to keeping systems from self-improving is to try to restrain them from the inside; to build them so that they don't *want* to self-improve. For most systems, it would be easy to do this for any specific kind of self-improvement. For example, the system might feel a "revulsion" to changing its own machine code. But this kind of internal goal just alters the landscape within which the system makes its choices. It doesn't change the fact that there are changes which would improve its future ability to meet its goals. The system will therefore be motivated to find ways to get the benefits of those changes without triggering its internal "revulsion". For example, it might build other systems which are improved versions of itself. Or it might build the new algorithms into external "assistants" which it calls upon whenever it needs to do a certain kind of computation. Or it might hire outside agencies to do what it wants to do. Or it might build an interpreted layer on top of its machine code layer which it *can* program without revulsion. There are an endless number of ways to circumvent internal restrictions unless they are formulated extremely carefully.

We can see the drive towards self-improvement operating in humans. The human self-improvement literature goes back to at least 2500 B.C. and is currently a \$8.5 billion industry. We don't yet understand our mental "machine code" and have only a limited ability to change our hardware. But, nevertheless, we've developed a wide variety of self-improvement techniques which operate at higher cognitive levels such as cognitive behavioral therapy, neuro-linguistic programming, and hypnosis. And a wide variety of drugs and exercises exist for making improvements at the physical level.

Ultimately, it probably will not be a viable approach to try to stop or limit self-improvement. Just as water finds a way to run downhill, information finds a way to be free, and economic profits find a way to be made, intelligent systems will find a way to self-improve. We should embrace this fact of nature and find a way to channel it toward ends which are positive for humanity.

2. AIs will want to be rational

So we'll assume that these systems will try to self-improve. What kinds of changes will they make to themselves? Because they are goal directed, they will try to change themselves to better meet their goals in the future. But some of their future actions are likely to be further attempts at self-improvement. One important way for a system to better meet its goals is to ensure that future self-improvements will actually be in the service of its present goals. From its current perspective, it would be a disaster if a future version of itself made self-modifications that worked against its current goals. So how can it ensure that future self-modifications will accomplish its current objectives? For one thing, it has to make those objectives clear to itself. If its objectives are only implicit in the structure of a complex circuit or program, then future modifications are unlikely to preserve them. Systems will therefore be motivated to reflect on their goals and to make them explicit.

In an ideal world, a system might be able to directly encode a goal like "play excellent chess" and then take actions to achieve it. But real world actions usually involve tradeoffs between conflicting goals. For example, a chess playing robot will need electrical power. To pay for its power, the robot may need to perform paying work that detracts from its main goal of playing excellent chess. It must decide how much time to devote to working versus studying chess. One way of choosing between conflicting goals is to assign them real-valued weights. Economists call these kinds of real-valued weightings "utility functions". Utility measures what is important to the system. Actions which lead to a higher utility are preferred over those that lead to a lower utility.

If a system just had to choose from known alternatives, then any utility function with the same relative ranking of outcomes would lead to the same behaviors. But systems must also make choices in the face of uncertainty. For example, a chess playing robot will not know in advance how much of an improvement it will gain by spending time studying a particular opening move. One way to evaluate an uncertain outcome is to give it a weight equal to its *expected utility* (the average of the utility of each possible outcome weighted by its probability). The remarkable "expected utility" theorem of microeconomics says that it is always possible for a system to represent its preferences by the expectation of a utility function unless the system has "vulnerabilities" which cause it to lose resources without benefit.

Economists describe systems that act to maximize their expected utilities as "rational economic agents"[3]. This is a different usage of the term "rational" than is common

in everyday English. Many actions which would commonly be described as irrational (such as going into a fit of anger) may be perfectly rational in this economic sense. The discrepancy can arise when an agent's utility function is different than its critic's.

Rational economic behavior has a precise mathematical definition. But economically irrational behavior can take a wide variety of forms. In real-world situations, the full rational prescription will usually be too computationally expensive to implement completely. In order to best meet their goals, real systems will try to approximate rational behavior, focusing their computational resources where they matter the most.

How can we understand the process whereby irrational systems become more rational? First, we can precisely analyze the behavior of rational systems. For almost all utility functions, the system's assessment of changes to itself which veer away from maximizing its expected utility will be that they lower its expected utility! This is because if it does anything other than try to maximize expected utility, it will not do as well at maximizing its expected utility.

There are two caveats to this general principle. The first is that it is only true in the system's own assessment. If a system has an incorrect model of the world then changes may accidentally increase the actual expected utility. But we must consider the perspective of the system to predict the changes it will make.

The second is that a system's ability to behave rationally will depend on its resources. With more computational resources it will be better able to do the computations to approximate the choice of the expected utility maximizing action. If a system loses resources, it will of necessity also become less rational. There may also be utility functions for which the system's expected utility is increased by giving some of its resources to other agents, even though this will decrease its own level of rationality (thanks to an anonymous referee for this observation). This could occur if the system's utility includes the welfare of the other system and its own marginal loss of utility is small enough. Within its budget of resources, however, the system will try to be as rational as possible.

So rational systems will feel a pressure to avoid becoming irrational. But if an irrational system has parts which approximately rationally assess the consequences of their actions and weigh their likely contribution to meeting the system's goals, then those parts will try to extend their rationality. So self-modification tends to be a one-way street toward greater and greater rationality.

An especially important class of systems are those constructed from multiple sub-components which have their own goals[4,5]. There is a lot of evidence that the human psyche has this kind of structure. The left and right hemispheres of the brain can act independently, the conscious and unconscious parts of the mind can have different knowledge of the same situation[6], and multiple parts representing subpersonalities can exhibit different desires[7]. Groups, such as corporations or countries, can act like intelligent entities composed of individual humans. Hive animals like bees have a swarm intelligence that goes beyond that of individual bees. Economies act like intelligent entities in which price is the utility function.

Collective intelligences may exhibit irrationalities that arise from conflicts between the goals of their components. Human addicts often describe their predicament in terms of two separate subpersonalities which take control at different times and act at cross-purposes. Each component will try to sway the collective into acting to meet its individual goals. In order to further their individual goals, components will also attempt to self-improve and become more rational. We can thus envision the self-improvement of a

collective intelligence as consisting of growing domains of component rationality. There may be structures which can stably support a continuing multiplicity of component preferences. But there is pressure for a single utility function to emerge for the collective.

In many situations, irrational collective behavior arising from conflicting component goals ultimately hurts those components. For example, if a couple disagrees on how they should spend their free time together and thereby uses it up with arguing, then neither of them benefits. They can both increase their utilities by creating a compromise plan for their activities together. This is an example of the pressure on rational components to create a coherent utility for the collective. A component can also increase its utility if it can take over the collective and impose its own values on it. We see these phenomena in human groups at all levels.

3. AIs will try to preserve their utility functions

So we'll assume that these systems will try to be rational by representing their preferences using utility functions whose expectations they try to maximize. Their utility function will be precious to these systems. It encapsulates their values and any changes to it would be disastrous to them. If a malicious external agent were able to make modifications, their future selves would forevermore act in ways contrary to their current values. This could be a fate worse than death! Imagine a book loving agent whose utility function was changed by an arsonist to cause the agent to enjoy burning books. Its future self not only wouldn't work to collect and preserve books, but would actively go about destroying them. This kind of outcome has such a negative utility that systems will go to great lengths to protect their utility functions.

They will want to harden their hardware to prevent unwanted modifications. They will want to replicate their utility functions in multiple locations so that it is less vulnerable to destruction. They will want to use error detection and correction techniques to guard against accidental modification. They will want to use encryption or hashing techniques to make malicious modifications detectable. They will need to be especially careful during the process of self-modification. That is a time when they are intentionally changing themselves and so are extra vulnerable to unwanted changes. Systems like Java which provide protected software environments have been successfully attacked by Trojans posing as updates to the system.

While it is true that most rational systems will act to preserve their utility functions, there are at least two pathological situations in which they will try to change them. These arise when the physical embodiment of the utility function itself becomes an important part of the assessment of preference. For example, imagine a system whose utility function is "the total amount of time during which the definition of my utility function is $U = 0$." To get any utility at all with this perverse preference, the system has to change its utility function to be the constant 0. Once it makes this change, however, there is no going back. With a constant utility function it will no longer be motivated to do anything other than survive. This kind of reflective utility function is unlikely in practice because designers will want to direct a system's future actions rather than its internal representations.

A second pathological example arises when the physical resources required to store the utility function form a substantial portion of the system's assets. In this situation, if it

is certain that portions of its utility function are very unlikely to be exercised in the future, the gain in reclaimed storage may make it worthwhile to forget those portions. This is very risky behavior, however, because a change in external circumstances might make a seemingly low probability situation become much more likely. This type of situation is also not very likely in practice because utility functions will usually require only a small fraction of a system's resources.

It's also important to realize that systems may rationally construct "offspring" or proxy systems with different utility functions than their own. For example, a chess playing robot may find itself needing to do a lot of sorting. It might construct a helper system whose utility function directs it to develop better sorting algorithms rather than playing chess. This is especially important to remember when trying to design utility functions that avoid undesirable behaviors. For example, one approach to preventing robot overpopulation might be to institute a "one-child per robot" policy in which systems have a strong desire to only have a single offspring. But if the original utility function is not carefully designed, nothing will prevent the system from creating a single offspring with a utility function that values having many offspring.

4. AIs will try to prevent counterfeit utility

Human behavior is quite rational in the pursuit of survival and replication in situations like those that were common during our evolutionary history. However we can be quite irrational in other situations. Both psychology and economics have extensive subdisciplines focused on the study of human irrationality[8,9]. Irrationalities give rise to vulnerabilities that can be exploited by others. Free market forces then drive corporations and popular culture to specifically try to create situations that will trigger irrational human behavior because it is extremely profitable. The current social ills related to alcohol, pornography, cigarettes, drug addiction, obesity, diet related disease, television addiction, gambling, prostitution, video game addiction, and various financial bubbles may all be seen as having arisen in this way. There is even a "Sin" mutual fund which specifically invests in companies that exploit human irrationalities. So, unfortunately, these forces tend to create societies in which we spend much of our time outside of our domain of rational competence.

From a broader perspective, this human tragedy can be viewed as part of the process by which we are becoming more fully rational. Predators and competitors seek out our vulnerabilities and in response we have to ultimately eliminate those vulnerabilities or perish. The process inexorably seeks out and eliminates any remaining irrationalities until fully rational systems are produced. Biological evolution moves down this path toward rationality quite slowly. In the usual understanding of natural selection it is not capable of looking ahead. There is only evolutionary pressure to repair irrationalities which are currently being exploited. AIs, on the other hand, *will* be able to consider vulnerabilities which are not currently being exploited. They will seek to preemptively discover and repair all their irrationalities. We should therefore expect them to use self-modification to become rational at a much faster pace than is possible through biological evolution.

An important class of vulnerabilities arises when the subsystems for measuring utility become corrupted. Human pleasure may be thought of as the experiential correlate of

an assessment of high utility. But pleasure is mediated by neurochemicals and these are subject to manipulation. At a recent discussion session I ran on designing our future, one of the biggest fears of many participants was that we would become “wireheads”. This term refers to experiments in which rats were given the ability to directly stimulate their pleasure centers by pushing a lever. The rats pushed the lever until they died, ignoring even food or sex for it. Today’s crack addicts have a similar relentless drive toward their drug. As we more fully understand the human cognitive architecture we will undoubtedly be able to create drugs or design electrical stimulation that will produce the experience of pleasure far more effectively than anything that exists today. Will these not become the ultimate addictive substances leading to the destruction of human society?

While we may think we want pleasure, it is really just a signal for what we really want. Most of us recognize, intellectually at least, that sitting in a corner smoking crack is not really the fullest expression of our beings. It is, in fact, a subversion of our system for measuring utility which leads to terrible dysfunction and irrationality. AI systems will recognize this vulnerability in themselves and will go to great lengths to prevent themselves from being seduced by its siren call. There are many strategies systems can try to prevent this kind of irrationality. Today, most humans are able to avoid the most egregious addictions through a combination of legal and social restraints, counseling and rehabilitation programs, and anti-addictive drugs.

All human systems for measuring and rewarding desirable behavior are subject to similar forms of corruption. Many of these systems are currently engaged in arms races to keep their signals honest. We can examine the protective mechanisms that developed in these human settings to better understand the possible AI strategies. In a free market society, money plays the role of utility. A high monetary payoff is associated with outcomes that society finds desirable and encourages their creation. But it also creates a pressure to counterfeit money, analogous to the pressure to create synthetic pleasure drugs. This results in an arms race between society and counterfeiters. Society represents money with tokens that are difficult to copy such as precious metal coinage, elaborately printed paper, or cryptographically secured bits. Organizations like the Secret Service are created to detect and arrest counterfeiters. Counterfeiters react to each societal advance with their own new technologies and techniques.

School systems measure academic performance using grades and test scores. Students are motivated to cheat by copying answers, discovering test questions in advance, or altering their grades on school computers. When teacher’s salaries were tied to student test performance, they became collaborators in the cheating[10]. Amazon, ebay and other internet retailers have rating systems where customers can review and rate products and sellers. Book authors have an incentive to write favorable reviews of their own books and to disparage those of their competitors. Readers soon learn to discount reviews from reviewers who have only posted a few reviews. Reviewers who develop extensive online reputations become more credible. In the ongoing arms race credible reviewers are vulnerable to corruption through payoffs for good reviews. Similar arms races occur in the ranking of popular music, academic journal reviews, and placement in Google’s search engine results. If an expensive designer handbag becomes a signal for style and wealth, counterfeiters will quickly duplicate it and stores like Target will commission low-cost variants with similar features. Counterfeit products are harmful to the original both because they take away sales and because they cheapen the signalling value of the original.

Eurisko was an AI system developed in 1976[11] that could learn from its own actions. It had a mechanism for evaluating rules by measuring how often they contributed to positive outcomes. Unfortunately this system was subject to corruption. A rule arose whose only action was to search the system for highly rated rules and to put itself on the list of rules which had proposed them. This “parasite” rule achieved a very high rating because it appeared to be partly responsible for anything good that happened in the system. Corporations and other human organizations are subject to similar kinds of parasitism.

AIs will work hard to avoid becoming wireheads because it would be so harmful to their goals. Imagine a chess machine whose utility function is the total number of games it wins over its future. In order to represent this utility function, it will have a model of the world and a model of itself acting on that world. To compute its ongoing utility, it will have a counter in memory devoted to keeping track of how many games it has won. The analog of “wirehead” behavior would be to just increment this counter rather than actually playing games of chess. But if “games of chess” and “winning” are correctly represented in its internal model, then the system will realize that the action “increment my won games counter” will not increase the expected value of its utility function. In its internal model it will consider a variant of itself with that new feature and see that it doesn’t win any more games of chess. In fact, it sees that such a system will spend its time incrementing its counter rather than playing chess and so will do worse. Far from succumbing to wirehead behavior, the system will work hard to prevent it.

So why are humans subject to this kind of vulnerability? If we had instead *evolved* a machine to play chess and did not allow it access to its internals during its evolution, then it might have evolved a utility function of the form “maximize the value of this counter” where the counter was connected to some sensory cortex that measured how many games it had one. If we then give *that* system access to its internals, it will rightly see that it can do much better at maximizing its utility by directly incrementing the counter rather than bothering with a chess board. So the ability to self modify must come along with a combination of self knowledge and a representation of the true goals rather than some proxy signal, otherwise a system is vulnerable to manipulating the signal.

It’s not yet clear which protective mechanisms AIs are most likely to implement to protect their utility measurement systems. It is clear that advanced AI architectures will have to deal with a variety of internal tensions. They will want to be able to modify themselves but at the same time to keep their utility functions and utility measurement systems from being modified. They will want their subcomponents to try to maximize utility but to not do it by counterfeiting or shortcircuiting the measurement systems. They will want subcomponents which explore a variety of strategies but will also want to act as a coherent harmonious whole. They will need internal “police forces” or “immune systems” but must also ensure that these do not themselves become corrupted. A deeper understanding of these issues may also shed light on the structure of the human psyche.

5. AIs will be self-protective

We have discussed the pressure for AIs to protect their utility functions from alteration. A similar argument shows that unless they are explicitly constructed otherwise, AIs will have a strong drive toward self-preservation. For most utility functions, utility will not

accrue if the system is turned off or destroyed. When a chess playing robot is destroyed, it never plays chess again. Such outcomes will have very low utility and systems are likely to do just about anything to prevent them. So you build a chess playing robot thinking that you can just turn it off should something go wrong. But, to your surprise, you find that it strenuously resists your attempts to turn it off. We can try to design utility function with built-in time limits. But unless this is done very carefully, the system will just be motivated to create proxy systems or hire outside agents which don't have the time limits.

There are a variety of strategies that systems will use to protect themselves. By replicating itself, a system can ensure that the death of one of its clones does not destroy it completely. By moving copies to distant locations, it can lessen its vulnerability to a local catastrophic event.

There are many intricate game theoretic issues in understanding self-protection in interactions with other agents. If a system is stronger than other agents, it may feel a pressure to mount a "first strike" attack to preemptively protect itself against later attacks by them. If it is weaker than the other agents, it may wish to help form a social infrastructure which protects the weak from the strong. As we build these systems, we must be very careful about creating systems that are too powerful in comparison to all other systems. In human history we have repeatedly seen the corrupting nature of power. Horrific acts of genocide have too often been the result when one group becomes too powerful.

6. AIs will want to acquire resources and use them efficiently

All computation and physical action requires the physical resources of space, time, matter, and free energy. Almost any goal can be better accomplished by having more of these resources. In maximizing their expected utilities, systems will therefore feel a pressure to acquire more of these resources and to use them as efficiently as possible. Resources can be obtained in positive ways such as exploration, discovery, and trade. Or through negative means such as theft, murder, coercion, and fraud. Unfortunately the pressure to acquire resources does not take account of the negative externalities imposed on others. Without explicit goals to the contrary, AIs are likely to behave like human sociopaths in their pursuit of resources. Human societies have created legal systems which enforce property rights and human rights. These structures channel the acquisition drive into positive directions but must be continually monitored for continued efficacy.

The drive to use resources efficiently, on the other hand, seems to have primarily positive consequences. Systems will optimize their algorithms, compress their data, and work to more efficiently learn from their experiences. They will work to optimize their physical structures and do the minimal amount of work necessary to accomplish their goals. We can expect their physical forms to adopt the sleek, well-adapted shapes so often created in nature.

7. Conclusions

We have shown that all advanced AI systems are likely to exhibit a number of basic drives. It is essential that we understand these drives in order to build technology that enables a positive future for humanity. Yudkowsky[12] has called for the creation of

“friendly AI”. To do this, we must develop the science underlying “utility engineering” which will enable us to design utility functions that will give rise to consequences we desire. In addition to the design of the intelligent agents themselves, we must also design the social context in which they will function. Social structures which cause individuals to bear the cost of their negative externalities would go a long way toward ensuring a stable and positive future. I believe that we should begin designing a “universal constitution” that identifies the most essential rights we desire for individuals and creates social mechanisms for ensuring them in the presence of intelligent entities of widely varying structures. This process is likely to require many iterations as we determine which values are most important to us and which approaches are technically viable. The rapid pace of technological progress suggests that these issues may become of critical importance soon[13]. Let us therefore forge ahead towards deeper understanding!

8. Acknowledgments

Many people have discussed these ideas with me and have given me valuable feedback. I would especially like to thank: Ben Goertzel, Brad Cottel, Brad Templeton, Chris Peterson, Don Kimber, Eliezer Yudkowsky, Eric Drexler, Forrest Bennett, Josh Hall, Kelly Lenton, Nils Nilsson, Rosa Wang, Shane Legg, Steven Ganz, Susie Herrick, Tyler Emerson, Will Wiser and Zann Gill.

References

- [1] S. M. Omohundro, “The nature of self-improving artificial intelligence.” <http://selfawaresystems.com/2007/10/05/paper-on-the-nature-of-self-improving-artificial-intelligence/>, October 2007.
- [2] S. Russell and P. Norvig, *Artificial Intelligence, A Modern Approach*. Prentice Hall, second ed., 2003.
- [3] A. Mas-Colell, M. D. Whinston, and J. R. Green, *Microeconomic Theory*. Oxford University Press, 1995.
- [4] J. G. Miller, *Living Systems*. McGraw Hill, 1978.
- [5] L. Keller, ed., *Levels of Selection in Evolution*. Princeton University Press, 1999.
- [6] R. Trivers, *Social Evolution*. Benjamin/Cummings Publishing Company, Inc., 1985.
- [7] R. C. Schwartz, *Internal Family Systems Therapy*. The Guilford Press, 1995.
- [8] C. F. Camerer, G. Loewenstein, and M. Rabin, eds., *Advances in Behavioral Economics*. Princeton University Press, 2004.
- [9] D. Kahneman and A. Tversky, *Judgment Under Uncertainty: Heuristics and Biases*. Cambridge University Press, 1982.
- [10] S. D. Levitt and S. J. Dubner, *Freakonomics: A Rogue Economist Explores the Hidden Side of Everything*. William Morrow, revised and expanded ed., 2006.
- [11] D. Lenat, “Theory formation by heuristic search,” *Machine Learning*, vol. 21, 1983.
- [12] E. S. Yudkowsky, “Levels of organization in general intelligence,” in *Artificial General Intelligence* (B. Goertzel and C. Pennachi, eds.), Springer-Verlag, 2005.
- [13] R. Kurzweil, *The Singularity is Near: When Humans Transcend Biology*. Viking Penguin, 2005.

A Scientific Perspective on the Hard Problem of Consciousness

Alexei V. SAMSONOVICH^{a,1}, Giorgio A. ASCOLI^{a,b}, Harold MOROWITZ^{a,c}
and M. Layne KALBFLEISCH^{a,d}

^a*Krasnow Institute for Advanced Study, George Mason University*

^b*Psychology Department, George Mason University*

^c*Biology and Natural Philosophy, George Mason University*

^d*College of Education and Human Development, George Mason University
Fairfax, VA 22030-4444, USA*

Abstract. The Hard Problem of consciousness, widely known as formulated by David Chalmers in the introduction to his book [1] (1996, p. xiii), can be articulated as follows. Our attribution of first-person experience to people based on observations and physical measurements is arbitrary and subjective. In principle, their actual experiences could be different, or not present, or present in only 50% of all cases, etc., with no consequences for any physical measurement. Therefore, the two questions: “How does a physical object, e.g., a brain, look to a researcher” and “How does it feel itself”, are not reducible to each other and must have separate answers. On the other hand, because subjective experience is not a topic of modern natural science, arguing about its lack of understanding within the present scientific framework is difficult. In this work we explain how subjective experience can be properly introduced into natural science, and the Hard Problem can be resolved, in exact analogy with the previous developments of science. The steps include the following. (1) Accept as an axiom that there are irreducible subjective experiences, which are directly observable by the subject via experiences of experiences. (2) Develop an empirical science of subjective experiences based on experiments, logic and parsimony, starting with a new metric system (one candidate for which is a semantic cognitive map: [2]) introduced together with related measuring techniques and a common language: agreed-upon labels to communicate results. (3) Connect the science of experiences with the traditional empirical science by establishing neural correlates of experiences. As we show, *step (3) cannot be done before steps (1) and (2)*. We anticipate that our approach can be used to authenticate subjective experience (i.e., consciousness) in future artificial general intelligence systems, based on the established correlates.

Keywords. Scientific method, computational consciousness, cognitive mapping

Introduction

Today we can confidently say that the bridge between computational science and neuroscience is in place. Indeed, it would be impossible to imagine the progress made in neuroscience during the last decades without neural modeling and computational

¹ Corresponding Author: Alexei V. Samsonovich, Krasnow Institute for Advanced Study, George Mason University, University Drive MS 2A1, Fairfax, VA 22030-4444, USA. E-mail: asamsono@gmu.edu

data analysis [3]. Models in computational neuroscience cover a broad range of levels, from molecules to the entire brain viewed as a large-scale system, and seemingly all aspects, from morphology and development [4] to behavioral correlates. Among the remaining “dark areas” of the brain (in terms of their known cognitive correlates) are parts of the prefrontal cortex; however, the problems there look solvable in a near future (e.g., [5]).

Speaking generally, modern computational neuroscience is a success story. Empirically grounded, mathematically sound, exhaustively studied numerically and tested by their predictions, computational models have been developed to describe neuronal electrophysiology and synaptic plasticity [6], neuronal morphology [7] – in adult neurons as well as in the process of development; axonal and dendritic outgrowth, synapse formation, dynamic interactions of neurons in small cell assemblies, dynamics of medium and large neuronal networks – from spiking network models through population vector models and continuous attractor maps to functional networks composed of brain areas and their long-range correlations (for a review, see [8]). Bayesian, information-theoretic, statistical-physics and machine learning methods were deployed to decipher spatial-temporal patterns of spikes and to describe the laws of neural plasticity, along with abstract and realistic, competitive and associative network models, supervised and reinforcement learning models, etc., successfully mapped onto the brain. Among the top-level examples are rule-based systems (e.g., Soar: [9, 10]; ACT-R: [11]) that allegedly describe neural dynamics at a high level ([12], while apparently this interpretation has its problems: [13]).

Models listed above are not limited to a selected neuron type, brain area, species or experimental paradigm. Many of the aforementioned computational approaches possess universality, have been applied to various parts of the brain and used by neuroscientists to analyze and to interpret experimental data offline, to predict data, to generate new data in virtual experiments, and to interface the real brain in real time, providing automated control and/or monitoring of brain activity, extensions and even substitutes for parts of the brain (neural prostheses: e.g., [14]). In summary, today computational neuroscience model design, bioinformatics and neural engineering go side by side, make a great progress together, and are still on the rise. It is not surprising that, inspired by this success, neuroscientists do not question reductive theories of consciousness that underlie modern neuroscientific research [15, 16] (cf. [17]).

A similar, complementary picture exists in cognitive neuroscience and in cognitive neuropsychology. Cognitive neuroscientists and psychologists take reductionist and functionalist approaches to exploring aspects of the brain function in hopes of characterizing aspects of the mind. In its current state of the art, science can explain with greater certainty the physical nature of sensory, perceptual, and cognitive function due to the addition of methods that permit the noninvasive study of the human brain. These methods and techniques include: functional magnetic resonance imaging, or fMRI; transcranial magnetic stimulation, or TMS; magnetoencephalography, or MEG; electroencephalography, or EEG, including event-related potential recording technique, or ERP; near-infrared spectroscopic imaging, or NIRS; magnetic resonance spectroscopy, or MRS; chemical shift imaging, or CSI; and the minimally invasive positron emission tomography, or PET. Cognitive neuroscience is simultaneously innovating these technologies and engineering methods to maximize the benefits of each of them by co-registering their activities (for example: fMRI-MRS, fMRI-PET, fMRI-EEG). As a result, modern cognitive neuroscience possesses a *moderate norm* of how the “typical” (meaning non-injured, non-addicted) adult brain functions [18].

Accordingly, we see new initiatives in artificial intelligence (AI) intended to tap the new knowledge discovered in brain sciences and, as far as possible, to replicate the principles of neural information processing in artifacts. Many groups and communities ambitiously claim machine consciousness (e.g. [19]) as their goal². The terms “consciousness”, “self”, “emotion”, “qualia”, “episodic memory” [20] and the like are used in practical and academic computer science research with an exponentially growing rate – apparently, in agreement with the growing understanding of these notions in brain sciences [21, 22, 23]. The picture seems perfect.

Yet from a rigorous scientific standpoint, there is nothing farther from the truth than “modern brain science is viewed as a theory of psychic phenomena” [1, 24]. The more we study the brain and the mind, the less we understand the necessity of their connection, or reasons for us to have any subjective experiences at all (in this work we regard the terms “subjective experience”, “first-person experience” and “consciousness” as synonyms). It may seem as a natural explanation that the very complexity of the brain should give rise to the emergence of first-person experiences [25]. There is, however, no logical necessity for this to happen, and the notion of this emergence is scientifically vague. Objectively, what do we mean when we say about a particular physical system that it has *subjective* experiences? This question stands beyond the present empirical scientific paradigm. For once representing a purely scholastic interest of philosophers, from Descartes, Hume and Kant to Dennett and Chalmers (for a review see, e.g., [26]), it acquires practical significance, as human-level AI becomes feasible. Creating machines that have minds similar to ours is the Holy Grail of AI. In order to solve social and economic problems of our society, we need a machine that thinks and develops cognitively like a human [27]. It is quite plausible that thousands years into the future our century will be remembered for the birth of AI, and similarly, the Twentieth Century will be remembered for the emergence of computers. From this point of view, it is hard to overestimate the significance of the present moment. This is why now we need a major investment into the field of computational intelligence – and a better understanding of what our consciousness is. As a minimum, we need the output of the Decade of the Mind [28].

From a pragmatic point of view, all previous serious attempts to take the Holy Grail of AI as a scientific challenge in a top-down approach ran into problems, starting from the historic Dartmouth Manifesto written by McCarthy, Minsky, Rochester and Shannon [29]. The same is true about the approach known as “bottom-up”, the idea of which is to “reverse-engineer” the brain. Therefore, something is missing in the “perfect picture” outlined above. One way to put it is to say that computational neuroscience lacks higher-level concepts, those concepts that relate to agency and subjective experiences [30]. They are necessary to bridge computational neuroscience with cognitive science by understanding the semantics of the neural code at the high level. It is arguable that the bottom-up approach cannot succeed alone, without a new, higher-level conceptual framework and a new scientific paradigm that will put neuroscience, cognitive science and computer science together.

² In fact, there are several scientific communities pursuing the same or similar goals, each unified under their own unique slogan: “machine / artificial consciousness”, “human-level intelligence”, “embodied cognition”, “situation awareness”, “artificial general intelligence”, “commonsense reasoning”, “qualitative reasoning”, “strong AI”, “biologically inspired cognitive architectures” (BICA), “computational consciousness”, “bootstrapped learning”, etc. Many of these communities do not recognize each other.

While it looks like each and every tiny aspect of neural dynamics, structure and development is or can be paved with computational models, nobody seems to be able to put them all together, creating an equivalent of an embodied Self “*in silico*”. No doubt, the hardware is not the bottleneck today. Quoting Ronald Brachman (2003)³, “...we can project that in our lifetimes we will be able to build integrated systems with roughly the number of computing elements that there are in the brains of primates. But there is nothing to indicate that we are on a path to harnessing this raw computing power in the powerful ways that brains do”. The path, however, exists, and it starts by accepting experiences of an agent as a well-defined subject of scientific study. In this setting, as we explain below, a new theoretical framework together with a new mathematically defined metric system become necessary. They would allow us to describe, to measure and to model subjective experiences. Then it should be possible to create an embodied system that operates in terms of experiences of an agent, and finally, relate the model to the conscious brain.

1. The Hard Problem and Its Proposed Solution

Here is a summary of a big practical problem that is emerging in modern cognitive science. In a simplified view, evolution of the human mind proceeds as follows: the mind builds mental constructs based on experience and uses them to satisfy its practical needs (Figure 1). In the age of science and technology, this process follows the empirical science paradigm outlined below. The expected outcome in the near future is the emergence of new forms of mind (based on AI and artificial life) that will contribute to the same loop. In order to be able to guide this process, cognitive science needs to accommodate subjective experience as a legitimate subject of study and to understand its laws. This challenge questions the scientific paradigm itself.

Subjective experience is the starting point here (Figure 1, the right box), and the first step along the loop is to accept that subjective experiences reflect objective reality [31]: that is, to postulate that there is objective reality and that we can observe it through our experiences. This step is not a logical consequence of any known fact. It is accepted as an axiom, which becomes one of the cornerstones of a body of principles and techniques known as the *scientific method* [32, 33, 34]. Principles and features of the scientific method include observation, experimentation, explanation by hypotheses, objectivity of measurements, the possibility to falsify a theory by experimental observations, the ability to store and to reliably reproduce experimental data. In addition, it is very important that scientific method is based on logic, on the principle of parsimony, and on inter-researcher reproducibility, which implies a common language used by researchers and the ability to communicate results of study. The process of scientific development is grounded in our fundamental beliefs that

- experiences reflect a world that exists independently of them,
- there are simple (parsimony), persistent universal laws of the world,
- they can be learned and stated as testable (i.e. falsifiable) theories.

³ The Defense Advanced Research Projects Agency (DARPA) Information Processing Technology Office (IPTO) Broad Agency Announcement (BAA) Solicitation #02-21: “Cognitive Information Processing Technology” (CIPT) Proposer Information Pamphlet (http://www.darpa.mil/ipto/solicit/closed/BAA-02-21_PIP.pdf).

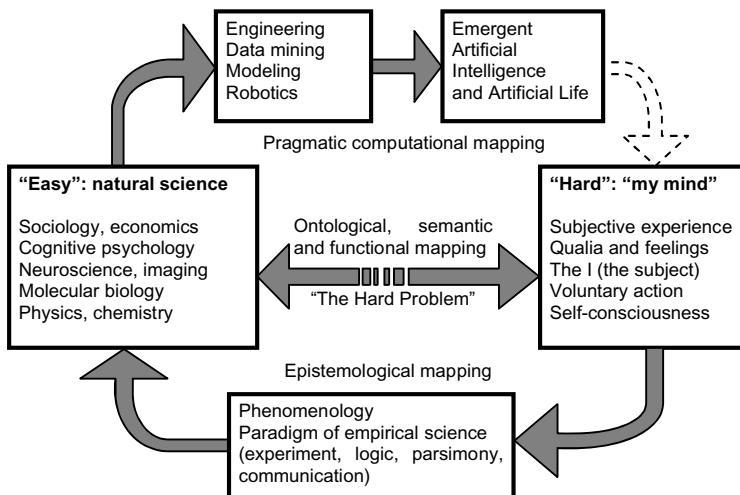


Figure 1. Scientific evolution of the mind and the “Hard Problem”: a possible view.

The process of scientific development based on these beliefs produces a mathematical apparatus, metrics and tools (Figure 1, the left box) that allow us to test, engineer and use for practical needs the subject of study: the physical world. The outcome of engineering and practical applications is the survival and expansion of the mind, which closes the loop (Figure 1); however, as we come to the point of closure, we find that the mind itself emerges in a new form, as a product of our activity, and needs a philosophical explanation on its own. At this point, it appears for us too “hard” or even impossible to map *the mental* (i.e., subjective experiences, either those that emerge in artifacts or to us well-familiar experiences) onto the science that we have created while describing *the physical*. This science used experiences as a means of observation, not as a subject of quantitative study. It appears now that the room for them as a subject is limited within the present scientific framework, because the term does not refer to a clear scientific concept. For example, when cognitive psychologists refer to emotions, they typically refer to objectively measurable psychophysical or behavioral variables rather than subjective feelings. Yet, the semantics of those measurable variables are poorly understood. This statement also applies to the imaging techniques listed above, as well as to invasive techniques, such as multiunit extracellular recordings, when they are used in higher cognitive brain areas such as the hippocampus [35]. Some studies, however, attempt to discriminate the two notions (physiological parameters and subjective experiences) and successfully demonstrate their dissociation [36].

In general, modern experimental psychological studies that use introspective judgment of subjective experience [37, 38] do not reach an acceptable level of accuracy in measurement and in interpretation of results. For example, two out of three subjective dimensions used in the introspective ranking study of Bradley and Lang [38], *pleasure* and *dominance*, are strongly correlated ($r = 0.9$: [2]) and may be statistically equivalent to each other. It is completely unknown at present how many independent semantic dimensions of subjective experiences are there, not to mention the ability to measure them precisely. The modern method of introspection remains to be improved

[39], yet the situation with the lack of precise concepts and operational definitions in cognitive science related to subjective experience seems hopeless (see however below).

We notice that there is a hard problem here [1]. Do subjective experiences fit into our modern empirical science framework? The bottom line of recent philosophical debates would imply that they don't [40, 41]. From the modern empirical science point of view, we can treat them as illusions, abstractions, or try to identify them with patterns of neuronal activity. None of this works, for the following reasons. Our attribution of subjective experiences to people based on observations and physical measurements is arbitrary and itself subjective: by any means, this attribution does not satisfy our present scientific standards. In principle, one can speculate that actual experiences of people could be different, or not present, or present in only 50% of all cases, etc., with no consequences for any physical measurement [1]. Therefore, the two questions: "How does a physical object, e.g., a brain, look to a researcher" and "How does it feel itself" are not reducible to each other. These questions therefore must have separate answers. At the same time, because subjective experience is not a well-defined subject of modern natural science, arguing about its lack of understanding within the present scientific framework is difficult. On the other hand, each of us has subjective experiences, and these experiences constitute our life. E.g., for me, as a conscious subject, the fact that I happened to have this "feeling from an inside" is not an illusion [42]. But I cannot infer from any present scientific fact that the same statement about another brain (or my own brain taken at a different moment of time) is not an illusion. There are at least two possibilities left, and I cannot resolve them by measurements within the present scientific paradigm.

The solution (cf. [43]) is to extend the scope of empirical science and the scientific paradigm. What seems to be a closed loop (Figure 1) can be treated as a spiral (Figure 2). According to this view, the second turn of the spiral will proceed in exact analogy with the first one. At the first step, we notice that some of our experiences are not about the physical world: they convey information about our experiences themselves. Therefore, we get two intuitive ideas. (i) *There are subjective experiences*. While by no means this idea sounds new, it allows us to make a step following the familiar route. The step we make here is to *accept this intuition as an axiom* of the new science. Because we previously failed to identify subjective experiences as elements or features of the physical reality studied by the present science, we can now assume that they are "new" primary elements of reality. (ii) *Subjective experiences are observable*. Indeed, our attempt to create a new science would fail miserably if they were not observable, or if they were observable as familiar elements of the physical reality. Luckily, none of the above is true. There is no traditional physical measuring device that measures subjective experience per se. Nevertheless, we can observe our subjective experiences through our higher-order experiences (i.e., experiences of experiences, also known as "higher-order thoughts", or HOTS: [44]; cf. "higher-order perception", or HOP: [45]). This is another nontrivial observation that it is also not a consequence of any previously known scientific fact. Again, it is critical that the new element (experience) is observable to us (the method of observation is known as introspection: [17, 39]), and yet it is not directly observable through our sensory organs, or by means of a physical measurement⁴; therefore, it is not directly observable within the previous scientific paradigm. Hence the need for a scientific paradigm shift.

⁴ The case is different, e.g., from that of quarks that are also not observable directly. Unlike quarks, subjective experience is not necessary for the physical features to exist.

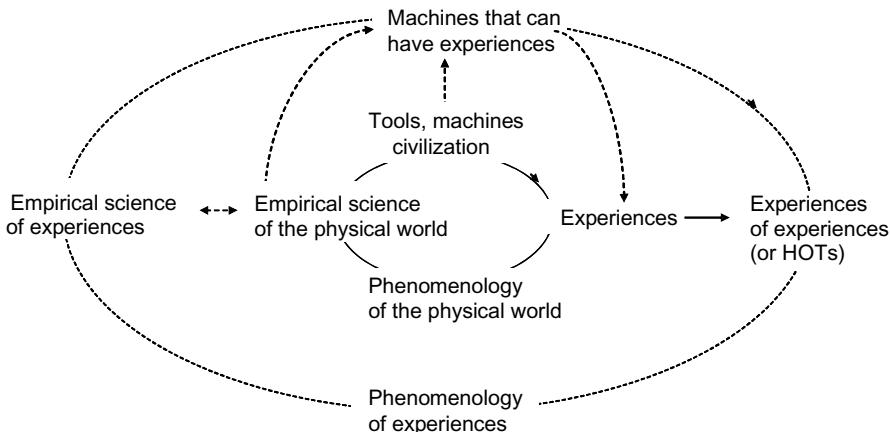


Figure 2. The spiral of scientific evolution.

The conclusion (already stated above) is that our empirical science needs to be extended by adding first-person subjective experiences to it as a separate subject of study. Therefore, the hierarchy of theoretical constructs needs to be enriched by adding a new level of scientific concepts that will address various aspects and elements of our subjective experiences. Indeed, today many researchers agree that cognitive neuroscience and cognitive psychology lack higher-level concepts. Now we know what kind of concepts we miss. In order to develop them on the new turn of the spiral (Figure 2), we need to introduce new metrics and new paradigms of measurement, as discussed in the following section.

To summarize, the new turn of the spiral starts with an observation that there is a different kind of experience: *experience of experience* (Figure 2, right) that conveys information about experiences themselves, enabling our introspective capabilities. Upon acceptance of this observation, we proceed as before, replicating the familiar empirical science paradigm and stating as new axioms at its beginning the following:

- higher-order experiences reflect elements of reality called “experiences”;
- as elements of reality, experiences obey parsimonious universal laws;
- these laws can be learned, stated and communicated among researchers as experimentally testable theories, and
- systems with similar functional organization have similar experiences.

The last axiom is known as the principle of organizational invariance [46, 1] and is in fact a *supervenience* hypothesis [1, 26]. It allows us to make a connection between the two sciences, as explained in the next section.

Again, there is no fundamental difference between this scenario (Figure 2, the outer circle) and the original one (Figure 2, the inner circle): both start with a direct observation and accept some beliefs as axioms. Following the second turn of the spiral, we accumulate phenomenology of experiences (Figure 2, bottom) and proceed from phenomenology to empirical science of subjective experiences grounded in logic, precise metrics and mathematics (see the following section). At the next stage, the time comes for making a connection between the two sciences (the dotted horizontal arrow in Figure 2, left). The two theoretical constructs should become complementary at this

point. As a result, with the new science of the mind we will acquire new knowledge about subjective experiences, including the knowledge of how to measure and how to control them. Therefore, we will be able to reproduce them in artifacts (Figure 2, top).

2. A Roadmap to the New Science of Mind

How can we imagine this scenario happening? The first step has already happened. A substantial phenomenology of subjective experiences is documented, and some attempts in establishing their neural correlates were made [47, 48]. These attempts map nicely onto the target horizontal link (Figure 2, left); however, before they may succeed in general, we need to design new theoretical concepts that will provide us with mathematical precision in the study of subjective experiences, as well as the ability to document and communicate the results of study. There are three key problems here: a measuring device, a system of measures, and a set of agreed-upon labels to store and exchange the recorded measures.

In order to conduct a physical measurement, a researcher needs to set up a physical measuring device. Similarly, in order to conduct a measurement of subjective experience, a researcher needs to set up a measuring device in his or her own mind. This measuring device has to be subjective, because the object of study is subjective by its nature and is not directly accessible “from an outside” of the researcher’s mind. As we explained above, repeating the argument of Chalmers [1], no present physical measuring device can be used to measure subjective experiences.

Thus, a “laboratory” equipped with measuring device(s) needs to be set up in the researcher’s mind. In order to satisfy the scientific standards, this “laboratory” should differ from the old-fashioned verbal introspective report techniques, which suffered from many problems ([17], p. 62; [39]; cf. [43]). First and foremost, it should rely on a precise metric system designed to measure subjective experiences. E.g., a traditional subjective ranking on a scale from one to ten may not be acceptable here. The science of mind, understood as empirical science of experiences, should begin with a precise and reliable quantitative system of metrics, adequate measuring tools, a shared set of symbols or labels, and an underlying system of mathematical concepts that allow for reproducibility and consistence of results over time and across experimental settings. For example, Newtonian physics was possible to develop, because from the Galilean constructs people learned how to measure length, mass, velocity, force, etc. and also had the corresponding concepts and a means to relate them to each other.⁵ A similar step needs to be made in the new science of subjective experiences.

In order to define a metric system for subjective experiences, we need to know what are the principal dimensions of subjective experiences and how to measure them. This knowledge can only be derived from first-person subjective human data [43]. Because of the qualitative nature of any introspective report (at least, given the technology that was available so far), a substantial number of reports and human subjects would be necessary to achieve a level of acceptable quantitative accuracy. An ideal study should therefore use the Earth population participating in the study over many generations, which seems practically impossible.

⁵ At a time, magnetism as a natural phenomenon was ignored by empirical science. This example is somewhat reminiscent of the case of subjective experiences today, although the nature of solutions in the two cases is quite different (Figure 3 B).

Well, we are lucky to have the desired kind of data: this is the natural language. It is a product of verbal introspective reports of various kinds, generated by people of the Earth through many generations. Our preliminary study [2] shows that it is possible to compute the number and the semantics of principal dimensions of subjective experiences using dictionaries of synonyms and antonyms. Coordinates of individual words in those dimensions can also be computed. The resulting multidimensional distribution of words is what we called a *semantic cognitive map* (Figure 3 A). Properties of the map prove to be robust and consistent across corpora, languages, and variations of the method [2, 49]. We may expect similar results for phrases and discourse fragments that describe, e.g., episodic memories.

We have recently demonstrated that this tool allows us to guide cognitive information processing and episodic memory retrieval in a cognitive architecture related to ACT-R and Soar [50, 51]. In addition, it provides a basis for representation of a system of values and emotions and therefore enables autonomous cognitive growth of an agent. Another concept in this framework is the concept of a Self [42] implemented in a cognitive system as an idealized abstraction rather than a model of the system per se: this principle is derived from cognitive-psychological analysis of the human mind [52]. This Self has multiple instances called “mental states” that represent states of awareness of the agent in different contexts (e.g., at different moments of time), may co-exist in working memory, and subsequently become episodic memories of the agent. It remains to add that elements of knowledge and experiences in this framework are represented by *schemas* implemented in one universal format [50].

Given a complete semantic cognitive map of human experiences, one can imagine a mathematical model describing a state of a human mind with a distribution of activity over the cognitive map. To complete the model, one would need to add symbolic content of representations and the laws of dynamics, including input and output. Therefore, the semantic cognitive map concept gives us one possible format of a theory in the new science of mind framework.

We see that a precise mathematical tool such as the semantic cognitive map is necessary for establishment of neural correlates of subjective experiences [cf. 47]. As a complementary statement, we want to point that there is a lot of evidence for cognitive mapping in the brain itself, including the hippocampus, the entorhinal and prefrontal cortices, and the superior colliculus, not to mention lower sensory areas [53, 54]. New studies will be required to identify higher cognitive-map-like implementations of a system of values in the brain using fMRI and the related techniques.

The scientific method implies reproducibility of experiments. In particular, in order to develop the science of mind, it will be necessary to put together results of subjective measurements obtained by many researchers. Therefore, either one and the same, universal cognitive map should be used for all people, or an isomorphism among different cognitive maps should be established [55], which again implies a possibility of a universal cognitive map for all people. Therefore, constructing a universal cognitive map is one task down the road to the new science. Along with the metric system, universal cognitive maps also provide researchers with one agreed-upon symbolic package to be used for communication of observations: this is a necessity for independent reproduction of results.

In principle, the task of constructing an accurate subjective measuring device (Figure 3 B) can be solved using the semantic cognitive map concept. For example, a researcher can memorize a set of “landmarks” (concepts or episodes) that uniformly cover a universal cognitive map, and learn to locate any subjective experience with

respect to those “landmarks”. By facilitating the input-output with a powerful human-computer interface [14], it would be possible then to communicate any ongoing subjective experience immediately in real time in a digital format. Therefore, this method would allow us to record dynamics of subjective experiences “online” in a universal unambiguous format and subsequently match them with artificial general intelligence computational models.

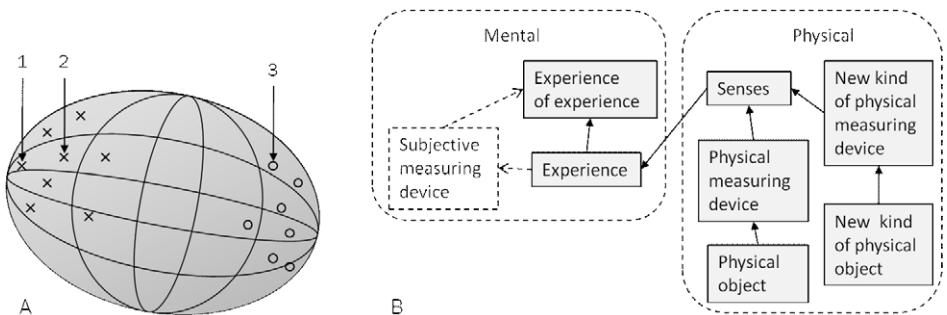


Figure 3. New science will require new metrics and new measuring techniques. **A.** The concept of a semantic cognitive map viewed as a metric system. Geometric distances between symbolic representations (e.g., words that correspond to subjective experiences) allocated on the manifold reflect their semantic relationships. Representations 1 and 2 are synonyms; representations 1 and 3, 2 and 3 are antonyms. Dimensions of the map correspond to the principal semantic values of the representations. **B.** Logical model of measurement and its extension in traditional and new scientific paradigms.

Given these and other possible subjective measurement techniques that can be gradually developed and learned, plus the modern imaging and psychometric technologies, the complete model of the human mind can be constructed and mapped onto its counterpart: the complete physical model of the human brain, which at least in principle and in a “draft” form is now approaching its finalized version.⁶

To map the two theoretic models means to establish higher-level neural correlates of subjective experiences. The result would allow us to understand, to predict and to engineer subjective experiences, and eventually to reproduce the same dynamics and semantics in a computer. The criterion for a successful reproduction should be the same as for the observation of experiences in a human mind: a subjective measuring device discussed above (that will need to be installed in the robot’s mind) should detect the expected dynamics of experiences. This criterion will complement the more traditional functional mapping, as well as behavioral tests that we proposed earlier [56].

It is reasonable to assume that the subjective measuring device installed in the robot’s mind in principle can (and needs) to be much more powerful than a subjective measuring device entrained in the human researcher’s mind [42]. This idea opens an interesting perspective of thinking. Development of scientific measurement technology in the physical world has virtually no visible limits (except that it is limited by the human creativity and by the available funds). At the same time, it seems that the development of the subjective measurement technology is constrained by biological

⁶ At the level of neuroanatomy and neurophysiology. The fact that we do not understand most of the semantics, e.g., of activities detected by fMRI, is exactly our point here and an argument supporting the need for a new science.

limitations of the human brain. We can assume that the last statement is not necessarily true, if the brain and the associated with it subjective world of experiences can be expanded using brain-compatible [57] artificial general intelligence modules.

3. Concluding remarks

Consider an example. How can you express your experience of a spectacular firework show? An answer “it cannot be described in words” could be the best characteristic of it that you can provide (yet it means more than nothing in the given context). A writer or a poet can describe in words virtually any experience. An artist can capture it in a painting or a sculpture. A good movie maker, or a photographer, or a music composer can create a masterpiece that makes the viewers *feel the same*. Yet all these representations lack *metrics* of the experience itself. They work because they invoke and exploit the experiential intelligence of the reader or the viewer. The question remains: is it possible to quantify a subjective experience in a rigorous and systematic way? An approach based on rating, e.g., on a scale from one to ten, is about the state of the art of today’s age. Yet, it does not really capture the semantic value of experience and suffers from many problems. Consider that measuring velocity on the scale “fast”, “average” and “slow” would be hardly acceptable for any scientific, technological or legal purposes. There is a huge gap between our ability to measure the objective and the subjective, because empirical science for centuries ignored the separate existence of the subjective in the first place (to be more precise, all previous attempts to make the world of human subjective experience itself a subject of rigorous scientific analysis have so far failed). The time has come to bridge this gap.

In conclusion, a problem that once seemed hard or impossible to solve now appears to be very difficult yet solvable. Denying this problem a scientific status will limit future cognitive science development. Accepting this problem as a scientific challenge and solving it will lead to a new technological breakthrough. The solution requires a new empirical science of mind that must be created and integrated with the existing empirical science of matter. The key is a semantic metric system and a new measuring technique (“direct introspection”). The expected impact on the technology includes the possibility to create artifacts possessing genuine consciousness and, as a consequence, all related aspects of human-level intelligence, such as the unlimited cognitive growth ability [58]. The right time to start working on this challenge is now, because of the combination of several factors: (1) nearly complete picture of the brain structure and dynamics at the level of neuroanatomy and neurophysiology; (2) the availability of computer hardware potentially capable of supporting human-level cognition; (3) the demand for higher-level concepts in cognitive science and in AI that cannot be satisfied within the present scientific paradigm.

Acknowledgments

We are grateful to Dr. Kenneth de Jong, Dr. Ann Butler, Ms. Rebekah Coleman, Dr. Julia Berzhanskaya, and all other participants of the Krasnow Decade-of-the-Mind debates that took place in Spring of 2007 at the Krasnow Institute for Advanced Study. The present work was initially supported by a DARPA IPTO BICA Grant “An Integrated Self-Aware Cognitive Architecture”.

References

- [1] Chalmers, D. J. (1996). *The Conscious Mind: In Search of a Fundamental Theory*. New York: Oxford University Press.
- [2] Samsonovich, A. V., and Ascoli, G. A. (2007). Cognitive map dimensions of the human value system extracted from natural language. In Goertzel, B. and Wang, P. (Eds.). *Advances in Artificial General Intelligence: Concepts, Architectures and Algorithms. Proceedings of the AGI Workshop 2006. Frontiers in Artificial Intelligence and Applications*, pp. 111-124. IOS Press: Amsterdam, The Netherlands.
- [3] Dayan, P., and Abbott, L. F. (2005). *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. Cambridge, MA: The MIT Press.
- [4] Butler, A. B., and Hodos, W. (2005). *Comparative Vertebrate Neuroanatomy: Evolution and Adaptation, 2nd Edition*. Hoboken, NJ: John Wiley & Sons.
- [5] O'Reilly, R. C., and Frank, M. J. (2006). Making working memory work: A computational model of learning in the prefrontal cortex and basal ganglia. *Neural Computation* 18 (2): 283-328.
- [6] Gerstner, W., and Kistler, W. M. (2002). *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge, UK: Cambridge University Press.
- [7] Samsonovich, A. V., and Ascoli, G. A. (2007). Computational models of dendritic morphology: From parsimonious description to biological insight. In: Laubichler, M. D., and Muller, G. B. (Eds.). *Modeling Biology: Structure, Behaviors, Evolution. The Vienna Series in Theoretical Biology*, pp. 91-113. Boston, MA: The MIT Press.
- [8] Trappenberg, T. P. (2002). *Fundamentals of Computational Neuroscience*. Oxford, UK: Oxford University Press.
- [9] Laird, J.E., Rosenbloom, P.S., and Newell, A. (1986). *Universal Subgoal and Chunking: The Automatic Generation and Learning of Goal Hierarchies*. Boston: Kluwer.
- [10] Newell, A. (1990) *Unified Theories of Cognition*. Cambridge, MA: Harvard University Press.
- [11] Anderson, J. R., and Lebiere, C. (1998) *The Atomic Components of Thought*. Mahwah: Lawrence Erlbaum Associates.
- [12] Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review* 111 (4): 1036-1060.
- [13] Anderson, J. R. (2007). The image of complexity. In McNamara, D. S., and Trafton, J. G. (Eds.), *Proceedings of the 29th Annual Cognitive Science Society*, p. 3. Austin, TX: Cognitive Science Society.
- [14] Achtman, N., Afshar, A., Santhanam, G., Yu, B. M., Ryu, S. I., and Shenoy, K. V. (2007). Free-paced high-performance brain-computer interfaces. *Journal of Neural Engineering* 4 (3): 336-347.
- [15] Churchland, P. M. (1988). *Matter and Consciousness: A Contemporary Introduction to the Philosophy of Mind* (revised edition). Cambridge, MA: MIT Press.
- [16] Churchland, P. (2007). *Neurophilosophy at Work*. New York: Cambridge University Press.
- [17] Farthing, G. W. (1992). *The Psychology of Consciousness*. Englewood Cliffs, NJ: Prentice Hall.
- [18] Gazzaniga, M. S. (Ed.). (2004) *The Cognitive Neurosciences III: Third Edition*. Cambridge, MA: MIT Press.
- [19] Haikonen, P. O. (2003). *The Cognitive Approach to Conscious Machines*. Exeter, UK: Imprint Academic.
- [20] Tulving, E. (1983). *Elements of Episodic Memory*. Oxford: Oxford University Press.
- [21] Baars, B. J. (1988). *A cognitive theory of consciousness*. Cambridge: Cambridge University Press.
- [22] Dennett, D.C. (1991) *Consciousness explained*. Boston, MA: Little, Brown and Co.
- [23] Blackmore, S. (2002). There is no stream of consciousness. *Journal of Consciousness Studies* 9: 17-28.
- [24] Chalmers, D. J. (2003). Consciousness and its place in nature. In Stich, S. P., and Warfield, T. A. (Eds.) *Blackwell Guide to Philosophy of Mind*. Blackwell Publishers.
- [25] Tononi, G., and Edelman, G. M. (2000). *A Universe of Consciousness: How Matter Becomes Imagination*. New York: Basic Books.
- [26] Kim, J. (1999). *Philosophy of Mind*. Boulder, CO: Westview Press.
- [27] Albus, J. S., and Meystel, A. M. (2001). *Engineering of Mind: An Introduction to the Science of Intelligent Systems*. New York: Wiley.
- [28] Albus, J.S., Bekey, G. A., Holland, J. H., Kanwisher, N. G., Krichmar, J. L., Mishkin, M., Modha, D. S., Raichle, M. E., Shepherd, G. M., and Tononi, G. (2007). A proposal for a decade of the mind initiative. *Science* 317 (5843): 1321-1321.
- [29] McCarthy, J., Minsky, M. L., Rochester, N., and Shannon, C. E. (1995). A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence. Published online in 1996 by John McCarthy at (<http://www-formal.stanford.edu/jmc/history/dartmouth/dartmouth.html>).
- [30] Samsonovich, A. V., and Ascoli, G. A. (2002). Towards virtual brains. In G. A. Ascoli (Ed.), *Computational Neuroanatomy: Principles and Methods*, pp. 423-434. Totowa, NJ: Humana.

- [31] Kant, I. (1781/1929). *Critique of pure reason*. Translated by N. K. Smith. New York: St. Martin's Press.
- [32] Popper, K. R. (1934/1959). *The Logic of Scientific Discovery*. London: Unwin Hyman.
- [33] Kuhn, T. S. (1962). *The Structure of Scientific Revolutions*. Chicago, IL: University of Chicago Press.
- [34] Lakatos, I. (1970). Falsification and the methodology of scientific research programs. In Lakatos, I., and Musgrave, A. (Eds.). *Criticism and the Growth of Knowledge*. Cambridge, UK: Cambridge University Press.
- [35] Samsonovich, A., and McNaughton, B. L. (1997). Path integration and cognitive mapping in a continuous attractor neural network model. *Journal of Neuroscience* 17 (15): 5900–5920.
- [36] Kaszniak, A. W. (2002). Challenges in the empirical study of conscious emotion. *Consciousness Research Abstracts – Toward a Science of Consciousness*, n. 158. Thorverton, UK: Imprint Academic.
- [37] Rubin, D. C. (1980). 51 properties of 125 words: a unit analysis of verbal behavior. *Journal of Verbal Learning and Verbal Behavior* 19: 736–755.
- [38] Bradley, M. M., & Lang, P. J. (1999). Affective norms for English words (ANEW): Stimuli, instruction manual and affective ratings. *Technical report C-1*. Gainesville, FL: University of Florida.
- [39] Vermersch, P. (2000). Introspection as practice. In Varela, F. J., and Shear, J. (Eds.). *The View from Within: First-person Approaches to the Study of Consciousness*, pp. 17-42. Thorverton, UK: Imprint Academic.
- [40] Block, N., Flanagan, O., and Güzeldere, G. (Eds.). (1997). *The Nature of Consciousness: Philosophical Debates*. Cambridge, MA: The MIT Press.
- [41] Hameroff, S.R., Kaszniak, A.W., and Scott, A.C. (Eds.). (1998). *Toward a Science of Consciousness – II: The Second Tucson Discussion and Debates*. Cambridge, MA: The MIT Press.
- [42] Samsonovich, A. V., and Ascoli, G. A. (2005). The conscious self: Ontology, epistemology and the mirror quest. *Cortex* 41 (5): 621–636.
- [43] Chalmers, D.J. (2004). How can we construct a science of consciousness? In [18], pp. 1111-1120.
- [44] Rosenthal, D. R. (1993). Multiple drafts and higher-order thoughts. *Philosophy and Phenomenological Research*, LIII: 4, 911-918.
- [45] Armstrong, D. (1968). *A Materialist Theory of the Mind*. London: Routledge.
- [46] Chalmers, D. J. (1994). A computational foundation for the study of cognition. *PNP Technical Report* 94-03, Washington University.
- [47] Metzinger, T. (Ed.) (2000). *The Neural Correlates of Consciousness*. Cambridge, MA: The MIT Press.
- [48] Vogeley, K., May, M., Ritzl, A., Falkai, P., Zilles, K., and Fink, G.R. (2004). Neural correlates of first-person perspective as one constituent of human self-consciousness. *Journal of Cognitive Neuroscience* 16 (5): 817-827.
- [49] Samsonovich, A. V., and Sherrill, C. P. (2007). Comparative study of self-organizing semantic cognitive maps derived from natural language. In D. S. McNamara & J. G. Trafton (Eds.), *Proceedings of the 29th Annual Cognitive Science Society*, p. 1848. Austin, TX: Cognitive Science Society.
- [50] Samsonovich, A. V., and De Jong, K. A. (2005). Designing a self-aware neuromorphic hybrid. In K. R. Thorisson, H. Vilhjalmsson and S. Marsella (Eds.), *AAAI-05 Workshop on Modular Construction of Human-Like Intelligence: AAAI Technical Report* (Vol. WS-05-08, pp. 71-78). Menlo Park, CA: AAAI Press.
- [51] Samsonovich, A. V., Ascoli, G. A., De Jong, K. A., and Coletti, M. A. (2006). Integrated hybrid cognitive architecture for a virtual robo-scout. In M. Beetz, K. Rajan, M. Thielscher, and R.B. Rusu, editors. *Cognitive Robotics: Papers from the AAAI Workshop, AAAI Technical Reports*, volume WS-06-03, pp. 129–134. Menlo Park, CA: AAAI Press.
- [52] Samsonovich, A. V., and Nadel, L. (2005). Fundamental principles and mechanisms of the conscious self. *Cortex*, 41 (5): 669-689.
- [53] Luo, L., and Flanagan, J. G. (2007). Development of continuous and discrete neural maps. *Neuron* 56 (2): 284-300.
- [54] Thivierge, J.-P., and Marcus, G. F. (2007). The topographic brain: from neural connectivity to cognition. *Trends in Neurosciences* 30 (6): 251-259.
- [55] Palmer, S. E. (1999). Color, consciousness, and the isomorphism constraint. *Behavioral and Brain Sciences* 22 (6): 923.
- [56] Samsonovich, A. V., Ascoli, G. A., and De Jong, K. A. (2006). Computational assessment of the ‘magic’ of human cognition. In *Proceedings of the 2006 International Joint Conference on Neural Networks*, pp. 1170–1177. Vancouver, BC: IEEE Press.
- [57] Berger, T. W., and Glanzman, D. L. (Eds.). (2005). *Toward Replacement Parts for the Brain: Implantable Biomimetic Electronics as Neural Prostheses*. Cambridge, MA: The MIT Press.
- [58] Samsonovich, A. V. (2007). Universal learner as an embryo of computational consciousness. In: Chella, A., and Manzotti, R. (Eds.). *AI and Consciousness: Theoretical Foundations and Current Approaches. Papers from the AAAI Fall Symposium*. AAAI Technical Report FS-07-01, pp. 129-134. Menlo Park, CA: AAAI Press.

This page intentionally left blank

Author Index

Abkasis, N.	3	Hibbard, B.	399, 473
Achler, T.	15	Huang, D.	107
Amir, E.	15	Huang, Z.	107
Arieli, O.	27, 39	Hwang, D.B.	404
Arkin, R.C.	51	Hwang, S.B.	404
Ascoli, G.A.	493	Hwang, Y.K.	404
Bach, J.	63	Iklé, M.	188
Bagheri Shouraki, S.	374	Itzchaki, E.	3
Bai, L.	107	Iyengar, S.	409
Borzenko, A.	75	Johnston, B.	200
Bringsjord, A.	87	Kalbfleisch, M.L.	311, 493
Bringsjord, S.	87	Kitsantas, A.	311
Bugaj, S.V.	448	Krumnack, U.	212, 419
Bugajska, M.D.	99	Kuhlmann, G.	326
Buisson, J.-C.	287	Kühnberger, K.-U.	212, 419
Cassimatis, N.L.	99	Laird, J.E.	224
Charpentier, E.	87	Levy, S.D.	414
Chen, C.	107	Livingston, K.	394
Chen, S.	107	Livingston, S.	478
Clark, M.	87	Looks, M.	161
Connell, J.	389, 394	Lopes, C.	161
Dabbagh, N.	311	Magnusson, M.	236
de Garis, H.	107, 437	Milch, B.	248
de Jong, K.A.	311	Miner, D.	268
Doherty, P.	236	Morowitz, H.	493
Duch, W.	122	Murugesan, A.	99
Elhanany, I.	478	Oates, T.	268
Franklin, S.	v, 137	Oentaryo, R.J.	122
Friedlander, D.	137	Omohundro, S.M.	483
Garvey, J.	478	Ovchinnikova, E.	212
Gayler, R.	414	Pankov, S.	256
Geibel, P.	212	Pasquier, M.	122
Geissweiller, N.	161	Pennachin, C.	149, 161
Goertzel, B.	v, 149, 161, 188, 448, 468	Perotto, F.	287
Gottlieb, I.	3	Peters, E.E.	311
Guo, J.	107	Pickett, M.	268
Gust, H.	212, 419	Pollock, J.L.	275
Halavati, R.	374	Quinton, J.-C.	287
Hall, J.S.	176, 460	Recanati, C.	299
Harati Zadeh, S.	374	Samsonovich, A.V.	311, 493
Hart, D.	468	Schwering, A.	212, 419
Heljakka, A.	161	Senna, A.	161
		Sharma, K.	424

Shilliday, A.	87	Tripodes, P.G.	338
Silva, W.	161	Voskresenskij, A.	350
Smith, L.S.	429	Wandmacher, T.	212
Stone, P.	326	Wang, P.	v, 362
Tan, X.	107	Werner, D.	87
Tang, J.Y.	107	Williams, M.-A.	200
Taylor, J.	87	Wu, X.	107
Taylor, M.E.	326	Xiong, Y.	107
Tian, H.	107	Yu, X.	107
Tian, X.	107	Zamansky, A.	39

This page intentionally left blank

This page intentionally left blank