Lecture Notes in Artificial Intelligence 6208

Edited by R. Goebel, J. Siekmann, and W. Wahlster

Subseries of Lecture Notes in Computer Science

Madalina Croitoru
Sébastien Ferré
Dickson Lukose (Eds.)

Conceptual Structures: From Information to Intelligence

18th International Conference
on Conceptual Structures, ICCS 2010
Kuching, Sarawak, Malaysia, July 26-30, 2010
Proceedings

Springer

Volume Editors

Madalina Croitoru
LIRMM
Université Montpellier II and CNRS
34392 Montpellier, France
E-mail: madalina.croitoru@lirmm.fr

Sébastien Ferré
IRISA
Université de Rennes 1
35042 Rennes, France
E-mail: sebastien.ferre@irisa.fr

Dickson Lukose
MIMOS BERHAD
Technology Park Malysia
57000 Kuala Lumpur, Malaysia
E-mail: Dickson.Lukose@mimos.my

Library of Congress Control Number: 2010929654

CR Subject Classification (1998): I.2, H.2, I.5, I.2.7, I.2.4, F.4.3

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 0302-9743

ISBN-10 3-642-14196-X Springer Berlin Heidelberg New York
ISBN-13 978-3-642-14196-6 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper 06/3180

Preface

The 18^{th} International Conference on Conceptual Structures (ICCS 2010) was the latest in a series of annual conferences that have been held in Europe, Australia, and North America since 1993. The focus of the conference has been the representation and analysis of conceptual knowledge for research and practical application. ICCS brings together researchers and practitioners in information and computer sciences as well as social science to explore novel ways that conceptual structures can be deployed.

Arising from the research on knowledge representation and reasoning with conceptual graphs, over the years ICCS has broadened its scope to include innovations from a wider range of theories and related practices, among them other forms of graph-based reasoning systems like RDF or existential graphs, formal concept analysis, Semantic Web technologies, ontologies, concept mapping and more. Accordingly, ICCS represents a family of approaches related to conceptual structures that build on the successes with techniques derived from artificial intelligence, knowledge representation and reasoning, applied mathematics and lattice theory, computational linguistics, conceptual modeling and design, diagrammatic reasoning and logic, intelligent systems and knowledge management.

The ICCS 2010 theme "From Information to Intelligence" hints at unveiling the reasoning capabilities of conceptual structures. Indeed, improvements in storage capacity and performance of computing infrastructure have also affected the nature of knowledge representation and reasoning (KRR) systems, shifting their focus toward representational power and execution performance. Therefore, KRR research is now faced with a challenge of developing knowledge representation and reasoning structures optimized for such reasonings.

There were 36 papers submitted to ICCS 2010 for peer review. All submissions were assessed by at least three referees, and most of them, by four referees. The top-ranked 12 submissions were accepted as full papers, amounting to an acceptance rate of 33%, and 6 other submissions were accepted as posters, amounting to a total acceptance rate of 50%. The programme also included five invited talks by leading researchers, two from the ICCS community itself and three from related communities. The thorough selection process would not have been possible without the help of the numerous reviewers to whom we express our acknowledgement.

We take this opportunity to thank the Ministry of Science, Technology and Innovation, Malaysia (MOSTI), MIMOS BERHAD, the National ICT Association of Malaysia (PICOM), Multimedia Development Corporation (MDeC), Center of Excellence in Semantic Technology and Augmented Reality, the Faculty of Cognitive Sciences and Human Development, as well as the Faculty of

Computer Science and Information Technology of University Malaysia Sarawak, for providing support in funding, promotion and local arrangements, making this event a success.

July 2010

Madalina Croitoru
Sébastien Ferré
Dickson Lukose

Conference Organization

Executive Committee

General Chair

Dickson Lukose	MIMOS BERHAD, Malaysia

Program Chairs

Madalina Croitoru	LIRMM, Université Montpellier II and CNRS, France
Sébastien Ferré	IRISA, Université de Rennes 1, France

Workshops Chair

Tan Yew Seng	MIMOS BERHAD, Malaysia

Tutorials Chair

Daniel Bahls	MIMOS BERHAD, Malaysia

Editorial Board

Galia Angelova	Bulgarian Academy of Sciences, Bulgaria
Frithjof Dau	SAP Dresden, Germany
Aldo de Moor	CommunitySense, The Netherlands
Harry Delugach	University of Alabama in Huntsville, USA
Peter Eklund	University of Wollongong, Australia
Bernhard Ganter	Technische Universität Dresden, Germany
Ollivier Haemmerlé	Université de Toulouse le Mirail, France
Pascal Hitzler	Universität Karlsruhe, Germany
Mary Keeler	VivoMind Intelligence, Inc., USA
Sergei Kuznetsov	Higher School of Economics, Moscow, Russia
Bernard Moulin	Université Laval, Canada
Marie-Laure Mugnier	LIRMM, France
Peter Øhrstrøm	Aalborg University, Denmark
Heather D. Pfeiffer	New Mexico State University, USA
Simon Polovina	Sheffield Hallam University, UK
Uta Priss	Edinburgh Napier University, UK
Sebastian Rudolph	University of Karlsruhe, Germany
Henrik Schärfe	Aalborg University, Denmark
John F. Sowa	VivoMind Intelligence Inc., USA

Gerd Stumme	University of Kassel, Germany
Rudolf Wille	Technische Universität Darmstadt, Germany
Karl Erich Wolff	University of Applied Sciences Darmstadt, Germany

Program Committee

Jean-François Baget	LIRMM-RCR and INRIA Rhône-Alpes, France
Radim Belohlavek	Palacky University of Olomouc, Czech Republic
Tru H. Cao	Ho Chi Minh City University of Technology, Vietnam
Dan Corbett	DARPA, Washington, DC, USA
Olivier Corby	INRIA Sophia-Antipolis, France
Juliette Dibie-Barthélemy	AgroParisTech, France
Pavlin Dobrev	ProSyst Labs EOOD, Bulgaria
Udo Hebisch	Technische Universität Freiberg, Germany
Joachim Hereth	DMC GmbH, Germany
Nathalie Hernandez	Université Toulouse le Mirail, France
Wolfgang Hesse	Philipps-Universität Marburg, Germany
Richard Hill	Sheffield Hallam University, UK
Jan Hladik	SAP Research Dresden, Germany
Adil Kabbaj	INSEA, Rabat, Morocco
Rob Kremer	University of Calgary, Canada
Markus Krötzsch	Universität Karlsruhe (TH), Germany
Leonard Kwuida	Zurich University of Applied Sciences, Switzerland
Michel Leclère	LIRMM, France
Robert Levinson	UC Santa Cruz, USA
Philippe Martin	Eurécom, France
Claudio Masolo	ISTC, Trento, Italy
Daniel Oberle	SAP Research Karlsruhe, Germany
Sergei Obiedkov	Higher School of Economics, Moscow, Russia
John Old	Edinburgh Napier University, UK
Anne-Marie Rassinoux	University Hospital of Geneva, Switzerland
Gary Richmond	City University of New York, USA
Olivier Ridoux	Université de Rennes 1, France
Eric Salvat	IMERIR, Perpignan, France
Ulrik Sandborg-Petersen	University of Aalborg, Denmark
Jeffrey Schiffel	The Boeing Company, USA
Denny Vrandecic	AIFB, Universität Karlsruhe, Germany
Guo-Qiang Zhang	Case Western Reserve University, Cleveland, USA

External Reviewers

Jason Heard
Melanie Kellar

Sponsoring Institutions

Ministry of Science, Technology and Innovation, Malaysia (MOSTI)
MIMOS BERHAD, Kuala Lumpur, Malaysia
The National ICT Association of Malaysia (PICOM)
Multimedia Development Corporation (MDeC)
Faculty of Cognitive Sciences and Human Development,
University Malaysia Sarawak, Malaysia
Faculty of Computer Science and Information Technology,
University Malaysia Sarawak, Malaysia
Center of Excellence in Semantic Technology and Augmented Reality,
University Malaysia Sarawak, Malaysia

Table of Contents

Invited Papers

Accepted Papers

Accepted Posters

Entities and Surrogates in Knowledge Representation

Michel Chein

LIRMM, University Montpellier2
161 Rue Ada, 34392 Montpellier, France
chein@lirmm.fr

Abstract. The question of the relationships between a word, or a text, or a symbol, and the object or concept, or idea to which it refers is a fundamental problem in many domains: philosophy, linguistics, psychology, etc. This reference problem has been tackled in many domains of Computer Science, especially in databases integration (e.g., entity resolution, record linkage, duplicates elimination, reference reconciliation) and computational linguistics (e.g., disambiguation, referring expressions). Most approaches are statistical, e.g., the object identification problem is viewed as a classification problem, but recent works, as ours, use AI techniques. We propose in this talk a simple logical framework for studying the relationships between a surrogate (a symbol in a computer system) and an entity to which it refers in an application domain. The two worlds linked by such a reference relation are irreconcilable ("La réalité est impossible" said Jacques Lacan), thus there is no hope to automatically solved reference problems. Nevertheless, if knowledge are used, it can be possible to help users faced with reference problems. In the proposed framework, which is motivated by actual problems in bibliographical databases, knowledge are described in terms of first order logic or in terms of conceptual graphs.

M. Croitoru, S. Ferré, and D. Lukose (Eds.): ICCS 2010, LNAI 6208, p. 1, 2010.

Exploring Conceptual Possibilities

Bernhard Ganter

Institut für Algebra
Technische Universität Dresden
Bernhard.Ganter@tu-dresden.de

Abstract. Shortly after the ideas of Formal Concept Analysis were first presented by Rudolf Wille in his seminal paper of 1982, one of the basic FCA methods came up: Attribute Exploration. It offers an interactive technique for exploring the possible attribute combinations in a given domain, supporting the search by a powerful and still somewhat peculiar algorithm.

Since then, remarkable progress has been made, so that the theoretical foundations of Formal Concept Analysis nowadays are broad and well-established. There are still remarkable research activities in the field, but many of the basic questions are solved and one may wonder what the future directions of research might be. What are worthwhile directions of further investigations on conceptual structures?

Suggestions for answers may be obtained from applications of the attribute exploration method, which, when applied to real world situations, often is confronted with problems that require more than the basic technique. Many extensions of the method, with additional features, have been discussed and investigated, mainly because there was demand from the side of applications. Quite in the beginning it was studied how "background knowledge" can be taken into account. An natural question also is how incomplete and imprecise data can be handled. The study of data with symmetries led to an extensions of the method to predicate logic (Horn formulae).

More recently, several attempts were made to handle structured data as well. What if the objects under consideration have an inner structure that is related to their attributes? For example, if the objects are molecules, or mathematical items. Or what, if the objects are related to other objects, as it is the case in processes or in causal networks? Then more expressive logics, like description logics, are needed, but that raises difficult questions. And perhaps even more challenging are situations where the data are unreliable, not merely because they are imprecise or incomplete, but because they are provided by many users not all of whom can be trusted.

We give an overview of our present knowledge of this theme and indicate some possible goals.

M. Croitoru, S. Ferré, and D. Lukose (Eds.): ICCS 2010, LNAI 6208, p. 2, 2010.

Graphical Representation of Ordinal Preferences: Languages and Applications

Jérôme Lang

LAMSADE, Université Paris-Dauphine, 75775 Paris Cedex 16, France
lang@lamsade.dauphine.fr

1 Introduction

The specification of a decision making problem includes the agent's preferences on the available alternatives. The choice of a *model* of preferences (*e.g.*, utility functions or binary relations) does not say how preferences should be *represented* (or *specified*). A naive idea would consist in writing them *explicitly*, simply by enumerating all possible alternatives together with their utility (in the case of cardinal preferences) or the list of all pairs of alternatives contained in the relation (in the case of ordinal preferences). This is feasible in practice only when the number of alternatives is small enough with respect to the available computational resources. This assumption is often unrealistic, in particular when the set of alternatives has a combinatorial (or multiattribute) structure, *i.e.*, when each alternative consists of a tuple of values, one for each of a given set of *variables* (or *attributes*): in this case, the set of alternatives is the Cartesian product of the value domains, and its cardinality grows exponentially with the number of variables.

For such combinatorial domains, we need a language allowing to express preferences as succinctly as possible. Such *compact preference representation languages* have been particularly studied in the Artificial Intelligence research community. A significant number of these languages are called "graphical', because they consist of a graphical component describing preferential dependencies between variables, together with a collection of local preferences on single variables or small subsets of variables, compatible with the dependence structure.

In this short paper we will focus on graphical languages for *ordinal preferences*, and especially on CP-nets and their extensions and variants[1]. After giving an brief presentation of this family of languages, we will show how they can be used for individual, collective or distributed decision making.

2 Graphical Languages for Ordinal Preference Representation

CP-Nets

Let $V = \{X_1, \ldots, X_n\}$ be a set of *variables*, also called *attributes*, with their associated finite *domains* $D_1, \ldots, D_n$. By $\mathcal{V} = \times_{X_i \in V} D_i$, we denote the set of all complete assignments, called *outcomes* (or *alternatives*). For $X \subseteq V$, we let $\mathcal{X} = \times_{X_i \in X} D_i$. For any

[1] However, at some places in the text we will refer to a graphical languages for *cardinal* preference representation such as *GAI*-nets [2], which specify local utility functions on small subsets of variables.

M. Croitoru, S. Ferré, and D. Lukose (Eds.): ICCS 2010, LNAI 6208, pp. 3–9, 2010.

disjoint subsets X and Y of V, the concatenation of assignments $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, denoted xy, is the $(X \cup Y)$-assignment which assigns to variables in X (resp. Y) the value assigned by x (resp. y).

A *(strict) preference relation* $\succ$ is an irreflexive and transitive (thus asymmetric) binary relation over $\mathcal{V}$. A strict preference relation $\succ$ is *linear* if it is connected, that is, for every $x \neq y$ we have either $x \succ y$ or $y \succ x$.

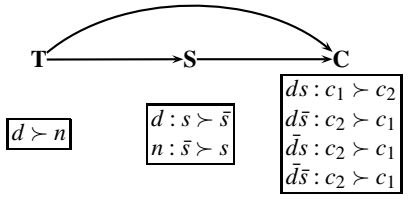
Let $\{X,Y,Z\}$ be a partition of the set of variables and $\succ$ a strict preference relation. X is *preferentially independent of Y given Z* (with respect to $\succ$) if and only if for all $x_1,x_2 \in \mathcal{X}$, $y_1,y_2 \in \mathcal{Y}$, $z \in D_Z$, we have $x_1y_1z \succ x_2y_1z$ if and only if $x_1y_2z \succ x_2y_2z$ [15].

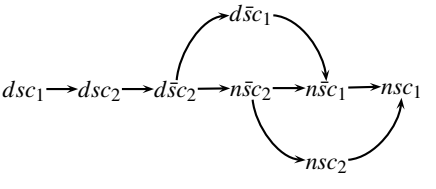
A CP-net [5] is composed of a directed graph representing the preferential dependencies between variables, and a set of conditional preference tables expressing, for each variable, the local preference on the values of its domain given the possible combination of values of its parents. Formally, a *CP-net* over $V = \{X_1,\ldots,X_n\}$ is a pair $\mathcal{N} = \langle G,P\rangle$ where G is a directed graph over $X_1,\ldots,X_n$ and P is a set of conditional preference tables $CPT(X_i)$ for each $X_i \in V$. For variable X_i, we denote by $Par(X_i)$ the set of parents of X_i in G and we let $NonPar(X_i) = \mathcal{V} \setminus (\{X_i\} \cup Par(X_i))$. Each conditional preference table associates a linear order on D_i with each instanciation u of $Par(X_i)$, denoted by $u :>$. The edges of $\mathcal{G}$ express preferential independencies: every variable is preferentially independent of its non-parents in G given its parents.

The semantics of a CP-net is defined as follows. A strict preference relation $\succ$ satisfies $\mathcal{N}$ if for all variables X_i, values $x_i,x_i' \in D_i$, assignments u of $Par(X_i)$, and assignments z of $NonPar(X_i)$, we have $ux_iz \succ ux_i'z$ if and only if $CPT(X_i)$ contains $u : x_i > x_i'$. A CP-net is *satisfiable* if there is some preference relation satisfying it. For any satisfiable CP-net $\mathcal{N}$, $\succ_{\mathcal{N}}$ is the smallest preference relation that satisfies $\mathcal{N}$.

Example 1. A user is looking for an airplane ticket. There are three variables: T (time of the flight), with possible values d (day) and n (night); S (stopover), with possible values s (yes) and $\bar{s}$ (no); and C (company), with two possible values c_1 and c_2. The agent has the following preferences: (a) she prefers a day flight to a night flight, unconditionally; (b) for a day flight she prefers to make a stopover; for a night flight she'd rather prefer not; (c) for a day flight with a stopover she prefers company c_1 because it implies spending a few hours in an airport she likes; in all other cases she prefers c_2.

The user's preferences are expressed by the CP-net $\mathcal{N}$ depicted below, together with the induced preference relation $\succ_{\mathcal{N}}$.





Many works on CP-nets make the additional assumption that the graph $\mathcal{G}$ is *acyclic*. Under this assumption, the CP-net is satisfiable, and the associated requests, consisting in comparing two outcomes or in searching for a non-dominated outcome, are computationally reasonable [6].

The preference relation $\succ_{\mathcal{N}}$ induced from a CP-net $\mathcal{N}$ is generally not complete. The complete preference relations extending $\succ_{\mathcal{N}}$ can be viewed as possible models of the user's preferences, and any preference assertion that holds in all such models can be viewed as a consequence of the CP-net [6]. Finally, for any complete preference relation $\succ$ there exists a satisfiable CP-net $\mathcal{N}$ such that $\succ$ extends $\succ_{\mathcal{N}}$ (note however that the dependence graph of $\mathcal{N}$ may contain cycles).

Extensions and Variants of CP-Nets

TCP-nets [9] enrich CP-nets by allowing the expression of relative importance statements between variables. CP-theories [20] are still more general; they allow conditional preference statements on the values of a variable, together with a set of variables that are allowed to vary when interpreting the preference statement. The language considered in [21] is even more general: there the preference statements do not compare single values of variables but tuples of values of different variables. *Conditional Importance Networks* [7] express monotonic preferences between sets of goods, *ceteris paribus*.

3 Individual Decision Making

Quoting from [6], the main purpose of any preference representation language is to support answering various queries about the preferences of the decision maker. Two fundamental types of queries are (a) *outcome comparison* (given two outcomes, determine if one of them dominates the other) and (b) *outcome optimization* (determine the best outcome(s)). Now, in many decision making problems, not all outcomes are feasible. A *constrained CP-net* consists of a CP-net $\mathcal{N}$ and a set of constraints Γ. Constraints can be expressed compactly using some representation language, typically CSPs (or, in the case of Boolean variables, propoisitional logic). Any outcome satisfying Γ is said to be *feasible*. The goal is to find an outcome α that is both feasible and undominated, *i.e.*, such that there is no feasible outcome β such that $\beta \succ_{\mathcal{N}} \alpha$.

Consider Example 1 again, and let us add the constraint that no day flight with a stopover is available: $T = d \Rightarrow S = \bar{s}$, and the constraint that company c_2 has only night flights: $C = c_2 \Rightarrow T = n$. The outcome dsc_1, which was the optimal outcome of $\succ_{\mathcal{N}}$, is now unfeasible. The new undominated outcomes are $d\bar{s}c_1$ and $n\bar{s}c_2$.

A different way of defining optimal solutions for constrained CSPs is suggested in [18]: α dominates β if there is a flipping sequence from α to β *that goes through feasible outcomes only*, and again we look for non-dominated feasible outcomes. Back to Example 1, suppose that we have the constraint $C = c_2 \Rightarrow T = n$. According to [18], dsc_1 and $n\bar{s}c_2$ are undominated, whereas only dsc_1 is undominated according to [6].

Constrained optimization is particularly relevant for configuration problems (see for instance [11] for an application to personalized configuration of web-page content). Another form of constrained optimization can come from the fact that an outcome is feasible only if there is a plan that realizes it; in [8], preferences among goals states are specified using a TCP-net, and one looks for a plan resulting in an optimal outcome, that is, an state α such that no other reachable state dominates α. A last example of using CP-nets for individual decision making is [4], which describes an information retrieval approach where CP-nets are used for expressing preferences over documents.

4 Group Decision Making

An important problem in computational social choice is voting on a combinatorial set of alternatives: a number of voters have to make a common decision on several possibly related issues. For instance, the inhabitants of some local community may have to make a joint decision over several related issues of local interest, such as deciding whether some new public facility such as a swimming pool or a tennis court should be built. Some of the voters may have preferential dependencies, for instance, they may prefer the tennis court to be built only if the swimming pool is not. As soon as voters have preferential dependencies between issues, it is generally a bad idea to decompose the problem into a set of smaller problems, each one bearing on a single issue: doing so may lead to *multiple election paradoxes* [10].

The input of a voting problem is typically a collection of linear preference relations. Therefore, graphical language for ordinal preferences, which express preferences locally and exploit preferential independencies, are particularly well-suited to the design of methods for collective decision making on combinatorial domains. There are two different ways of making use of graphical languages in such contexts.

The first way consists in *eliciting preferences globally, and then aggregating them*. For instance, if we are using CP-nets, we first elicit a CP-net for each voter, these CP-nets are then aggregated so as to output an outcome or a set of outcomes. See [19] and [22] for two approaches for CP-net aggregation (see also [12] for group decision making based on the aggregation of GAI-nets).

The second way consists in proceeding *sequentially*: at every step, we elicit the voters' preferences about a single variable, we use a local rule to compute the value chosen for this variable, and this value is communicated to all voters. For this we need to make an important domain restriction, namely, that there exists an order on variables, say $X_1 > \ldots > X_p$, such that for every voter and for every i, X_i is preferentially independent of $X_{i+1}, \ldots, X_p$ given $X_1, \ldots, X_{i-1}$ [17].

Consider the following example. We have a variable F(ood), with three possible values m(eat), f(ish) and v(egetarian) and a variable W(ine), with two possible values w(hite) and r(ed). Assume we have seven voters, all with unconditional preferences on F (and possible preferences on W depending on the value of F) whose conditional preferences are as follows:

1 voter: $m \succ f \succ v$; $m : r \succ w$; $f, v : w \succ r$
2 voters: $m \succ v \succ f$; $r \succ w$
2 voters: $f \succ v \succ m$; $m : r \succ w$; $f, v : w \succ r$
2 voters: $v \succ f \succ m$; $w \succ r$

If we apply the second method, assume that for F we use the Borda rule, which assigns a score or +2 (resp. +1, 0) to a candidate every time it is ranked first (resp. second, last) by a voter, and chooses the candidate that maximizes the sum of scores obtained for the different voters. Then preferences about food are elicited, the chosen value is v, then the voters' preferences about wine given that $F = v$ are elicited, and finally, after using the majority rule, we get that the collectively chosen value of W is w. If we apply the first method, we first aggregate the CP-nets, which, using the same local rules, results in the collective CP-net $v \succ f \succ m$; $m : r \succ w$; $f, v : w \succ r$, whose unique

undominated outcome is $F = v, W = w$, which again is the resulting collective decision (note that there are cases where the results given by both methods would have been different). Whereas the second method is cheaper in communication and computation, its range of applicability is much more restricted. Indeed, if the first voter's conditional preferences were $r : m \succ f \succ v$; $w : f \succ v \succ m$, $m : r \succ w$; $f, v : w \succ r$, then asking her to express her preferences on F would make her feel ill at ease, since her preferences on F depend on the value of W.

5 Game Theory

Game theory attempts to analyze formally strategic interactions between agents. Roughly speaking, a static game consists of a set of agents, and for each agent, a set of possible strategies and a utility function mapping every possible combination of strategies to a real value. Utility functions are usually represented explicitly, by listing the values for each combination of strategies. However, the number of utility values which must be specified, that is, the number of possible combinations of strategies, is exponential in the number of players, which renders such an explicit way of representing the preferences of the players unreasonable when the number of players is more than a few units. This becomes even more problematic when the set of strategies available to an agent consists in assigning a value from a finite domain to each of a given set of variables. In these cases, compact representation are once again useful.

Now, several key notions in game theory, such as pure strategy Nash equilibria or dominated strategies, need only ordinal preferences. Graphical languages for ordinal preference representation have been used in some places for representing and analyzing games, such as Section 5 of [3]: there, a game is described as a set of agents, a set of (binary) variables, a control assignment function assigning each variable to an agent, and finally, a compact description of the agents' preferences under the form of a collection of CP-nets. The structure of the agent's preferences can sometimes guarantee the existence or the unicity of a pure Nash equilibrium. For instance, assume we have two agents $\{1,2\}$ and two variables A and B, with domains $\{a, \bar{a}\}$ and $\{b, \bar{b}\}$, such that 1 controls A and 2 controls B. Preferential dependencies are as follows:

	situation 1	situation 2	situation 3
agent 1	$A \longrightarrow B$	$A \longrightarrow B$	$A \longleftarrow B$
agent 2	$A \longleftarrow B$	$A \longrightarrow B$	$A \longrightarrow B$

In situation 1, whatever the local preferences of 1 and 2, there is a unique equilibrium consisting of 1's preferred value of A and 2's preferred value of B. In situation 2, again there is a unique equilibrium, consisting of 1's preferred value of A and 2's preferred value of B given 1's preferred value of A. In situation 3, neither the existence nor the unicity of a pure Nash equilibrium is guaranteed. For instance, if the local preferences of 1 are $\bar{b} \succ b$, $b : a \succ \bar{a}$, $\bar{b} : \bar{a} \succ a$, and the local preferences of 2 are $a \succ \bar{a}$, $a : b \succ \bar{b}$, $\bar{a} : \bar{b} \succ b$, then the game has two pure Nash equilibria, namely ab and $\bar{a}\bar{b}$ [3].

A different connection between CP-nets and games appears in [1], where CP-nets are viewed as games in normal form and vice versa: each player corresponds to a variable of the CP-net, whose domain is the set of actions available to the player.

Normal form games with *cardinal* preferences have received even more attention: graphical games [16,14,13] specify, for each agent, the set of all players that have an influence on her; the utility of player i is compactly represented by a utility table that specifies a value for each combination of actions of the players on which i depends.

References

1. Apt, K.R., Rossi, F., Venable, K.B.: CP-nets and Nash equilibria. In: Proceedings of Third International Conference on Computational Intelligence, Robotics and Autonomous Systems (CIRAS 2005). Elsevier, Amsterdam (2005)
2. Bacchus, F., Grove, A.: Graphical models for preference and utility. In: Proceedings of UAI 1995, pp. 3–10 (1995)
3. Bonzon, E., Lagasquie-Schiex, M.-C., Lang, J., Zanuttini, B.: Compact preference representation and Boolean games. Autonomous Agents and Multi-Agent Systems 1(18), 1–35 (2009)
4. Boubekeur, F., Boughanem, M., Tamine-Lechani, L.: Towards flexible information retrieval based on CP-nets. In: Larsen, H.L., Pasi, G., Ortiz-Arroyo, D., Andreasen, T., Christiansen, H. (eds.) FQAS 2006. LNCS (LNAI), vol. 4027, pp. 222–231. Springer, Heidelberg (2006)
5. Boutilier, C., Brafman, R., Domshlak, C., Hoos, H., Poole, D.: CP-nets: a tool for representing and reasoning with conditional ceteris paribus statements. JAIR 21, 135–191 (2004)
6. Boutilier, C., Brafman, R., Domshlak, C., Hoos, H., Poole, D.: Preference-based constrained optimization with CP-nets. Computational Intelligence 20(2), 137–157 (2004)
7. Bouveret, S., Endriss, U., Lang, J.: Conditional importance networks: A graphical language for representing ordinal, monotonic preferences over sets of goods. In: Proceedings of IJCAI 2009 (2009)
8. Brafman, R., Chernyavsky, Y.: Planning with goal preferences and constraints. In: Proceedings of ICAPS 2005, pp. 182–191 (2005)
9. Brafman, R., Domshlak, C., Shimony, S.E.: On graphical modeling of preference and importance. JAIR 25, 389–424 (2006)
10. Brams, S., Kilgour, D., Zwicker, W.: The paradox of multiple elections. Social Choice and Welfare 15(2), 211–236 (1998)
11. Domshlak, C., Brafman, R.I., Shimony, S.E.: Preference-based configuration of web page content. In: Proceedings of IJCAI 2001, pp. 1451–1456 (2001)
12. Gonzales, C., Perny, P., Queiroz, S.: Preference aggregation with graphical utility models. In: Proceedings of AAAI 2008, pp. 1037–1042 (2008)
13. Gottlob, G., Greco, G., Scarcello, F.: Pure Nash equilibria: Hard and easy games. J. Artif. Intell. Res. (JAIR) 24, 357–406 (2005)
14. Kearns, M., Littman, M.L., Singh, S.: Graphical models for game theory. In: Proceedings of UAI 2001 (2001)
15. Keeney, R.L., Raiffa, H.: Decision with Multiple Objectives: Preferences and Value Tradeoffs. Wiley, Chichester (1976)
16. Koller, D., Milch, B.: Multi-agent influence diagrams for representing and solving games. In: Proceedings of IJCAI 2001, pp. 1027–1034 (2001)
17. Lang, J., Xia, L.: Sequential composition of voting rules in multi-issue domains. In: Mathematical Social Sciences, pp. 304–324 (2009)
18. Prestwich, S., Rossi, F., Venable, B., Walsh, T.: Constrained CP nets. In: Italian Conference on Computational Logic (2004)

19. Rossi, F., Venable, K.B., Walsh, T.: mCP nets: Representing and reasoning with preferences of multiple agents. In: Proceedings of AAAI 2004, pp. 729–734 (2004)
20. Wilson, N.: Extending CP-nets with stronger conditional preference statements. In: Proceedings of AAAI 2004, pp. 735–741 (2004)
21. Wilson, N.: Efficient inference for expressive comparative preference languages. In: Proceedings of IJCAI 2009, pp. 961–966 (2009)
22. Xia, L., Conitzer, V., Lang, J.: Voting on multiattribute domains with cyclic preferential dependencies. In: Proceedings of AAAI 2008, pp. 202–207 (2008)

Combining Description Logics, Description Graphs, and Rules

Boris Motik

Computing Laboratory, University of Oxford, UK

Abstract. The Web Ontology Language (OWL) is a well-known language for ontology modeling in the Semantic Web [9]. The World Wide Web Consortium (W3C) is currently working on a revision of OWL—called OWL 2 [2]—whose main goal is to address some of the limitations of OWL. The formal underpinnings of OWL and OWL 2 are provided by description logics (DLs)[1]–knowledge representation formalisms with well-understood formal properties.

DLs are often used to describe *structured objects*—objects whose parts are interconnected in complex ways. Such objects abound in molecular biology and the clinical sciences, and clinical ontologies such as GALEN, the Foundational Model of Anatomy (FMA), and the National Cancer Institute (NCI) Thesaurus describe numerous structured objects. For example, FMA models the human hand as consisting of the fingers, the palm, various bones, blood vessels, and so on, all of which are highly interconnected.

Modeling structured objects poses numerous problems to DLs and the OWL family of languages. The design of DLs has been driven by the desire to provide practically useful knowledge modeling primitives while ensuring decidability of the core reasoning problems. To achieve the latter goal, the modeling constructs available in DLs are usually carefully crafted so that the resulting language exhibits a variant of the *tree-model property* [10]: each satisfiable DL ontology always has at least one model whose elements are connected in a tree-like manner. This property can be used to derive a decision procedure; however, it also prevents one from accurately describing (usually non-tree-like) structured objects since, whenever a model exists, at least one model does not reflect the intended structure. This technical problem has severe consequences in practice [6]. In search of the "correct" way of describing structured objects, modelers often create overly complex descriptions; however, since the required expressive power is actually missing, such descriptions do not entail the consequences that would follow if the descriptions accurately captured the intended structure.

In order to address this lack of expressivity, we extended DLs with *description graphs*, which can be understood as schema-level descriptions of structured objects. To allow for the representation of conditional statements about structured objects, we also incorporated first-order rules [3] into our extension. In this way we obtain a powerful and versatile knowledge representation formalism. It allows us, for example, to describe the structure of the hand using description graphs, statements such as "if a bone in the hand is fractured, then the hand is fractured as well" using

M. Croitoru, S. Ferré, and D. Lukose (Eds.): ICCS 2010, LNAI 6208, pp. 10–12, 2010.

rules, and nonstructural aspects of the domain such as "a medical doctor is a person with an MD degree" using DLs.

To study the computational properties of our formalism, we base the DL component on the $\mathcal{SHOIQ}^+$description logic, as this DL provides the semantic underpinning of OWL 2. The resulting formalism is quite expressive, and it is unsurprising that it is undecidable. We investigate restrictions under which the formalism becomes decidable. In particular, we have observed that structured objects can often be described by a possibly large, yet bounded number of parts. For example, a human body consists of organs all of which can be decomposed into smaller parts; however, further decomposition will eventually lead to parts that one does not want or know how to describe any further. In this vein, FMA describes the skeleton of the hand, but it does not describe the internal structure of the distal phalanges of the fingers. The number of parts needed to describe the hand is therefore determined by the granularity of the hierarchical decomposition of the hand. This decomposition naturally defines an acyclic hierarchy of description graphs. For example, the fingers can be described by description graphs that are subordinate to that of the hand; however, the description graph for the hand is not naturally subordinate to the description graphs for the fingers. We used this observation to define an *acyclicity* restriction on description graphs. Acyclicity bounds the number of parts that one needs to reason with, which, provided that there are no DL axioms, can be used to obtain a decision procedure for the basic reasoning problems.

If description graphs are used in combination with DL axioms, the acyclicity condition alone does not ensure decidability due to possible interactions between DL axioms, graphs, and rules [5]. To obtain decidability, we limit this interaction by imposing an additional condition on the usage of roles: the roles (i.e., the binary predicates) that can be used in DL axioms must be separated from the roles that can be used in rules. We developed a hypertableau-based [7] reasoning algorithm that decides the satisfiability problem for our formalism, together with tight complexity bounds.

All proofs and additional decidability and complexity results for the case when DL axioms are expressed in $\mathcal{SHOIQ}^+$can be found in [8].

References

1. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation and Applications, 2nd edn. Cambridge University Press, Cambridge (2007)
2. Grau, B.C., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P., Sattler, U.: OWL 2: The next step for OWL. Journal of Web Semantics 6(4), 309–322 (2008)
3. Horrocks, I., Patel-Schneider, P.F.: A Proposal for an OWL Rules Language. In: Proc. WWW 2004, New York, NY, USA, pp. 723–731 (2004)
4. Horrocks, I., Sattler, U.: A Tableau Decision Procedure for SHOIQ. Journal of Automated Reasoning 39(3), 249–276 (2007)
5. Levy, A.Y., Rousset, M.-C.: Combining Horn Rules and Description Logics in CARIN. Artificial Intelligence 104(1-2), 165–209 (1998)

6. Motik, B., Cuenca Grau, B., Sattler, U.: Structured Objects in OWL: Representation and Reasoning. In: Proc. WWW 2008, Beijing, China (2008)
7. Motik, B., Shearer, R., Horrocks, I.: Hypertableau Reasoning for Description Logics. Technical report, University of Oxford (2008); Submitted to an International Journal
8. Motik, B., Grau, B.C., Horrocks, I., Sattler, U.: Representing Ontologies Using Description Logics, Description Graphs, and Rules (2009) (submitted to a Journal)
9. Patel-Schneider, P.F., Hayes, P., Horrocks, I.: OWL Web Ontology Language: Semantics and Abstract Syntax, W3C Recommendation (February 10, 2004), `http://www.w3.org/TR/owl-semantics/`
10. Vardi, M.Y.: Why Is Modal Logic So Robustly Decidable? In: Proc. of a DIMACS Workshop on Descriptive Complexity and Finite Models, pp. 149–184 (1996)

Practical Graph Mining

Mohammed J. Zaki

Department of Computer Science, Rensselaer Polytechnic Institute
Troy, New York
zaki@cs.rpi.edu

Abstract. Given the ubiquity of large-scale graphs and networks, graph mining has rapidly grown to occupy a center-stage within data analysis and mining. In this talk I will present our recent work on mining interesting, representative subgraph patterns from very large graph databases. Some aspects of graph indexing may also be covered. I'll conclude with thoughts on future challenges and research directions.

M. Croitoru, S. Ferré, and D. Lukose (Eds.): ICCS 2010, LNAI 6208, p. 13, 2010.

Use of Domain Knowledge in the Automatic Extraction of Structured Representations from Patient-Related Texts

Galia Angelova

Institute for Parallel Processing, Bulgarian Academy of Sciences
25A Acad. G. Bonchev Str., 1113 Sofia, Bulgaria
galia@lml.bas.bg

Abstract. Domain knowledge is essential resource in Information Extraction (IE) from free text since it supports the decisions about structuring the extracted text objects into domain statements. Thus manually-created conceptual structures enable the semantic representation of textual information. This paper discusses the role of domain knowledge in information extraction of structured data from patient-related texts. The article shows that domain knowledge is encoded not only in the conceptual structures, which provide the ontological framework for the IE task, but also in the IE templates that are designed to capture domain semantics. A prototype system and IE examples of domain knowledge usage are considered together with results of the current prototype evaluation.

1 Introduction

Medical information is exchanged among experts predominantly in textual format. Studying hospital patient records, which discuss a single hospital episode, we discover that the most important findings, opinions, summaries and recommendations are stated as free text while clinical data usually support the textual statements or clarify particular facts. In this way the essence of patient-related information is communicated as unstructured text messages. Furthermore, nowadays the electronic text archives in the medical domain include much more than patient records; the number of scientific papers is rapidly increasing as well as the document repositories at various web sites. The volume of texts in the (bio)medical domain is growing so fast that text mining is viewed as the only means to identify, extract and store facts about entities and relationships of interest. The challenging task of medical text processing integrates different approaches for text and data mining, information retrieval and extraction in order to achieve the ultimate objective - semantic representation of textual medical information. Declarative conceptual models of domain knowledge are considered as ontological backbones which support the identification of important facts in the text and their translation to tractable internal structures. Most generally, domain knowledge has three roles: *(i)* to support the algorithms for partial text

M. Croitoru, S. Ferré, and D. Lukose (Eds.): ICCS 2010, LNAI 6208, pp. 14–27, 2010.

analysis, which discover important facts in the text; *(ii)* to provide the framework where the internal structured representation is constructed and *(iii)* to enable the inference procedures which process the internal semantic structures.

Natural Language Processing (NLP) applications in the medical domain need background knowledge to properly interpret the terms and words that have been encountered in the text. A major NLP problem is that the implicit relations between text units have to be explicated while the text is transferred into internal structured representation. Therefore conceptual models should contain detailed description of domain entities and domain-specific relations (like e.g. *has-location*, *clinically-associated-with* etc.). However, even when a single standardised ontology is integrated within a NLP system, it is not trivial to automatically link text units to ontology elements and fix the mapping from text items to ontology labels. One needs at first a domain model labeled by the terms in the respective natural language. Other major obstacles are: *(i)* inconsistent and imprecise practice in the terminological naming of (bio)medical concepts and *(ii)* incomplete ontologies as a result of rapid knowledge expansion [1]. Despite the difficulties there is a growing interest in the area of biomedical text processing; a variety of domain models exist, often labeled by English terminology but resources in other languages appear and are constantly growing. Especially for the IE task, which extracts facts about prespecified types of entities and relationships among them, the overview [1] distinguishes between *ontology-based* and *ontology-driven* IE approaches. The ontology-based systems map terms occurring in text to the ontology labels while the ontology-driven approaches actively use domain knowledge to strongly guide and constraint the analysis. In this paper we present in more details the role of conceptual structures in a prototype which should be classified as an ontology-driven IE system.

The article is structured as follows. Section 2 introduces the project context and overviews some related work. Section 3 considers the role of domain knowledge in text analysis and interpretation. Section 4 discusses the evaluation of the current prototype. Section 5 contains the conclusion and summarises plans for further work.

2 Project Background and Related Work

Information Extraction arose in computational linguistics in the late 80ties as an approach for partial NL understanding, i.e. extraction of entities and relations of interest without full text understanding (see e.g. [2]). Various IE applications work on free text and produce the so-called templates: fixed-format, unambiguous representations of available information about preselected events. The IE process runs after text preprocessing (tokenisation, lemmatisation) and the task is usually split into several components [3]:

- *Named entity recognition* - finds entities (e.g. nouns, proper names) and classifies them as person names, places, organisations etc.;
- *Coreference resolution* - finds which entities and references (e.g. pronouns) are identical, i.e. refer to the same thing;

- *Template element construction* - finds additional information about template entities - e.g. their attributes;
- *Template relation construction* - finds relations between template entities;
- *Scenario template production* - fills in the event template with specified entities and relationships.

These five steps are convenient for performance evaluation which enables the comparison of IE systems (because one needs intermediate tasks where the performance results can be measured). Recent achievements for English are: more than 95% accuracy in Named entities recognition, about 80% in template elements construction and about 60% in scenario template production [3].

The prototype, which is described here, extracts structured representations of patient status data in Bulgarian language. It deals with hospital records of patients who are diagnosed with different forms of diabetes. The Patient Record (PR) text is usually 2-3 pages long and consists of the following standard sections: *(i)* personal details; *(ii)* diagnoses of the leading and accompanying diseases; *(iii)* anamnesis (personal medical history), including current complains, past diseases, family medical history, allergies, risk factors; *(iv)* patient status, including results from physical examinations; *(v)* laboratory and other tests findings; *(vi)* medical examiners comments; *(vii)* discussion; *(viii)* treatment; *(ix)* recommendations. Here we shall discuss examples concerning the extraction of patient status from PR section *(iv)*. The IE process runs stepwise and is accomplished into stages which combine text analysis and building of structured representations. Adopting the classical IE scheme, we can split the main IE steps into components as shown in Fig. 1:

(Step 1) Named entities recognition, implemented by text analysis and mappings of text terminology to domain ontology vocabulary. In this way the system finds the anatomic organs, diagnoses, sympthoms, treatment procedures etc.;

(Step 2) Coreference resolution, which might be a very complicated procedure when processing e.g. the patient medical history; it is simpler for the patient status section due to the specific telegraphic style of PR texts;

(Step 3) Selection of template to be filled in and determining the text fragment to be analysed for performing the extraction task;

(Step 4) Sentence analysis by regular expressions within the selected text fragment which identifies template entities and their attributes;

(Step 5) Finding relations between entities and their attributes;

(Step 6) Dynamic template extension to capture all relevant entities, attributes and relations which occur in the selected text fragment;

(Step 7) Filling in the template, which might include also adding default values to empty template slots [4].

In this article we shall discuss steps 3, 6 and 7. They rely on domain knowledge, which in general contains:

- conceptual model (a semantic network of body parts is shown below),
- templates for capturing the extracted facts and
- default values and correlations as data-driven status characteristics [4].

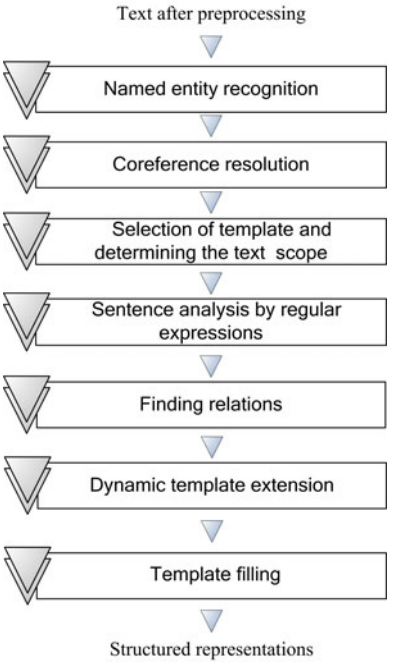


Fig. 1. Major IE steps with focus on template filling (adopted from [3])

Major obstacles to the IE process, shown in Fig. 1, are:

(i) Domain-specific: occurrences of terminological and phrasal variations in the PR text, which prevent the correct recognition of textual elements and their mapping to labels of conceptual units at step 4;

(ii) Corpus-specific: missing descriptions in the PR texts which hamper step 7 (they are due to tacit knowledge or incomplete descriptions entered by the medical expert: as [4] reports, only 86% of the PRs in our corpus discuss explicitly the skin colour, 63% - the fat tissue, about 42% - the skin turgor and elasticity, and 13% - the skin hydration);

(iii) General: lack of language and ontological resources and tools. For instance, no Named Entity Recognition module has been implemented for Bulgarian entities in the medical domain; the syntactic rules for shallow sentence analysis are constructed for the first time; there are no conceptual resources labeled by Bulgarian medical vocabulary except ICD-9 (the International Classification of Diseases) and ICD-10 without clinical terms; the list of drugs and medications is supported with Latin names by the Bulgarian Drug Agency, even for drugs produced in Bulgaria [5], but medications in the PR texts are predominantly referred to in Bulgarian language and so on. In this way the background IE infrastructure in our project is to be manually developed from scratch.

There are numerous projects in biomedical text analysis. Some performance results are recently reported in CLEF - the Clinical E-Science Framework project which was running in the UK for five years. The paper [6] discusses the automatic entity recognition in biomedical texts using a gold standard corpus of 77 English documents with 2124 entities of five types (condition, drug or device, intervention, investigation and locus). Studying in depth different methods, ranging from dictionary look-up to machine learning approaches, the authors of [6] report about maximal success of 83% in entities recognition and conclude that dictionary look-up is a promising basic strategy for terminology recognition.

The advantages of ontology-driven approaches to medical IE tasks are seen in [7] and [8]. These articles present MedScan, an engine that efficiently processes sentences from MEDLINE abstracts and produces a set of semantic structures representing the meaning of each sentence. MedScan in 2003 is aimed at the extraction of information about pathways and molecular networks, so the system is tuned to process sentences containing relevant words. After parsing, each sentence is represented as semantic frame which stores the logical relationships between the sentence words. The ontological interpreter evaluates the outputs of the NLP component and converts the valid ones into ontological representation. The following accuracy figures are reported: processed 4.6 million sentences, with 34% correctly parsed sentences but the analysis of errors shows that with larger lexicon and better grammar MedScan can extract protein function information with high (above 90%) precision [7]; 91% correct extraction of human protein interactions in 2004 [8]. MedScan applies the ontology as a filter to select correct semantic sentence structures. Without the ontological filter, there would be too many extracted statements concerning text units which are irrelevant to the target subject; therefore putting the emphasis on the domain interpretation ensures the consideration of relevant sentences only. Similarly, in the prototype we present here, the decision-making process exploits the ontological vocabulary of domain objects and relations. In this way large text fragments remain isolated while performing the analysis of the entities of interest.

Semantic technologies are widely used in research projects dealing with medical terminology and analysis of medical text. The paper [9] exemplifies important conceptual relations among medical concepts and proposes to tackle the polysemy in medical lexicons by well-defined conceptual models. Indexing of medical terms in free text and translation of the extracted structures to conceptual graphs is considered in [10]. Text mining is applied to collections of medical documents, to enable semi-automatic acquisition of conceptual structures [11]. This approach helps to identify the relations between the entities in certain corpus. Mark-up tags for medical entities in patient records are considered in [12]; they cover a wide variety of semantic descriptions in patient clinical profiles.

3 Domain Knowledge in Text Analysis and Interpretation

To extract structured representations from free text, we assume that text units have to be identified and interpreted as domain entities, i.e. they have to be

mapped to certain conceptual object or conceptual relation in the domain. Manually-created conceptual models provide the necessary framework for semantic interpretation of textual information. At least four types of conceptual resources are needed to back up the extraction of structured representations in the case of hospital PRs. They provide domain knowledge about:

(i) diseases and their symptoms and complications;
(ii) drugs, medications, therapeutic procedures;
(iii) anatomy - body parts, organs, tissues, clinical findings etc. and
(iv) medical appliances that might be mentioned in text.

In this article we focus our attention to the usage of domain knowledge in text analysis since its role in inference is well-known.

3.1 Text Segmentation and Scoping

To perform step 3 of the algorithm shown at Fig. 1, the IE system has to find in the PR text a term referring to some relevant anatomic organ, e.g. skin, eyes, thyroid gland, neck, limbs, etc. The PR status section contains mostly short declarative sentences in present tense or list of separate phrases. In our representative corpus of 1697 PRs we find descriptions of more than 45 different anatomic organs and status conditions. Preliminary data-driven observations explicate typical text genre features, for instance there is a default sequence of describing the diabetic patient status (*general attributes - skin - head - neck - respiratory organs* etc). However, different organs are considered with various levels of details and their descriptions have different length. Therefore it is important to identify the discussion boundaries for each particular entity of interest. The scope is determined using domain knowledge concerning the human body and its parts, which helps the IE system to decide where an organ description ends and another one begins. Lists of medical terms and phrases, collected from the experimental corpus, help significantly to design the text analysis rules.

Figure 2 shows an excerpt of semantic network which encodes important relations among human body parts. Knowledge about *head* and associated organs and tissues helps to scope the description of *eyes* and *thyroid gland* status. We shall illustrate the main ideas in this section using the following examples:

Sample text 1: Глава - без патологични изменения. Очни ябълки - правилно положени в орбитите, без очедвигателни нарушения. Език - суховат, зачервен. Шия - запазена подвижност. Щитовидна жлеза не се палпира увеличена, пресен несекретиращ оперативен цикатрикс. Нормостеничен гръден кош, ...

Head - without pathological changes. Eyes - correctly placed in the eye-sockets, without disturbances in the eye-movements. Tongue - dry, red. Neck - preserved mobility. Thyroid gland does not palpate enlarged, fresh non-secreting operative cicatrix. Normostenic thorax, ...

Here the IE system finds the term *head* in the first sentence and runs the IE process in order to extract *head* status. The second and third sentences contain terms referring to body parts, linked to *head* by the *has-location* and *part-of*

relations - *eye* and *tongue* which is located in the *head* part *mouth*. Mapping these terms to the concepts and relations at Fig. 2, the IE system considers sentences 1-3 as status descriptions of *head*, *eye*, and *tongue* correspondingly. The fourth and fifth sentences contain the terms *neck* and *thyroid gland* which usually appear in consecutive sentences. The *neck* is not *head* part and therefore, the IE system considers the fourth sentence as a beginning of new body part description. The *cicatrix* in the fifth sentence refers to *neck* despite the fact that it is mentioned together with the *thyroid gland* in the same sentence. The discussion continues by presenting the *thorax* status in the sixth sentence which signals focus shift to another body part. Usually new descriptions start in another sentence but all statements are mixed into one paragraph. Other examples are:

Sample text 2: Eyes correctly placed in the eye-sockets, slow pupil reactions. Strong vision impairment. Neck - preserved mobility. Thyroid gland enlarged IA stage. Lymph nodes do not palpate. Emphysematous thorax, ...

Sample text 3: Head - without pathological changes. Eyes correctly placed in the eye-sockets, normal pupil reactions. Bilateral exophthalmos, without disturbances in the eye-movements. Neck - preserved mobility, palpable veins. Thyroid gland not enlarged. Vesicular breathing, ...

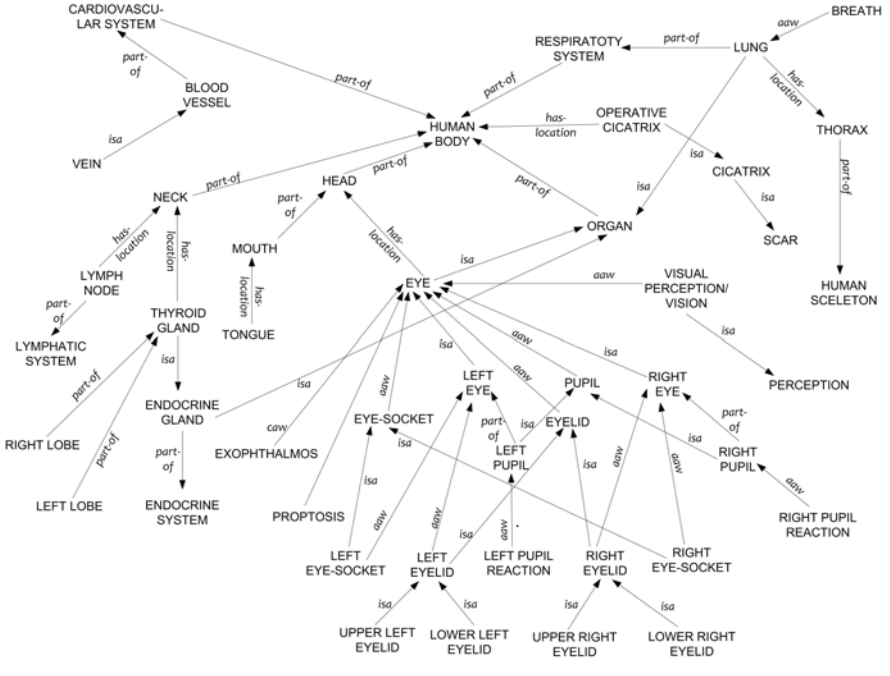


Fig. 2. Semantic network of concepts and five conceptual relations: *isa*, *part-of*, *has-location*, *anatomically-associated-with (aaw)* and *clinically-associated-with (caw)*

In the sample text 2, the *lymph nodes* mentioned in the fifth sentence will be interpreted as *neck lymph nodes* because this sentence is located immediately after the discussion of *neck* and *thyroid gland* but *lymph nodes* are not part of the *thyroid gland*. The occurrence of terms like *thorax* and *breathing* marks the beginning of new body part discussions. Our present algorithm fills in the IE templates only by status values which are extracted from the scoped text fragments. In other words, if step 3 of the algorithm at Fig. 1 is run for certain selected body part, it will not be performed again for the particular PR text. No record in the current corpus of 1697 PRs contains a status description which is split into two disconnected text fragments.

About 96% of the PRs in our corpus present organ descriptions as a sequence of declarative nominal phrases, organised into short sentences. In this way the shallow analysis of phrases and sentences by cascades of regular expressions, which is proposed in [4,13], proves to be a successful approach for structuring the statements concerning patient status. For each particular body part of interest, there is a predefined template, where text units are stored as conceptual entities during the domain interpretation phase. Evaluation results are presented in section 4.

3.2 Mapping Text Units to IE Template Slots

We shall consider in more details the template slots which structure available information about eyes and thyroid. The PRs contain different status descriptions for each individual patient. Sometimes the discussion is reduced to a short note that no deviations are encountered, e.g. *thyroid gland without changes* or the remark is omitted at all. Other PRs, however, contain long descriptions concerning dozens of status attributes. Moreover, many words are used as terminological variants, including adjectives coming from terms-nouns. For instance, our corpus contains about 90 words and phrases which characterise skin colour [4]. Despite the repetitive patterns of word and phrase choice and the medical language limitations, there are too many entities and relationships which are expressed in various ways. To cope with the free text variability, the IE system has to meet at least two challenges:

(i) to ensure certain flexibility of template slots and techniques for dynamic template extension/compression, when longer descriptions are met and

(ii) to support mappings of many different words and phrases into a prestructured set of attributes with typical categories.

The IE template design starts by studying domain corpus together with medical professionals. At first we have constructed some "canonical" templates for capturing information about the status of important organs, like eyes and thyroid gland as shown at Fig. 3(A). These templates enable direct production of predicate-argument logical statements since the predicate names are given by the slot names. The templates at Fig. 3(A) are sufficient to capture the standard statement *"Eyes correctly placed in the eye-sockets, normal pupil reactions. Thyroid gland not enlarged"* which is the default patient status. However,

dynamic template extension is needed when more details appear in the text. These detailed statements usually discuss *(i)* the status of all organ parts, *(ii)* the attributes and functions of the respective organ or body part, as well as *(iii)* clinical considerations concerning diseases of the organs and their complications and treatment.

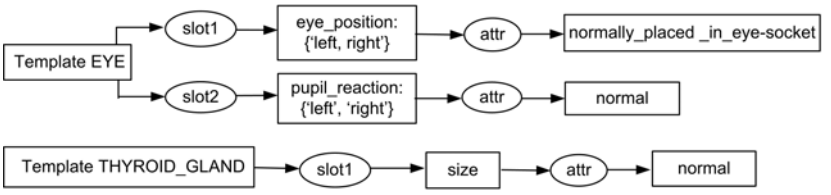
The templates at Fig. 3(B) are generated after the analysis of the PR texts *"Eyes correctly placed in the eye-sockets, normal pupil reactions, oedema of the upper eyelids"* and *"Eyes - proptosis of the right eye, without disturbances in the eye-movements"*.

The template shown at Fig. 3(C) is generated after the analysis of another PR text *"Eyes - ptosis of left eyelid, strongly impaired vision of the left eye"*. Two new slots are generated: for the left upper eyelid and the impaired vision of the left eye. It is very important to notice that the referents in Conceptual Graphs [14] are quite useful for template design since they enable a more compact specification of slots and their attributes.

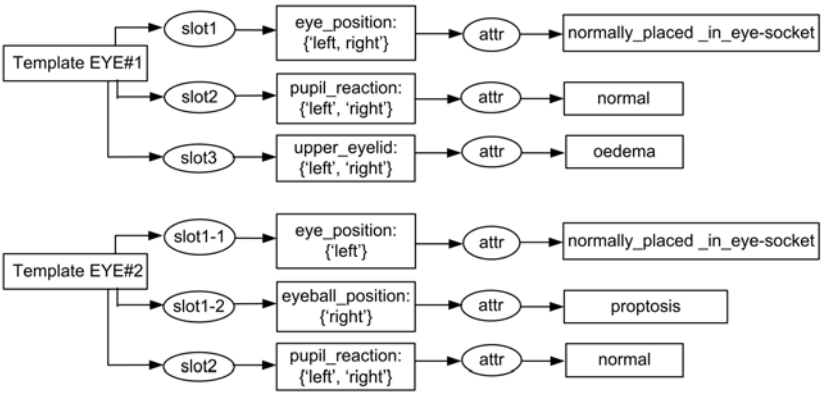
The template at Fig. 3(D) is generated for the description *"Thyroid - enlarged IB stage. Nodules on the left lobe"*.

Regarding the coreference resolution, for the PR texts we assume that all entities within the focused fragment of consecutive sentences refer to the same object of interest. Therefore, all these entities are to be captured in the corresponding object template by its dynamic extension. For instance, the template at Fig. 3(D) reflects the assumption, that *left lobe* in the second sentence is related to the *thyroid* since this is the only specific organ with left lobe mentioned in the previous phrase; therefore it is the only possible antecedent. In this way designing an IE template requires to discover from the PR text the essential information relationships and their linguistic expressions. Studying a representative corpus of PR texts, we see the basic classes of domain objects, how they are modified and related to other objects. The extraction of linguistic patterns combines semi-automatic and manual acquisition but it is important to optimise the number of these patterns, in order to predefine template slots in a flexible manner.

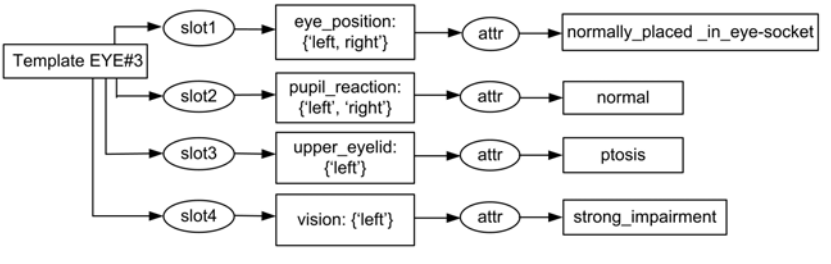
Another important issue is the number of attributes which are assigned as values to different status characteristics. There could be a variety of words or phrasal descriptions which describe the patient status. To simplify and unify the internal structures, the article [4] proposes a "normalisation" of attributes discussing the status, by introducing a scale of *normal*, *worse* and *bad* conditions. So various expressions like *increased / slightly increased / moderately developed / well developed / reduced / highly reduced / strongly reduced* etc. are mapped onto these predefined categories. There are many negative expressions in Bulgarian PR texts which were studied earlier [15]; in out present IE prototype the negative phrases are considered as phrasal text units which often denote the normal conditions like e.g. *no pathological changes*, *without changes*, *missing changes*, *does not palpate (enlarged)* and so on. In this way the template design is influenced by linguistic studies of the domain corpus but medical knowledge also needs to be taken into account. Linguistic analysis, performed together with domain



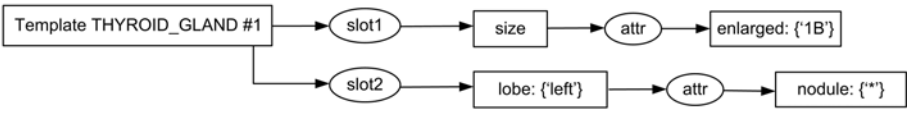
(A) Default templates about eyes and thyroid gland



(B) Templates extended to capture text descriptions about status of eyelids and eyeballs



(C) Template extended to capture text descriptions about status of eyelids and reduced vision abilities



(D) Template extended to capture information about thyroid gland status

Fig. 3. Default templates and their dynamic augmentations to particular PR descriptions

experts, helps to identify the essential facts to be extracted and the corresponding templates for their storage.

3.3 Heuristic Strategies for Analysis and Template Filling

Studying the experimental corpus, we have acquired the following heuristic procedures for text fragmentation and entities linking:

Strategy for determining phrases and sentences, which discuss a body part of interest X: Identify in the PR status text a sentence S which contains a term referring to X (searching with priority for X which occupies the subject position of S). Consider the adjacent sentences $\{S_1, S_2, ..., S_u\}$ which contain only terms referring to conceptual entities as follows:

- body parts $\{X_1, X_2, ..., X_n\}$ that are linked by relations *isa*, *part-of* and *has-location* to X,
- body parts $\{Y_1, Y_2, ..., Y_m\}$ where Y_i, $1 \leq i \leq m$ are *anatomically-associated-with* X or $\{X_1, X_2, ..., X_n\}$ and
- diseases, abilities, features $\{Z_1, Z_2, ..., Z_k\}$ where Z_j, $1 \leq j \leq k$ are *clinically-associated-with* X, $\{X_1, X_2, ..., X_n\}$ or $\{Y_1, Y_2, ..., Y_m\}$.

Fill in the predefined IE template for X using the entities extracted from the sentences S, $\{S_1, S_2, ..., S_u\}$ and evaluate the precision and recall of the resulting structured representation.

Strategy for resolving the referential links: Let the IE system performs the template filling process for entity X on sentences S, $\{S_1, S_2, ..., S_u\}$. Let $T \neq X$ be a term occurring in the sentences S, $\{S_1, S_2, ..., S_u\}$ which is included in the system medical lexicon. Map T to the predefined slots of the template for X or to the predefined values, which are eligible for these slots. In case that the mapping is impossible, skip T as irrelevant for the template. Evaluate the precision and recall of the resulting structured representation.

These heuristic procedures essentially facilitate the template filling. For instance, the *cicatrix* in the sample text 1 will be stored as a value in the template slot for *scars* at the *neck*. Otherwise, due to linguistic considerations, the *cicatrix* could be wrongly related to the *thyroid gland* which is discussed in the same sentence. But the latter choice is incorrect for the PRs of diabetic patients where only skin scars are described. In this way the predefined template slots for *neck* and *thyroid gland* enable to constraint and shape the information structuring.

3.4 Declarative Conceptual Resources in the Medical Domain

There is a variety of medical nomenclatures and ontologies. We have considered them and concluded that no single one is readily suited for our purposes. The language vocabulary is an essential problem since only ICD-9 and ICD-10 are translated to Bulgarian language and can be directly used as a basic terminological lexicon in the IE tasks. The available taxonomies of body parts in Bulgarian can

only be a starting point for knowledge-intensive NLP. As we see at Fig. 2, the IE task needs domain knowledge which represents various interrelations between the entities. ICD contains the list of diseases but the IE system needs to know which organs are affected, what are the symptoms and the treatment etc., in order to process correctly the PR content. So we have to build manually a conceptual model which is specifically tailored towards the representation of entities and relations with respect to diabetes interpretation in the NLP tasks.

A taxonomy of diabetes is adopted for our purposes using the sources at the Bioportal cite [16]. It contains some 150 terms for different types of diabetes and their complications. This taxonomy is highly relevant for the analysis of the PR anamnesis (personal medical history). The list of drugs used in Bulgaria contains some 5000 items [5] but less than 200 occur in the corpus we have at present. The drugs has to be classified into groups according to their substances and ways of admission, with definition of their (side-)effects and so on. We assume that a conceptual model of 1000 entities might be sufficient to back-up the development of an IE research prototype in the domain of diabetes. Currently the conceptual model is under progressive elaboration.

4 Evaluation

In-depth experiments and assessment of our present IE procedures were performed using a corpus of 197 PRs as a training set and another 1500 PRs as a test set [13]. The corpus contains anonymised PRs which include phrases discussing the patient sex, age and diabetes duration. The extraction of these attributes was evaluated as well. There are few PRs without any description of organ status but they are removed from the evaluation figures in Table 1. The accuracy is measured by the *precision* (percentage of correctly extracted entities as a subset of all extracted entities), *recall* (percentage correctly extracted entities as a subset of all entities available in the corpus) and their harmonic mean $F - measure$: $F = 2 * Precision * Recall \ / \ (Precision + Recall)$.

Table 1 shows relatively low recall values for *sex*, *duration* and *neck descriptions*, which means that our present IE algorithm does not identify all descriptions presented in the corpus. Regarding the precision, the cases of incorrect extraction are due to complex sentence structures in the PR text which need to be processed by a deep syntactic analyser and more complicated extraction patterns. Further details concerning the evaluation can be found in [13].

Table 1. IE performance evaluated for patient age, sex, diagnoses, illness duration and status attributes

Feature	Age	Sex	Diagnose	Diabetes duration	Thyroid gland	Neck
Precision	88.89	80.00	98.28	96.00	94.94	95.65
Recall	90.00	50.00	96.67	83.33	90.36	88.00
F-measure	89.44	61.54	97.47	89.22	92.59	91.67

Compared to the performance of other IE systems (cf. Section 2), the values in Table 1 look relatively good. However, our IE task is tested on a rather narrow domain and specific text genre. It is also clear that we need more data to properly develop the algorithms for production of dynamic templates. Currently we have designed one base template for each anatomic organ but more flexible extraction schemes are needed to tackle other PR types. We consider our present achievements as work in progress, which has to be developed further.

5 Conclusion and Future Work

The article presents current results in extraction of patient status data from medical text. Information extraction is tuned to discover certain preliminary selected entities and relationships; it aims at the partial analysis of free text and the production of unambiguous structures in fixed format. This paper shows that domain-specific IE is based on explicitly-declared domain knowledge which constraints and guides the extraction process. The article discusses how templates are filled in one by one from disconnected text fragments. There are no best practices of how to recognise the sentences which "fit" certain template; heuristic observations help to develop IE strategies that are evaluated on large corpora in practical settings. Despite the successful extraction of certain facts concerning the patient status, the progress in the medical domain is incremental and slow as it requires much effort for the construction of conceptual resources with vocabulary in the respective natural language, lexicons with medical terminology, NLP environment and basic information infrastructure.

Medical patient records contain complex temporal structures which are connected to patient case history and the family history. It would be useful for a hospital information system to monitor different hospital episodes of the patient thus supporting temporal information as well [17]. Another interesting question is related to the automatic recognition of illness duration and the periods of drug admission. This article presents our first results in processing of PR temporal information. We plan to develop algorithms for discovering more complex relations and other dependences, which is a target for our future work.

Acknowledgement. The research work presented in this paper is supported by grant №DO 02-292, funded by the Bulgarian National Science Fund in 2009-2011.

References

1. Spasic, I., Ananiadou, S., McNaught, J., Kumar, A.: Text mining and ontologies in biomedicine: Making sense of raw text. Briefings in Bioinformatics 6(3), 239–251 (2005)
2. Grishman, R., Sundheim, B.: Message understanding conference - 6: A brief history. In: Proceedings of the 16th International Conference on Computational Linguistics COLING 1996, Copenhagen (July 1996)

3. Cunningham, H.: Information Extraction, Automatic. In: Encyclopedia of Language and Linguistics. Elsevier, Amsterdam (2005), http://gate.ac.uk/sale/ell2/ie/main.pdf (last visited April 2010)
4. Boytcheva, S., Nikolova, I., Paskaleva, E., Angelova, G., Tcharaktchiev, D., Dimitrova, N.: Extraction and Exploration of Correlations in Patient Status Data. In: Savova, G., Karkaletsis, V., Angelova, G. (eds.) Biomedical Information Extraction, Proceedings of the International Workshop Held in Conjunction with RANLP 2009, Borovets, Bulgaria, September 18, vol. 18, pp. 1–7 (2009)
5. Bulgarian Drug Agency, http://www.bda.bg/index.php?lang=en (last visited April 2010)
6. Roberts, A., Gaizauskas, R., Hepple, M., Guo, Y.: Combining terminology resources and statistical methods for entity recognition: an evaluation. In: Proc. of the Sixth Int. Conf. on Language Resources and Evaluation (LREC 2008). CLEF Clinical E-Science Framework, University of Sheffield (2008), http://nlp.shef.ac.uk/clef/ (last visited April 2010)
7. Novichkova, S., Egorov, S., Daraselia, N.: MedScan, a NL processing engine for MEDLINE abstracts. Bioinformatics 19(13), 1699–1706 (2003)
8. Daraselia, N., Yuryev, A., Egorov, S., Novichkova, S., Nikitin, A., Mazo, I.: Extracting human protein interactions from Medline using a full-sentence parser. Bioinformatics 20(5), 604–611 (2004)
9. Gangemi, A., Pisanelli, D.M., Steve, G.: Understanding Systematic Conceptual Structures in Polysemous Medical Terms. In: Marc Overhage, J. (ed.) Proc. of AMIA An. Symposium on Converging Information, Technology and Health Care (2000)
10. Denecke, K., Kohlhof, I., Bernauer, J.: Use of Multiaxial Indexing for IE from Medical Texts. In: Proc. FCTC 2006, Int. Workshop on Foundations of Clinical Terminologies and Classifications, Timisoara, Romania, ROMEDINF (April 2006)
11. Lee, C.H., Khoo, C., Na, J.C.: Automatic identification of treatment relations for medical ontology learning: An exploratory study. In: McIlwaine, I.C. (ed.) Knowledge Organization and the Global Information Society: Proc. of the Eighth Int. ISKO Conference, pp. 245–250. Ergon Verlag, Wurzburg (2004)
12. Zhang, Y., Patrick, J.: Extracting Semantics in a Clinical Scenario. In: Roddick, J.F., Warren, J.R. (eds.) Proc. Australasian Workshop on Health Knowledge Management and Discovery (HKMD 2007), CRPIT, Ballarat, Australia, ACS, vol. 68, pp. 241–247 (2007)
13. Boytcheva, S., Nikolova, I., Paskaleva, E., Angelova, G., Tcharaktchiev, D., Dimitrova, N.: Structuring of Status Descriptions in Hospital Patient Records. In: The Proc. 2nd Int. Workshop on Building and Evaluating Resources for BioMedical Text Mining, associated to the 7th Int. Conf. on Language Resources and Evaluation (LREC 2010), Malta (to appear) (May 2010)
14. Sowa, J.: Conceptual Information Processing in Mind and Machines. Reading, MA (1984)
15. Boytcheva, S., Strupchanska, A., Paskaleva, E., Tcharaktchiev, D.: Some Aspects of Negation Processing in Electronic Health Records. In: Proc. of International Workshop Language and Speech Infrastructure for Information Access in the Balkan Countries, Borovets, Bulgaria, pp. 1–8 (2005)
16. BioPortal, http://bioportal.bioontology.org/visualize/13578/Diabetes_Mellitus (last visited April 2010)
17. Boytcheva, S., Angelova, G.: Towards Extraction of Conceptual Structures from Electronic Health Records. In: Rudolph, S., Dau, F., Kuznetsov, S.O. (eds.) Conceptual Structures: Leveraging Semantic Technologies. LNCS (LNAI), vol. 5662, pp. 100–113. Springer, Heidelberg (2009)

Translations between RDF(S) and Conceptual Graphs

Jean-François Baget, Madalina Croitoru, Alain Gutierrez,
Michel Leclère, and Marie-Laure Mugnier

LIRMM (University of Montpellier II & CNRS), INRIA Sophia-Antipolis, France

Abstract. Though similarities between the Semantic Web language RDF(S) and languages of the Conceptual Graphs family have often been pointed out, the differences between these formalisms have been a source of difficulties while trying to translate objets of a language into the other. In this paper, we present two such transformations, that have been implemented into the CoGUI platform, and discuss their respective strengths and weaknesses.

1 Introduction

The scope of this paper is the problem of querying hybrid knowledge bases (KBs), i.e. with several components that can be expressed in different formalisms (Conceptual Graphs, RDF(S), OWL, relational model, etc.). The ontology itself can be described using different formalisms, but we make the assumption that the ontological knowledge it contains has the same meaning in all of the KBs considered (i.e. we do not address ontology alignment or mapping problems).

More specifically, we will focus on transformations between Conceptual Graphs (CGs) [10,4] and the RDF(S) language [8], the standard for Semantic Web annotations. Given the scope of the paper, a fundamental property of such transformations is the preservation of the notion of semantic entailment (the basis for reasoning, hence querying). Other desirable properties are the natural aspect of the transformation, i.e. the conciseness and intuitiveness of the generated objects, as well as the preservation of some algorithmic properties of the language to be translated. Developing these transformations will not only provide a step towards querying hybrid KBs, but also benefit certain tasks on the Semantic Web [7] where structural, graph based optimisations (extensively addressed for Conceptual Graphs [9][4]) are needed.

In the following we detail two proposed transformations and study their properties. Both transformations preserve the semantic entailment, in a sense that we will precise, but they behave differently with respect to conciseness and intuitiveness. Both transformations are implemented in the tool CoGUI[1]. All graphs pictured in this paper are screenshots from an example designed with CoGUI and available on CoGUI's website.

2 Basic Conceptual Graphs: The Core Formalism of CoGUI

We recall in this section the main elements of the basic CG formalism. See [4] for details and tools CoGUI and CoGITaNT[2] for a faithful implementation of this framework.

[1] `http://www.lirmm.fr/cogui/`
[2] `http://cogitant.sourceforge.net/`

M. Croitoru, S. Ferré, and D. Lukose (Eds.): ICCS 2010, LNAI 6208, pp. 28–41, 2010.

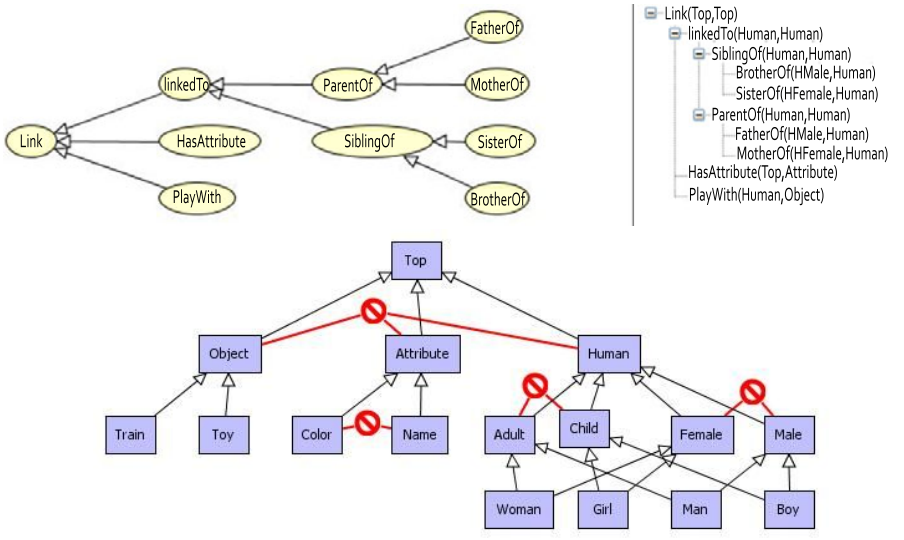


Fig. 1. Basic Conceptual Graphs (BG) Vocabulary

Ontological knowledge is encoded by a structure called the vocabulary, while factual knowledge is expressed by basic CGs.

A *vocabulary* is basically a tuple $\mathcal{V} = (T_C, T_R = (T_R^1, \ldots, T_R^k), I)$ where T_C is a partially ordered set of *concept types*, each T_R^i is a partially ordered set of *relation types of arity* i, and I is a set of *individual markers*. All these sets are pairwise disjoint and all partial orders are denoted by $\leq$. Other features may also appear in a vocabulary. A *conjunctive concept type* over a vocabulary $\mathcal{V}$ is a set $T = \{t_1, \ldots, t_p\}$ of concept types. If $T = \{t_1, \ldots, t_p\}$ and $T' = \{t'_1, \ldots, t'_q\}$ are two conjunctive concept types, then we also note $T \leq T' \Leftrightarrow \forall t'_i \in T', \exists t_j \in T$ such that $t_j \leq t'_i$. The *signature* σ maps each relation type of arity k to a k-tuple of conjunctive concept types, that encodes the maximal type of its arguments. The signature has the covariance property, meaning that if $r_2 \leq r_1$, then the i^{th} argument of r_2 is a specialization of the i^{th} argument of r_1. It is sometimes necessary, as in [1], to assert that an entity is an instance of several concept types. Finally, the vocabulary can be extended by adding *incompatibilities between (two) types*, i.e. asserting that a given conjunctive type is forbidden [6].

Fig. 1 depicts a hierarchy of relation types, their signature (e.g. $\sigma(MotherOf) = (HFemale, Human)$), and a hierarchy of concept types. The "forbidden" symbol encodes incompatibility (e.g. *Human*, *Attribute* and *Object* are pairwise incompatible).

A *basic conceptual graph* (BG) on a vocabulary $\mathcal{V} = (T_C, T_R, I)$ is a bipartite graph. The sets C and R contain respectively *concept* and *relation nodes*. A concept node $c \in C$ is labeled by a pair $(type(c), marker(c))$ where $type(c)$ is a conjunctive concept type built on T_C and *marker(c)* is either an individual marker of $\mathcal{I}$ – in that case c is said *individual* – or the generic marker $*$, possibly named by a variable, as in $*x$ – then c is said *generic*. A relation node $r \in R$ is labeled by $type(r) \in T_R$ and is linked to k concept nodes (its arguments), where k is the arity of $type(r)$.

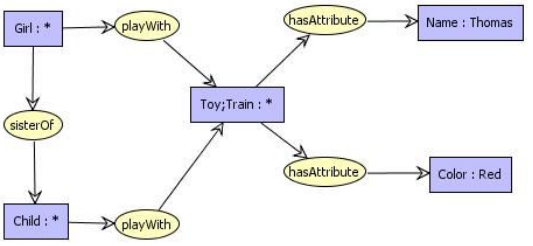


Fig. 2. BG Fact

Several concept nodes with the same individual marker or the same named generic marker denote the same entity. A BG is said to be *normal* when no two distinct concept nodes denote the same entity. Any BG can be transformed into a normal graph having the same logical semantics, called its normal form, by merging nodes that represent the same entity. The normal BG in Fig. 2, represents a fact stating that a child and its sister are playing with a toy train, which is has the name of Thomas and is red.

Note that the additional features of the vocabulary (signature and forbidden types) impose additional constraints on a BG. To respect the signature, the i^{th} argument of a relation typed r must be a specialization of the i^{th} element of $\sigma(r)$; to satisfy forbidden types, no concept node can be of a type that is a specialization of a forbidden conjunctive type. Implemented in CoGUI, these features are used to check the validity of a BG and have no other impact on reasoning.

CGs can be translated into first-order logic by a mapping classically called Φ [10]. The BG fragment is equivalent to the existential positive conjunctive fragment of first-order logic [4]. The fundamental deduction problem in this fragment is as follows: given BGs F and Q defined over a vocabulary $\mathcal{V}$, is the formula $\Phi(Q)$ the logical deduction of the formulae $\Phi(F)$ and $\Phi(\mathcal{V})$ (noted $\Phi(\mathcal{V}), \Phi(F) \vdash \Phi(Q)$)? This problem can be recast as a query answering problem: is the conjunctive query Q deducible from the KB composed of a set of facts F and a lightweight ontology $\mathcal{V}$?

The basic notion for reasoning with BGs is a graph homomorphism ("projection" in the CG world). It provides a sound and complete reasoning mechanism:

Theorem 1. *[10] [5] Let F and Q be BGs on a vocabulary $\mathcal{V}$. Then $\Phi(\mathcal{V}), \Phi(F) \vdash \Phi(Q)$ iff there is a homomorphism from Q to the normal form of F.*

Homomorphism checking (or deduction on BGs) is an NP-complete problem and is polynomial in data complexity (i.e. with respect to the size of F). Moreover, several cases with lower complexity can be obtained, mainly based on the structure of Q. Besides their interesting algorithmic properties, homomorphisms provide a visual way to express answers to a query Q. To compute homomorphisms, CoGUI relies upon a Client/Server architecture to communicate with the reasoning server CoGITaNT.

Apart from the vocabulary and the facts, CoGUI is also able to edit BG rules. These rules express knowledge of the form: "if hypothesis then conclusion", where the hypothesis and the conclusion are two BGs. In this paper, we will only consider simple rules, which do not increase the complexity of querying, i.e., rules that only add relation nodes or specialize the type of concept nodes (they are special cases of the so-called

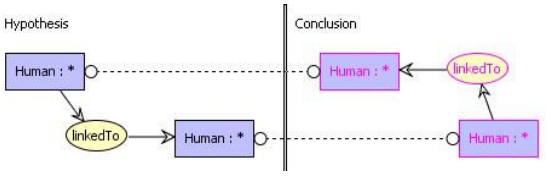


Fig. 3. BG+: Rule Example

range-restricted rules in [3], [4]). See, for example, the rule in Fig. 3, where the relation "linked to" is deemed symmetrical. We denote by BG+ the BG fragment added with rules of the above mentioned form.

From now on, we simply note $K \vdash Q$ for $\Phi(K) \vdash \Phi(Q)$, where K is a BG(+) KB (*i.e.* vocabulary, facts, and possibly rules) and Q is a BG on the same vocabulary.

3 The Semantic Web Language RDF(S)

RDF (Resource Description Framework) and its extension RDFS (RDF Schema) is a metadata model introduced by the W3C allowing the construction of semantic annotations given by a set of triples of the form *(subject, predicate, object)*. The RDF annotations are generally stored either in XML or in N3 files[3]. Fig. 4 shows a set of RDF triples in a simplified N3 notation, where names beginning with _ denote a *blank*, i.e. an anonymous resource; this set "naturally" corresponds to the BG in Fig. 2 (see Sect. 4.3). A set of RDF triples can be also visualized as a graph.

```
<:Red> <rdf:type> <:Color>.        _:b1 <:hasAttribute> <:Red>.
<:Thomas> <rdf:type> <:Name>.      _:b1 <:hasAttribute> <:Thomas>.
_:b1 <rdf:type> <:Toy>.            _:b2 <:playWith> _:b1.
_:b1 <rdf:type> <:Train>.          _:b2 <:sisterOf> _:b3.
_:b2 <rdf:type> <:Girl>.           _:b3 <:playWith> _:b1.
_:b3 <rdf:type> <:Child>.
```

Fig. 4. RDF Triples Corresponding to Fig. 2 Example

RDFS adds a lightweight ontological level structuring the vocabulary: it allows to declare classes and properties (binary predicates), to structure them by a preorder (*subClassOf* and *subPropertyOf* relations) and to define the signatures of properties via the notions of domain (*domain*) and co-domain (*range*). For instance, the following triples (in *(s,p,o)* form) "naturally" translate part of the BG vocabulary of Fig. 1. `(:Top, rdf:type, rdfs:Class)`, `(:Human, rdf:type, rdfs:Class)` and `(:Human, rdfs:subClassOf, :Top)` express that `Top` and `Human` are classes (concept types) and that `Human` $\leq$ `Top`; while triples `(:MotherOf, rdf:type,`

[3] The import / export tool of CoGUI uses the Jena (`http://jena.sourceforge.net/`) RDF parser for reading and outputting three formats: RDF/XML, N3 and TURTLE.

`rdf:Property)`, `(:MotherOf, rdfs:domain, :HFemale)`, `(:MotherOf, rdfs:range, :Human)` express that `MotherOf` is a property (relation) with signature `(HFemale, Human)`.

In this section, we define an RDF graph and three entailment relations of increasing preciseness: $\vdash_s$ (simple entailment), $\vdash_{rdf}$ (RDF entailment, which takes the so-called "RDF axiomatic triples" into account) and $\vdash_{rdfs}$ (RDFS entailment, which moreover takes the so-called "RDFS axiomatic triples" into account). Let us recall that $\vdash$ is the logical deduction in the BG fragment. The following definitions are reformulations of the ones provided by [8].

3.1 A Simple RDF

We first introduce a simplified version of RDF. Though its syntax remains the same, its semantics is weakened since no name is given any particular meaning. This RDF(S) fragment will be used as the building block in our transformations. On the other hand, we extend it to RDF*, which allows to use variables/blanks as predicate names. This is an important feature in the perspective of implementing the SPARQL query language, whose basic graph patterns rely on such a possibility.

Syntax. In what follows, we will consider 3 infinite pairwise disjoint sets of *terms*: the set $\mathcal{U}$ of *urirefs*, the set $\mathcal{L}$ of *literals*, and the set $\mathcal{B}$ of *blanks*. Among literals, we make a distinction between *plain literals*, and *typed literals*. A typed literal can be well-typed or ill-typed. The *value* $val(l)$ of a plain literal or an ill-typed literal l is the literal itself, and the value of a well-typed literal is determined by its type. For example, the value of a typed literal whose type is `rdf:XMLLiteral` is the XML value of that literal. The only type that is currently taken into account in the RDF semantics is `rdf:XMLLiteral`. An *RDF vocabulary* is a subset of $\mathcal{U} \cup \mathcal{L}$.

Definition 1 (RDF triple, RDF graph). *An* RDF triple *is an element of* $(\mathcal{U} \cup \mathcal{B}) \times \mathcal{U} \times (\mathcal{U} \cup \mathcal{B} \cup \mathcal{L})$. *An* RDF* triple *is an element of* $(\mathcal{U} \cup \mathcal{B}) \times (\mathcal{U} \cup \mathcal{B}) \times (\mathcal{U} \cup \mathcal{B} \cup \mathcal{L})$. *The first element of a triple is called its* subject*, the second its* predicate*, and the third its* object*. An RDF graph is a set of RDF triples. An RDF* graph is a set of RDF* triples.*

Note that literals appearing as subject are classically forbidden both in RDF and RDF*. This can be a problem since such triples can appear in reasonings. All further definitions and properties implicitly take that possibility in account.

If G is an RDF* graph, we call $\mathcal{U}$(G) (resp. $\mathcal{B}$(G), $\mathcal{L}(G)$, $\mathcal{T}(G)$) the set of *urirefs* (resp. *blanks*, *literals*, *terms*) appearing in G. An RDF* G graph admits a natural graph representation: a node is assigned to each term appearing as a subject or object in G, and a directed edge to each triple of G; this edge admits for origin the node assigned to its subject, and for destination the node assign to its object.

Semantics. In usual model-theoretic semantics, entities are mapped to elements of the interpretation domain and relations to a set of tuples of elements of the domain. Since RDF(S) does not consider a strict separation between entities and property names (which is considered as a requirement for the web), such an interpretation would lead

to an important mathematical problem: an element of the domain could be asserted equal to a set of tuples containing it. By encoding the extension of a property into the interpretation structure, it is possible to lift that difficulty:

Definition 2 (Interpretation). *An* interpretation *of an RDF vocabulary V is a triple $I = (\Delta, \iota, \epsilon)$ where Δ is a set of* resources *called the* interpretation *domain, $\iota : (V \cap \mathcal{U}) \rightarrow \Delta$ maps each uriref of the vocabulary to a resource, and $\iota : \Delta \rightarrow 2^{\Delta \times \Delta}$ maps each resource d to a set of pairs of resources called the* extension *of d.*

Definition 3 (Simple models). *An interpretation $I = (\Delta, \iota, \epsilon)$ of a vocabulary V is a* simple model *of an RDF or RDF* graph G iff there exists a mapping $\pi : \mathcal{T}(G) \rightarrow \Delta$ that maps urirefs to their interpretation ($\forall u \in \mathcal{U} \cap V$, $\pi(u) = \iota(u)$); maps literals to their value ($\forall l \in \mathcal{L} \cap V$, $\pi(l) =$ val(l)); and preserves triples ($\forall (s, p, o) \in G$, $(\pi(s), \pi(o)) \in \epsilon(\pi(p))$).*

Definition 4 (Simple Entailment). *Let F and Q be RDF* graphs. We say that F* simply entails *Q and note $F \vdash_s Q$ iff every simple model of F is a simple model of Q.*

Theorem 2. *Let F and Q be RDF or RDF* graphs. Then $F \vdash_s Q$ iff there exists a mapping $\pi : \mathcal{T}(Q) \rightarrow \mathcal{T}(F)$ that maps urirefs to themselves ($\forall u \in \mathcal{U}(Q)$, $\pi(t) = t$); maps literals to literals with same value ($\forall l \in \mathcal{L}(Q)$, val$(\pi(l)) =$ val(l)); and preserves triples ($\forall (s, p, o) \in Q$, $(\pi(s), \pi(p), \pi(o)) \in F$).*

3.2 RDF and RDFS Axiomatic Triples

RDF considers an infinite set $\mathcal{A}^{rdf}$ of triples said *axiomatic*, i.e. true for any RDF or RDF* graph. In the same way, RDFS considers the axiomatic set $\mathcal{A}^{rdfs}$. Both sets are infinite, due to the presence of an infinite set of properties `rdf:_i`. We can consider finite subsets by bounding the number of such properties allowed. If k is a positive integer, we denote by $\mathcal{A}^{rdf}_k$ the *finite* subset of RDF axiomatic triples defined by $\mathcal{A}^{rdf}_k = \mathcal{A}^{rdf} \setminus \{(\texttt{rdf:\_i}, \texttt{rdf:type}, \texttt{rdf:Property}) | i > k\}$. The RDFS graph $\mathcal{A}^{rdfs}_k$ is defined in a similar way. Furthermore, RDF and RDFS consider a set of semantic conditions specifying the meaning embedded by special names of RDF(S).

RDF semantics. We make here a simplification of RDFS semantics, since [8] modifies the structure of an interpretation by introducing a mapping $i' : \Delta \rightarrow \Delta$, used to define the extension of a class. But this is a redundant information, since $x \in i'(c)$ is defined as equivalent to "x has type c", that we can already encode in RDF interpretations.

Definition 5 (RDF and RDFS interpretations). *An interpretation $I = (\Delta, \iota, \epsilon)$ of a vocabulary V is an* RDF interpretation *of V iff I is a simple model for every RDF axiomatic triple, and I satisfies each RDF semantic rule. If, moreover, I is a simple model for every RDFS axiomatic triple and satisfies each RDFS semantic condition, then I is said an RDFS interpretation.*

Before expliciting some of these semantic conditions, let us first define RDF and RDFS entailments:

Definition 6 (RDF(S) Entailment). *Let F and Q be RDF or RDF* graphs. We say that F* RDF entails *(resp.* RDFS entails*) Q and note $F \vdash_{rdf} Q$ (resp. $F \vdash_{rdfs} Q$) iff every RDF (resp. RDFS) interpretation that is a model of F is also a model of Q.*

RDF semantic conditions An interpretation $I = (\Delta, \iota, \epsilon)$ satisfies the RDF semantic condition iff:

1. for every resource $\delta \in \Delta$ with $\epsilon(\delta) \neq \emptyset$, $(\delta, \iota(\texttt{rdf:Property})) \in \epsilon(\iota(\texttt{rdf:type}))$;
2. for every typed literal l whose type is `rdf:XMLLiteral`, l is well-typed iff $(\mathit{val}(l), \iota(\texttt{rdf:XMLLiteral})) \in \epsilon(\iota(\texttt{rdf:type}))$;

RDFS semantic rules In the same way, an RDFS interpretation must satisfy some semantic conditions. A complete list of these conditions can be found in [8]. F.i., the two following semantic conditions state 1) that if a property p has domain c and (s, p, o) is asserted, then o has type c; and 2) that if x has type c and c is a subclass of c', then x has type c'.

1. if $(p, c) \in \epsilon(\iota(\texttt{rdfs : domain}))$ and $(s, o) \in \epsilon(p)$, then $(s, c) \in \epsilon(\iota(\texttt{rdf : type}))$.
2. if $(x, c) \in \epsilon(\iota(\texttt{rdf : type}))$ and $(c, c') \in \epsilon(\iota(\texttt{rdfs : subClassOf}))$, then $(x, c') \in \epsilon(\iota(\texttt{rdf : type}))$.

Computing RDF and RDFS entailment. When we have to compute whether $F \vdash_{rdf} Q$ (or $F \vdash_{rdfs} Q$), we will add to F all necessary information to answer Q: first the axiomatic triples (at least a finite subset of them), then enrich it with all information that will force its simple model to be an RDF or RDFS interpretation. This can be done with the semantic rules of [8] that are used to generate a graph that respects all semantic conditions. A CG translation of one of these rules is presented in Fig. 7.

If G is an RDF or RDF* graph, then its *saturation* $S_k^{rdf}(G)$ (or $S_k^{rdfs}(G)$) is obtained from G as follows:

1. make the union of G and $\mathcal{A}_k^{rdf}$ (or $\mathcal{A}_k^{rdfs}$);
2. enrich the obtained graph with RDF or RDFS rules until a fixpoint is obtained.

The two previous conditions translating RDF semantics can be written as the following rules:

1. for each triple of form (s, p, o), add the triple $(p, \texttt{rdf:type}, \texttt{rdf:Property})$;
2. for each well-typed literal l of G whose type is `rdf:XMLLiteral`, add the triple $(l, \texttt{rdf:type}, \texttt{rdf:XMLLiteral})$.

RDFS semantic conditions can be translated in the same way (that is indeed done in [8]), and our two example semantic conditions can now be expressed as rules:

1. if there is a triple $(p, \texttt{rdfs:domain}, c)$ and a triple (s, p, o) in the graph, then add the triple $(s, \texttt{rdf:type}, c)$.
2. if there is a triple $(x, \texttt{rdf:type}, c)$ and a triple $(c, \texttt{rdfs:subClassOf}, c')$ in the graph, then add the triple $(x, \texttt{rdf:type}, c')$.

Property 1 (Satisfiability). An RDF or RDF* graph G is *RDF-satisfiable* (resp. *RDFS-satisfiable*) iff $S_0^{rdf}(G)$ (resp. $S_0^{rdfs}(G)$) does not contain any triple of form $(l, \texttt{rdf:type}, \texttt{rdf:XMLLiteral})$, where l is an ill-typed literal whose type is `rdf:XMLLiteral`.

Theorem 3 (RDF(S) Entailment Connection). *Let F and Q be RDF or RDF* graphs. Then $F \vdash_{rdf} Q$ (resp. $F \vdash_{rdfs} Q$) iff either F is not satisfiable or $S_k^{rdf}(F) \vdash_s Q$ (resp. $S_k^{rdfs}(F) \vdash_s Q$) where $k \geq 1$ is the greater number such that* `rdf:_k` *appears in F or Q.*

4 The RDF/BG Transformations

It was pointed out [4] that RDF and CGs share very similar characteristics. Fig. 5 summarizes the main points of the "natural" correspondence between RDF(S) and BGs, along with their logical translation. However, such an intuitive translation does not satisfy our main evaluation criterion, which is the equivalence between reasonings in the two formalisms.

RDFS Triple	*Equivalent BG*	*Logical Translation*
C rdf:type rdfs:Class	C concept type	C unary predicate
R rdf:type rdf:Property	R binary relation type	R binary predicate
C rdfs:subClassOf D	$C \leq D$	$\forall x(C(x) \rightarrow D(x))$
R rdfs:subPropertyOf S	$R \leq S$	$\forall x \forall y(R(x,y) \rightarrow S(x,y))$
R rdfs:domain C	$\sigma(R) = (C, -)$	$\forall x \forall y(R(x,y) \rightarrow C(x))$
R rdfs:range D	$\sigma(R) = (-, D)$	$\forall x \forall y(R(x,y) \rightarrow D(y))$

Fig. 5. Correspondences between RDFS, BG and logic

4.1 Problems with the Intuitive Translation

Assume we want to encode the simple entailment in RDF or RDF$*$ within the basic CG fragment. Let us note $\mathcal{T}_{basic}$ this transformation. An RDF graph G is encoded into a BG $\mathcal{T}_{basic}(G)$ in normal form as follows. For each term t that appears either as subject or object in a triple of G, we create a concept node whose type is $\top$ and whose marker is a named generic marker $*t$ if t is a blank, the individual marker t if t is a uriref and the individual marker *val*(t) if t is a literal. Then we merge literals that have the same value. Finally, for every triple $(s, p, o) \in G$, we add a relation node whose label is p if p is a uriref and $\top_2$ (with $\top_2$ being the maximal relation type for binary relations) if p is a blank, and whose arguments are respectively the node associated with s and the node associated with o. The case where p is a blank can happen only in RDF*.

The next theorem expresses that, even if RDF does not distinguish between entities and relations, that does not prevent the BG homomorphism to be complete w.r.t. RDF simple entailment.

Theorem 4. *Let F and Q be RDF graphs. Then $F \vdash_s Q$ iff $\mathcal{T}_{basic}(F) \vdash \mathcal{T}_{basic}(Q)$.*

Things change when we consider the RDF*language. Consider for instance the following triples and the translation of each of them into a BG:

1. $t_Q = (a, \_x, \_x)$, where $\_x$ is a blank, translated into $Q = [\top : a]\text{->}\ (\top_2)\text{->}\ [\top : *x]$;
2. $t_1 = (a, b, c)$, translated into $F_1 = [\top : a]\text{->}\ (b)\text{->}\ [\top : c]$;
3. $t_2 = (a, b, b)$, translated into $F_2 = [\top : a]\text{->}\ (b)\text{->}\ [\top : b]$.

[4] `http://www.w3.org/DesignIssues/CG.html`

One has $t_2 \vdash_s t_Q$ but not $t_1 \vdash_s t_Q$ since models of t_1 that map b and c to distinct elements are not models of t_Q. However, there is a BG homomorphism from Q to both F_1 and F_2. The trouble is that the translation of t_Q into a BG does not keep the information that the object and the predicate of the triple have the same variable name.

We present here two solutions solving that problem: in the first, we change the structure of the built BG, while in the second we restrict the domain of the translation to a subset of RDF.

4.2 The RDF/BG "3-Hypergraphs" Transformation

The transformation $\mathcal{T}_3$ described in this section relies on the work of [2]. It is sound and complete w.r.t. RDF(S) semantics.

Let G be an RDF or RDF* graph. $\mathcal{T}_3$ encodes G in a BG as follows. First, it assigns a concept node to every term appearing in G, with its marker being the same as in $\mathcal{T}_{basic}$; for now, we consider that its type is $\top$. Then, for every triple (s, p, o) in G, it adds a relation node typed by `triple`, whose arguments are respectively the concept nodes assigned to s, p, and o.

It is immediate to check that an interpretation I is a simple model of an RDF or RDF* graph G if and only if this interpretation is a model (in the CG sense) of the BG $\mathcal{T}_3(G)$. This transformation indeed encodes exactly the semantics of the language RDF. It follows that:

Theorem 5. *Let F and Q be RDF or RDF* graphs. Then $F \vdash_s Q$ iff $\mathcal{T}_3(F) \vdash \mathcal{T}_3(Q)$.*

Now, let us enhance this transformation by adding relevant types to the concept nodes, thus translating the RDF vocabulary. The added hierarchy of concept types is depicted in Fig. 6 (it is indeed the same for all RDF or RDF* input graphs). Some semantic rules require syntactic information in their hypothesis. This has to be taken into account in our syntactic transformations, thus, nodes translating an RDF term are typed according to the more specific syntactic category(ies).

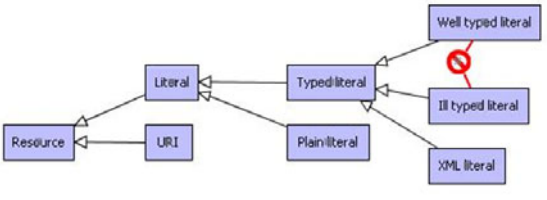


Fig. 6. T_C depiction of RDF transformation $\mathcal{T}_3$

To obtain the semantic completeness w.r.t. $\vdash_{rdf}$ and $\vdash_{rdfs}$, we add the translation of RDF(S) axiomatic triples as new facts, as well as the translation of the semantic rules of RDF(S) as rules. For instance the RDFS "domain rule" presented previously can be translated into the BG rule pictured in Fig. 7. Note that such a rule relies upon generic concept nodes associated with a property, and thus uses the RDF* language.

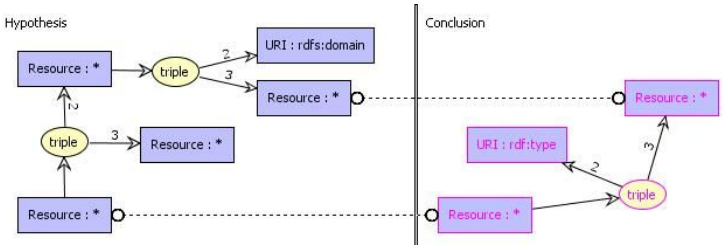


Fig. 7. BG+ depiction of an RDFS rule

Let us note $\mathcal{R}_{rdf}$ (resp. $\mathcal{R}_{rdfs}$) the set of BG rules associated with RDF (resp. RDFS) semantic rules. We can now express the equivalence between the BG and RDF(S) fragments based on transformation $\mathcal{T}_3$.

Theorem 6. *Let F and Q be RDF or RDF* graphs. Let $F_{rdf} = F \cup \mathcal{A}_k^{rdf}$, where $k \geq 1$ is the greatest integer such that* `rdf:_k` *appears in F or Q. Then $F \vdash_{rdf} Q$ iff one of the following conditions is satisfied:*

- $\mathcal{T}_3(F_{rdf}), \mathcal{R}_{rdf} \vdash \mathcal{T}_3(\{(x, \texttt{rdf:type}, \texttt{rdf:XMLLiteral})\})$, *where x is an ill-typed XMLLiteral typed literal.*
- $\mathcal{T}_3(F_{rdf}), \mathcal{R}_{rdf} \vdash \mathcal{T}_3(Q)$.

The same property, obtained by substituting $_{rdf}$ with $_{rdfs}$, holds for RDFS entailment.

The transformation $\mathcal{T}_3$ thus fulfills our main requirement: preserving the notion of entailment between RDF(S) and BGs. However, this transformation has severe drawbacks from a user perspective. First, the triples are harder to read than the binary relation they encode, and this default is made worse when saturating the graph by the application of rules. The second drawback is that $\mathcal{T}_3$, faithful to RDF, does not offer knowledge structuring. This was already one of the main criticisms addressed to semantic networks, and CGs answered that by establishing a clear distinction between factual and ontological knowledge. These are the drawbacks addressed in the next transformation.

4.3 The RDF/BG Intuitive Transformation

The second transformation, called $\mathcal{T}_{nat}$ and outlined in Fig. 5, has several qualities:

- It is natural (hence the notation $\mathcal{T}_{nat}$), in sense that it respects the kinds of knowledge: it translates classes into concept types (both represent sets of entities), properties into binary relations, and instance into instances.
- It preserves the visual qualities of the graph.
- It allows for a clear distinction between ontological and factual knowledge.

Due to the latter quality, $\mathcal{T}_{nat}$ cannot not translate RDF(S) entirely. However, we claim that it allows to translate exactly the subset of RDFS used for representing knowledge with the purpose of querying factual knowledge, i.e. typical semantic annotations. We

will denote this fragment of RDF(S) corresponding to BGs by RDFS-. Depending on the way we translate the signatures of properties, we will also obtain some rules that do not increase the complexity of deduction.

We define a transformation from RDFS- to BG(+) and a transformation from BG(+) to RDFS- in such way that these transformations are reciprocal one with respect to the other. Amongst the triples allowed in RDFS- we distinguish between *ontological* triples (corresponding to the vocabulary), *factual* triples (corresponding to the facts) and also *commentary* triples (corresponding to the elements not belonging to the CG formalism but present in CoGXML, the XML file format of CoGUI). The completeness result obtained states that, as long as the document to be entailed is composed only of RDF triples (as opposed to RDFS triples), then BG deduction is complete w.r.t. RDF(S) entailment. This means that this transformation is well-suited to the deduction of factual knowledge but not to ontological knowledge.

The RDFS- Fragment. In the notations below, *B* stands for *Blanks*, *L* for *Literals* and *SU* for *simple URIs*, that is URIs not starting by `rdf:` or `rdfs:`. We further refine *SU* into *SUc* for the *SU* belonging to classes, *SUp* for properties and *SUi* for instances. The following triple patterns are allowed in RDFS-:

- *Ontological* triples: (*SUc*, `rdf:type`, `rdfs:Class`), (*SUc*, `rdfs:subClassOf`, *SUc*), (*SUp*, `rdf:type`, `rdf:Property`), (*SUp*, `rdfs:subPropertyOf`, *SUp*), (*SUp*, `rdfs:domain`, *SUc*) and (*SUp*, `rdfs:range`, *SUc*);
- *factual* triples: (*SUi*, `rdf:type`, *SUc*), (*B*, `rdf:type`, *SUc*) and (*s*, *SUp*, *o*), where *s* is either *B* or *SUi*, and *o* is either *B*, *SUi* or *L*;
- *commentary* triples: (*SU*, `rdfs:label`, *L*) and (*SU*, `rdfs : comment`, *L*).

We do not allow for anonymous classes or properties: for this reason, we forbid ontological triples containing a blank either as a subject or as an object, as well as factual triples containing a blank as a property, or as an object when the predicate is rdf:type. Finally, the *separability* condition has to be fulfilled: a given *SU* can appear only in one of the categories "class" (*SUc*), "property" (*SUp*) and "instance" (*SUi*). In terms of CGs, this condition states that the sets T_C, T_R and I are pairwise disjoint.

Transformation $\mathcal{T}_{nat}$: RDFS- towards BG(+). In the description below, an element is added to the vocabulary or the fact graph only if it does not already exist. Any addition of a relation (obviously binary) is done by default with the signature (*rdfs:Resource*, *rdfs:Resource*). Further domain and range statements will induce a specialization of this signature. *rdfs:Resource* behaves as the top of the hierarchy. This specialization can be performed in two different ways: either by specializing the signature in the vocabulary directly, or by introducing a rule translating this specialization.

The transformation takes place as follows:

1. Creation of the concept types *rdfs:Resource*, *rdfs:Literal* and *rdfs:Datatype*, as well as the relation type *rdf:Property(rdfs:Resource, rdfs:Resource)*
2. Treatment of ontological triples:
   - (*SUc*, `rdf:type`, `rdfs:Class`) $\rightarrow$ addition of the concept type *SUc*
   - (*SUc1*, `rdfs:subClassOf`, *SUc2*) $\rightarrow$ addition of concept types *SUc1* and *SUc2*, where *SUc1* $\leq$ *SUc2*

- (*SUp*, `rdf:type`, `rdf:Property`) → addition of the relation type *SUp*
- (*SUp1*, `rdfs:subPropertyOf`, *SUp2*) → addition of binary relations *SUp1* and *SUp2* with $SUp1 \leq SUp2$
- (*SUp*, `rdfs:domain`, *SUc*) → addition of the binary relation type *SUp*, of the concept type *SUc*, and treatment of the domain information (as explained above)
- (*SUp*, `rdfs:range`, *SUc*) → similar to above

3. Treatment of factual triples (f.i. $\mathcal{T}_{nat}$ applied to the triples in Fig. 4 yields the BG in Fig. 2)
   - (*SUi*, `rdf:type`, *SUc*) → addition of the concept node *[SUc : SUi]*, addition to the vocabulary of the individual marker *SUi* and of the concept type *SUc*
   - (*B*, `rdf:type`, *SUc*) → similar to above with the only difference of the generic marker, i.e. the node *[SUc :*B]* is obtained
   - triples of form (*s*, *SUp*, *o*) → addition of the corresponding concept and relation nodes, along with the type insertions into the vocabulary
4. The commentary triples, i.e. containing `rdfs:comment` or `rdfs:label` are translated in labels and commentaries in CoGXML

When a document is violating the separability condition, there are several possibilities for dealing with the triples that are causing this violation. A drastic solution consists in rejecting the RDF(S) document. Another solution consists in only accepting a subset of the RDF(S) document that satisfies the separability condition: in this case, the choices made have to be independent of the order in which the triples have been analyzed, so that two RDFS documents with the same set of triples, thus semantically equivalent, are translated in the same way. Currently, the implemented solution consists in rejecting the RDF(S) documents violating the separability between the concept and relation type hierarchies. For the separation of individual markers with the concept / relation hierarchy, the priority is given to declarations concerning classes and properties.

Transformation $\mathcal{T}_{nat^-}$: BG towards RDFS- We assume that all the relations are binary. If not, we can first "binarize" the BGs. Binary BGs can be easily translated in RDFS-:

1. The vocabulary is translated into ontological triples:
   - For all concept types $t \rightarrow$ (t, `rdf:type`, `rdfs:Class`)
   - For all concept types t_1 and t_2 s. t. $t_2 \leq t_1 \rightarrow$ (t_2, `rdfs:subClassOf`, t_1)
   - For all relations $r \rightarrow$ (r, `rdf:type`, `rdf:Property`)
   - For all signatures (t_1,t_2) of a relation $r \rightarrow$ (r, `rdfs:domain`, t_1), (r, `rdfs:range`, t_2)
   - For all relations r_1 and r_2 s. t. $r_2 \leq r_1 \rightarrow$ (r_2, `rdfs:subPropertyOf`, r_1)
2. The fact graphs are translated into factual triples (f.i. $\mathcal{T}_{nat^-}$ applied to the BG in Fig. 2 yields the triples in Fig. 4):
   - We assign a different blank to each generic concept node. The term assigned to a generic concept node is the above mentioned blank and the one assigned to an individual concept node is the URI corresponding to its individual marker;

- For all concept nodes of type t_1,..., t_n and associated term *e*, we have: for *i* from 1 to *n* $\rightarrow$ (*e*, `rdf:type`, t_i)
- For all relation nodes *r* having as the first neighbor c_1 and second neighbor c_2 with the associated terms e_1 respectively e_2 $\rightarrow$ (e_1, *r*, e_2)

3. The commentaries and labels associated to concept and relation types are translated by commentary triples $\rightarrow$ (*t*, `rdfs:comment`, *literal*), (*t*, `rdfs:label`, *literal*).

Apart from n-ary relation types, the only element that we cannot translate into RDFS is the notion of forbidden conjunctive type (expressing that two concept types are disjoint). Note it can be translated by the OWL predicate owl:disjointWith (see Sect. 5).

Note that the rules associated with signatures could be translated into RDFS- (which is not implemented yet in the transformation provided by CoGUI).

Properties of $\mathcal{T}_{nat}$ and $\mathcal{T}_{nat^-}$. These transformations are "essentially" reciprocal, in the following sense: their composition is the identity, up to a fixed set of axiomatic knowledge, which is made explicit by the transformation, but has no incidence on the semantics of the transformed knowledge. More precisely:

Property 2. Let K be an RDF graph (resp. a vocabulary and a BG F). Let f be $\mathcal{T}_{nat^-} \circ \mathcal{T}_{nat}$ (resp. $\mathcal{T}_{nat} \circ \mathcal{T}_{nat^-}$). Then $K' = f(K) = K \cup A$, where A is a fixed set of triples (resp. $F \cup A$, where A is a fixed part of the vocabulary), and $f(K') = K'$.

The following result specifies the kind of completeness obtained:

Theorem 7
Let G_1 and G_2 be RDFS- graphs such that G_2 contains solely factual triples.
Then $G_1 \vdash_{rdfs} G_2$ iff $\mathcal{T}_{nat}(G_1) \vdash \mathcal{T}_{nat}(G_2)$.
Let F and Q be BGs on a vocabulary $\mathcal{V}$.
Then $\mathcal{V}, F \vdash Q$ iff $\mathcal{T}_{nat^-}(\mathcal{V}) \cup \mathcal{T}_{nat^-}(F) \vdash_{rdfs} \mathcal{T}_{nat}(Q)$.

5 Perspectives

Let us emphasize the interest of the second transformation, i.e. $\mathcal{T}_{nat}$. This transformation is in line with the knowledge representation vision of the Semantic Web, in the sense that it clearly distinguishes between different kinds of knowledge. Moreover, CoGUI allows to visualize the knowledge base obtained according to this separation. It defines a fragment of RDF(S) that can be provided with a semantics in classical first-order logics, which is compatible with most description logics, in particular the OWL-DL fragment. Hence, it is potentially extensible to take a richer ontology into account, that would be represented by description logics.

As a matter of fact, many RDFS files use simple OWL features. The combination of RDFS and OWL-DL in documents leads to a combinatorial explosion of the querying problem. Recently, restrictions of OWL-DL have been proposed to overcome this explosion (see OWL2 [5]). We are currently studying the precise relationships between these restrictions and fragments of CGs in the context of query answering. As a preliminary and pragmatic work, we have extended the $\mathcal{T}_{nat}$ translation to some OWL statements which can be expressed in BG+, thus without increasing complexity of reasoning.

[5] `http://www.w3.org/TR/owl2-profiles/`

References

1. Baget, J.-F.: Simple Conceptual Graphs Revisited: Hypergraphs and Conjunctive Types for Efficient Projection Algorithms. In: Ganter, B., de Moor, A., Lex, W. (eds.) ICCS 2003. LNCS, vol. 2746, pp. 229–242. Springer, Heidelberg (2003)
2. Baget, J.F.: RDF Entailment as a Graph Homomorphism. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 82–96. Springer, Heidelberg (2005)
3. Baget, J.-F., Mugnier, M.-L.: The Complexity of Rules and Constraints. JAIR 16, 425–465 (2002)
4. Chein, M., Mugnier, M.: Graph-based Knowledge Representation: Computational Foundations of Conceptual Graphs. Springer, Heidelberg (2009)
5. Chein, M., Mugnier, M.-L.: Conceptual Graphs: Fundamental Notions. Revue d'Intelligence Artificielle 6(4), 365–406 (1992)
6. Chein, M., Mugnier, M.-L.: Concept types and coreference in simple conceptual graphs. In: Wolff, K.E., Pfeiffer, H.D., Delugach, H.S. (eds.) ICCS 2004. LNCS (LNAI), vol. 3127, pp. 303–318. Springer, Heidelberg (2004)
7. Corby, O., Dieng, R., Hebert, C.: A Conceptual Graph Model for W3C RDF. In: Ganter, B., Mineau, G.W. (eds.) ICCS 2000. LNCS (LNAI), vol. 1867, pp. 172–192. Springer, Heidelberg (2000)
8. Hayes, P. (ed.): RDF Semantics. W3C Recommendation. W3C (2004)
9. Mugnier, M.-L.: Knowledge Representation and Reasoning based on Graph Homomorphism. In: Ganter, B., Mineau, G.W. (eds.) ICCS 2000. LNAI, vol. 1867, pp. 172–192. Springer, Heidelberg (2000)
10. Sowa, J.F.: Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley, Reading (1984)

Default Conceptual Graph Rules, Atomic Negation and Tic-Tac-Toe

Jean-François Baget[1,2] and Jérôme Fortin[3,2]

[1] INRIA Sophia Antipolis, 2004 Route des Lucioles 06902 Sophia Antipolis, France
baget@lirmm.fr
[2] LIRMM (CNRS & Université Montpellier II), F–34392 Montpellier Cedex 5, France
[3] IATE, UMR1208, F–34060 Montpellier Cedex 1, France
jerome.fortin@supagro.inra.fr

Abstract. In this paper, we explore the expressivity of default CG rules (a CG-oriented subset of Reiter's default logics) through two applications. In the first one, we show that default CG rules provide a unifying framework for CG rules as well as polarized CGs (CGs with atomic negation). This framework allows us to study decidable subclasses of a new language mixing CG rules with atomic negation. In the second application, we use default CG rules as a formalism to model a game, an application seldom explored by the CG community. This model puts into light the conciseness provided by defaults, as well as the possibilities they offer to achieve efficient reasonings.

1 Introduction

Default CG Rules have been introduced in [1] as a requirement for an agronomy application. These rules encode a subset of Reiter's default logics [2], with knowledge of form "if *hypothesis*, then *conclusion* is generally true, unless drawing that conclusion leads to contradict one of the *justifications* of the default", where the hypothesis, conclusion, and justifications are simple conceptual graphs. These default rules have two main interest. First they admit a natural graphical representation that extends the representation of CG rules, and thus inherits from the user-friendly charateristics of the CG formalism. Moreover, they form the corresponding fragment in Reiter's default logic of the rule fragment in FOL. We believe that the decidability arguments of the rule fragment [3,4] will find their counterpart in default rules. In this paper, we focus on two applications to study the expressiveness of that language.

The first uses default CG rules to encode *atomic negation* into conceptual graphs. Indeed, important extensions of Sowa's simple conceptual graphs [5] have concerned CG rules [6] and polarized graphs [7] (*i.e.* simple graphs enriched with atomic negation). There has been for now no unifying framework for these two extensions that relies upon graph-based reasonings to compute deduction (the work of [7], for example, that extends conceptual graphs to handle the whole first-order logics, mixes graph-based reasonings with a tableaux mechanism). We show here that default CG rules provide such a unifying framework, and put that framework to use to begin to explore decidable subclasses of a CG language that enriches CG rules with atomic negation.

M. Croitoru, S. Ferré, and D. Lukose (Eds.): ICCS 2010, LNAI 6208, pp. 42–55, 2010.

In the second application, we have chosen to present an example using default CG rules in a field seldom explored by the CG community: *games*. Our motivation was twofold: 1) we wanted a new type of application that forced us to think "out of the box", as was done with the ICCS Sisyphus-I initiative [8], and 2) we wanted a motivating example for Master's students in knowledge representation. This model has put into light two main interests of default CG rules: 1) they mix the intuitive graphical representation of CGs with the conciseness brought by Reiter's defaults, and 2) though default CG rules form a more complex language than rules, they offer mechanisms that allow for more efficient reasonings.

2 From Simple CGs to Default CGs

In this first section, we recall main notations and results required for the default CG rules used in this paper. In SECT. 2.1, we present the simple CGs of [5], in SECT. 2.2, the CG rules of [6], and finally in SECT. 2.3, the default CG rules of [1].

2.1 Simple Conceptual Graphs

Syntax With the simple CGs of [5], a knowledge base is structured into two objects: the vocabulary (also called support) encodes hierarchies of types, and the conceptual graphs (CGs) themselves represent entities and relations between them. Our simple CGs use *named generic markers*, and are extended to handle *conjunctive types*, as done in [9].

Definition 1 (Vocabulary). *We call* vocabulary *a tuple* $\mathcal{V} = (\mathcal{C}, \mathcal{R} = (\mathcal{R}_1, \ldots, \mathcal{R}_k), \mathcal{M}_I, \mathcal{M}_G)$ *where* $\mathcal{C}$ *is a partially ordered set of* concept types *that contains a greatest element* $\top$*, each* $\mathcal{R}_i$ *is a partially ordered set of* relation types of arity i*,* $\mathcal{M}_I$ *is a set of* individual markers*, and* $\mathcal{M}_G$ *is a set of* generic markers*. Note that all these sets are pairwise disjoint, and that we denote all the partial orders by* $\leq$*.*

Definition 2 (Conjunctive types). *A* conjunctive concept type *over a vocabulary* $\mathcal{V}$ *is a set* $T = \{t_1, \ldots, t_p\}$ *(that we can note* $T = t_1 \sqcap \ldots \sqcap t_p$*) of concept types of arity* k*. If* $T = \{t_1, \ldots, t_p\}$ *and* $T' = \{t'_1, \ldots, t'_q\}$ *are two conjunctive concept types, then we also note* $T \leq T' \Leftrightarrow \forall t'_i \in T', \exists t_j \in T$ *such that* $t_j \leq t'_i$*.*

Definition 3 (Simple CGs). *A* simple CG *is a tuple* $G = (C, R, \gamma, \lambda)$ *where* C *and* R *are two finite disjoint sets (*concept nodes *and* relations*) and* γ *and* λ *two mappings:*

- $\gamma : R \to C^+$ *associates to each relation a tuple of concept nodes* $\gamma(r) = (c_1, \ldots, c_k)$ *called the* arguments *of* r*,* $\gamma_i(r) = c_i$ *is its* i*th argument and* degree$(r) = k$*.*
- λ *maps each concept node and each relation to its* label*. If* $c \in C$ *is a concept node, then* $\lambda(c) =$ (type(c), marker(c)) *where* type(c) *is a conjunctive concept type and* marker(c) *is either an individual marker of* $\mathcal{M}_I$ *or a generic marker of* $\mathcal{M}_G$*. If* $r \in R$ *is a relation and* degree$(r) = k$*, then* $\lambda(r)$ *is a relation type of arity* k*.*

A simple CG is said to be normal *if all its concept nodes have different markers. Any simple CG* G *can be put into its equivalent* normal form nf(G) *in linear time.*

Semantics. We associate a first order logics (FOL) formula $\Phi(\mathcal{V})$ to a vocabulary $\mathcal{V}$, and a FOL formula $\Phi(G)$ to a simple CG G. These formulae are obtained as follows:
Interpretation of a vocabulary Let $\mathcal{V} = (\mathcal{C}, \mathcal{R} = (\mathcal{R}_1, \ldots, \mathcal{R}_k), \mathcal{M}_I, \mathcal{M}_G)$ be a vocabulary. We can consider each concept type of $\mathcal{C}$ as a unary predicate name, each relation type of $\mathcal{R}_i$ as a predicate name of arity i, each individual marker of $\mathcal{M}_I$ as a constant, and each generic marker of $\mathcal{M}_G$ as a variable. For each pair (t, t') of concept types of $\mathcal{C}$ such that $t \leq t'$, we have a formula $\phi((t, t')) = \forall x(t(x) \rightarrow t'(x))$. For each pair (t, t') of relation types of arity i such that $t \leq t'$, we have a formula $\phi((t, t')) = \forall x_1 \ldots \forall x_i(t(x_1, \ldots, x_i) \rightarrow t'(x_1, \ldots, x_i))$. Then the FOL interpretation $\Phi(\mathcal{V})$ of $\mathcal{V}$ is the conjunction of all $\phi((t, t'))$, for every pair (t, t') such that $t' < t'$.
Interpretation of a simple CG Let $G = (C, R, \gamma, \lambda)$ be a simple CG. We can associate a formula to each concept node and relation of G: if $c \in C$ and $\mathit{type}(c) = t_1 \sqcap \ldots \sqcap t_k$, then $\phi(c) = t_1(\mathit{marker}(c)) \wedge \ldots \wedge t_k(\mathit{marker}(c))$; and if $r \in R$, with $\gamma(r) = (c_1, \ldots, c_q)$ and $\lambda(r) = t$, then $\phi(r) = t(\mathit{marker}(c_1), \ldots, \mathit{marker}(c_q))$. We note $\phi(G) = \bigwedge_{c \in C} \phi(c) \wedge \bigwedge_{r \in R} \phi(r)$. The FOL formula $\Phi(G)$ associated with a simple CG is the existential closure of the formula $\phi(G)$.

Computing Deduction. Computing a graph homomorphism (known as projection in CGs) is a sound and complete algorithm for deduction of the associated FOL formulae. HOMOMORPHISM is an NP-complete problem, that becomes polynomial when the question graph is a tree (see [9,10] for more polynomial subclasses).

Definition 4 (Homomorphism). *Let $F = (C_F, R_F, \gamma_F, \lambda_F)$ and $Q = (C_Q, R_Q, \gamma_Q, \lambda_Q)$ be two simple CGs defined over a vocabulary $\mathcal{V}$. A* homomorphism *from Q to F is a mapping $\pi : C_Q \rightarrow C_F$ such that:*

- *if $c \in C_Q$ is individual, then $\mathit{marker}(c) = \mathit{marker}(\pi(c))$;*
- *if c and c' are two generic concept nodes with same markers, then $\pi(c) = \pi(c')$;*
- *$\forall c \in C_Q$, $\mathit{type}(\pi(c)) \leq \mathit{type}(c)$;*
- *$\forall r \in R_Q$, $\exists r' \in R_F$ such that $\lambda(r') \leq \lambda(r)$ and $\gamma(r') = \pi(\gamma(r))$.*

Theorem 1. *Let F and Q be two simple CGs defined over a vocabulary $\mathcal{V}$. Then $\Phi(\mathcal{V}), \Phi(F) \vdash \Phi(Q)$ iff there exists a homomorphism from Q to $\mathit{nf}(F)$.*

2.2 Conceptual Graph Rules

CG rules form an extension of CGs with knowledge of form "if *hypothesis* then *conclusion*". Introduced in [11], they have been further formalized and studied in [6,3].

Syntax. A usual way to define CG rules is to establish *co-reference relations* between the hypothesis and the conclusion. We rely here upon *named generic markers*: generic nodes with same marker represent the same entity.

Definition 5 (CG rule). *A* conceptual graph rule*, defined on a vocabulary $\mathcal{V}$, is a pair $R = (H, C)$ where H and C are two simple CGs, respectively called the hypothesis and the conclusion of the rule.*

Semantics. We present here the usual Φ semantics of a CG rule, and introduce an equivalent semantics Φ^f using function symbols. Basically, Φ^f translates in a straightforward way the skolemisation of existentially quantified variables. This equivalent semantics makes for an easier definition of default rules semantics: since default rules are composed of different formulas, we cannot rely upon the quantifier's scope to link variables, and thus have to link them through functional terms.

Let $R = (H, C)$ be a CG rule. Then the FOL interpretation of R is the formula $\Phi(R) = \forall x_1 \dots \forall x_k(\phi(H) \rightarrow (\exists y_1 \dots \exists y_q \phi(C)))$, where $x_1, \dots, x_k$ are all the variables appearing in $\phi(H)$ and $y_1, \dots, y_q$ are all the variables appearing in $\phi(C)$ but not in $\phi(H)$. If $\mathcal{R}$ is a set of CG rules, then $\Phi(R) = \bigwedge_{R \in \mathcal{R}} \Phi(R)$.

As an alternate semantics, let G be a simple CG and X be a set of nodes. We denote by $F = \{f_1, \dots, f_p\}$ the set of variables associated with generic markers that appear both in G and in X. The formula $\phi^f_X(G)$ is obtained from the formula $\phi(G)$ by replacing each variable y appearing in $\phi(G)$ but not in F by a functional term $f^y_G(f_1, \dots, f_p)$. Then the FOL interpretation (with function symbols) of a rule $R = (H, C)$ is the formula $\Phi^f(R) = \forall x_1 \dots \forall x_k(\phi(H) \rightarrow \phi^f_X(C))$ where X is the set of nodes appearing in H. If $\mathcal{R}$ is a set of CG rules, then $\Phi^f(\mathcal{R}) = \bigwedge_{R \in \mathcal{R}} \Phi^f(R)$.

The translations of the rule $R = (H, C)$ where $H = [Human : *x]$ and $C = [Human : *x] < -(isParent) < -[Human : *y]$ (in linear form, meaning that every human has a human parent) are:
$\Phi(R) = \forall x, (Human(x) \rightarrow \exists y(Human(y) \wedge isParent(y, x)))$
$\Phi(R)^f = \forall x, (Human(x) \rightarrow Human(f^y_C(x)) \wedge isParent(f^y_C(x), x))$

Computing Deduction. We present here the forward chaining mechanism used to compute deduction with CG rules. In general, this is an undecidable problem. The reader can refer to [3] for an up-to-date cartography of decidable subclasses of the problem.

Definition 6 (Application of a rule). *Let G be a simple CG, $R = (H, C)$ be a rule, and π be a homomorphism from H to G. The* application of R on G according to π *produces a normal simple CG $\alpha(G, R, \pi) = \text{nf}(G \oplus C_\pi)$ where:*

- *C_π is a simple CG obtained as follows from a copy of C: (i) associate to each generic marker x that appears in C but not in H a new distinct generic marker $\sigma(x)$; (ii) for every generic concept node c of C whose marker x does not appear in H, replace* marker(c) *with $\sigma(x)$; and (iii) for every generic concept node c of C, if* marker(c) *also appears in H, then replace* marker(c) *with* marker$(\pi(c))$.
- *the operator $\oplus$ generates the* disjoint union *of two simple CGs G and G': it is the simple CG whose drawing is the juxtaposition of the drawings of G and G'.*

Theorem 2. *Let G and Q be two simple CGs, and $\mathcal{R}$ be a set of CG rules, all defined on a vocabulary $\mathcal{V}$. Then the following assertions are equivalent:*

- $\Phi(\mathcal{V}), \Phi(G), \Phi(\mathcal{R}) \vdash \Phi(Q)$
- $\Phi(\mathcal{V}), \Phi(G), \Phi^f(\mathcal{R}) \vdash \Phi(Q)$
- *there exists a sequence $G_0 = \text{nf}(G), G_1, \dots, G_n$ of simple CGs such that: (i) $\forall 1 \leq i \leq n$, there is a rule $R = (H, C) \in \mathcal{R}$ and a homomorphism π of H to G_{i-1} such that $G_i = \alpha(G_{i-1}, R, \pi)$; and (ii) there is a homomorphism from Q to G_n.*

Note that the forward chaining algorithm that relies upon the above characterization is ensured to stop when the set of rules involved is *range restricted*, *i.e.* their logical semantics Φ does not contain any existentially quantified variable in the conclusion.

The "functional semantics" can provide us with an alternate rule application mechanism α^f. Let us begin by "*freezing*" the graph G, *e.g.* by replacing each occurrence of a generic marker by a distinct individual marker. Then, when applying a rule R on G (or a graph derived from G) according to a projection π, consider the formula $\Phi^f(R)$ associated with R. Should the application of $R = (H, C)$ produce a new generic node c from the copy of a generic node having marker y, consider the functional term $f^y_C(x_1, \ldots, x_k)$ associated to the variable y. Then the marker of c becomes $f^y_C(\pi(x_1), \ldots, \pi(x_k))$. Thanks to the previous theorem, α^f makes for an equivalent forward chaining mechanism, that has an added interest. It allows to have a "functional constant" identifying every concept node generated in the derivation. This feature will be used to explain default rules reasonings in an easier way than in [1].

2.3 Default CG Rules

A brief introduction. Let us recall some basic definitions of Reiter's default logics. For a more precise description and examples, the reader should refer to [12,2].

Definition 7 (Reiter's default logic). *A Reiter's* default theory *is a pair* (Δ, W) *where* W *is a set of FOL formulae and* Δ *is a set of* defaults *of form* $\delta = \frac{\alpha(\overrightarrow{x}):\beta_1(\overrightarrow{x}),\cdots,\beta_n(\overrightarrow{x})}{\gamma(\overrightarrow{x})}$, $n \geq 0$, *where* $\overrightarrow{x} = (x_1, \cdots, x_k)$ *is a tuple of variables,* $\alpha(\overrightarrow{x})$, $\beta_i(\overrightarrow{x})$ *and* $\gamma(\overrightarrow{x})$ *are FOL formulae for which each free variable is in* $\overrightarrow{x}$.

The intuitive meaning of a default δ is "For all individuals $(x_1, \cdots, x_k)$, if $\alpha(\overrightarrow{x})$ is believed and each of $\beta_1(\overrightarrow{x}), \cdots, \beta_n(\overrightarrow{x})$ can be consistently believed, then one is allowed to believe $\gamma(\overrightarrow{x})$". $\alpha(\overrightarrow{x})$ is called the *prerequisite*, $\beta_i(\overrightarrow{x})$ are called the *justifications* and $\gamma(\overrightarrow{x})$ is called the *consequent*. A default is said *closed* if $\alpha(\overrightarrow{x})$, $\beta_i(\overrightarrow{x})$ and $\gamma(\overrightarrow{x})$ are all closed FOL formulae.

Intuitively, an *extension* of a default theory (Δ, W) is a set of formulae that can be obtained from (Δ, W) while being consistently believed. More formally, an extension E of (Δ, W) is a minimal deductively closed set of formulae containing W such that for any $\frac{\alpha:\beta}{\gamma} \in \Delta$, if $a \in E$ and $\neg\beta \notin E$, then $\gamma \in E$. The following theorem provides an equivalent characterization of extensions that we use here as a formal definition.

Theorem 3 (Extension). *Let* (Δ, W) *be a closed default theory and* E *be a set of closed FOL formulae. We inductively define* $E_0 = W$ *and for all* $i \geq 0$, $E_{i+1} = Th(E_i) \cup \{\gamma \mid \frac{\alpha:\beta_1\cdots,\beta_n}{\gamma} \in \Delta, \alpha \in E_i \text{ and } \neg\beta_1, \cdots, \neg\beta_n \notin E\}$, *where* $Th(E_i)$ *is the deductive closure of* E_i. *Then* E *is an extension of* (Δ, W) *iff* $E = \cup_{i=0}^{\infty} E_i$.

Note that this characterization is not effective for computational purposes since both E_i and $E = \cup_{i=0}^{\infty} E_i$ are required for computing E_{i+1}.

Some problems that are to be addressed in Reiter's default logics are the following:
SKEPTICAL DEDUCTION: Given a default theory (Δ, W) and a formula Q, does Q belong to all extensions of (Δ, W)? In this case we note $(\Delta, W) \vdash_S Q$.
CREDULOUS DEDUCTION: Given a default theory (Δ, W) and a formula Q, does Q belong to an extension of (Δ, W)? In this case we note $(\Delta, W) \vdash_C Q$?

Definition 8 (Default CGs (Syntax)). *A* default CG*, defined on a vocabulary $\mathcal{V}$, is a tuple $D = (H, C, J_1, \ldots, J_k)$ where H, C, $J_1, \ldots$, and J_k are simple CGs respectively called the hypothesis, conclusion, and justifications of the default.*

Semantics. The semantics of a default CG $D = (H, C, J_1, \ldots, J_k)$ is expressed by a closed default $\Delta(D)$ in Reiter's default logics.

$$\Delta(D) = \frac{\phi(H) \ : \ \phi^f_X(C), \neg\phi^f_{X\cup Y}(J_1), \ldots, \neg\phi^f_{X\cup Y}(J_k)}{\phi^f_X(C)}$$

where X is the set of nodes of the hypothesis H and Y is the set of nodes of the conclusion C. If $D = (H, C, J_1, \ldots, J_k)$ is a default, we note *std*$(D) = (H, C)$ its *standard part*, which is a CG rule.

Computing Deduction. Our alternate derivation mechanism α^f makes for an easier description of the sound and complete reasoning mechanism of [1]. Let G and Q be two simple CGs, $\mathcal{R}$ be a set of CG rules, and $\mathcal{D}$ be a set of default CG rules, all defined over a vocabulary V. A node of the default derivation tree DDT$(\mathcal{K})$ of the knowledge base $\mathcal{K} = ((V, G, \mathcal{R}), \mathcal{D})$ is always labelled by a simple CG called fact and a set of simple CGs called constraints. A node of DDT$(\mathcal{K})$ is said *valid* if there is no homomorphism of one of its constraints or the constraints labelling one of its ancestors into its fact. Let us now inductively define the tree DDT$(\mathcal{K})$:

- its root is a node whose fact is G and whose constraint set is empty;
- if x is a valid node of DDT$(\mathcal{K})$ labelled by a fact F and constraints $\mathcal{C}$, then for every rule D in $\mathcal{D}$, for every homomorphism π of the hypothesis of D into a simple CG F' $\mathcal{R}$-derived from F, x admits a successor whose fact is the fact $\alpha^f(F', \textit{std}(D), \pi)$, and whose constraints are the $\pi(J_i)$ iff that successor is valid.

Theorem 4. *Let G and Q be two simple CGs, $\mathcal{R}$ be a set of CG rules, and $\mathcal{D}$ be a set of default CG rules, all defined over a vocabulary V. Then $\Phi(Q)$ belongs to an extension of the Reiter's default theory $(\{\Phi(V), \Phi(G), \Phi(\mathcal{R})\}, \Delta(\mathcal{D}))$ iff there exists a node x of DDT$((V, G, \mathcal{R}), \mathcal{D})$ labelled by a fact F such that $\Phi(V), \Phi(F), \Phi(\mathcal{R}) \vdash \Phi(Q)$.*

Intuitively, this result [1] states that the leaves of *DDT*$(\mathcal{K})$ encode extensions of the default. What is interesting in this characterization is that: 1) though our default CGs are not normal defaults in Reiter's sense, they share the same important property: every default theory admits an extension; and 2) if an answer to a query is found in any node of the default derivation tree, the same answer will still be found in any of its successors.

3 Using Default CG Rules for Atomic Negation

Neither simple CGs nor CG rules can handle negation, even in its basic atomic form. Indeed, their reasonings do not support branching, necessary in tableaux-like mechanisms as soon as negation or disjunction is involved. We show that default CG rules can handle as well the semantics of polarized graphs (simple CGs enriched with atomic negation [13]) as the semantics of their extension to polarized graphs rules.

3.1 Polarized Graphs

Simply put, polarized graphs [13] form an extension of simple CGs in which all types are polarized: a type with a positive polarization is translated into a positive atom in FOL, while a type with a negative polarization is translated by a negated atom.

Definition 9 (Signed types). *Let $\mathcal{V}$ be a vocabulary. A* signed concept (resp. relation) type *on $\mathcal{V}$ is a pair of form $(+,t)$ or $(-,t)$ where t is a concept type of $\mathcal{V}$ (resp. a relation type of $\mathcal{V}$). $(-,\top)$ is not an allowed signed type. A* signed conjunctive concept type *is a set $\{s_1,\ldots,s_p\}$ of signed concept types.*

Definition 10 (Polarized CGs). *A polarized CG is defined as a simple CG with signed types used in the labels of concept nodes and relations.*

Though that transformation does not preserve reasonings, it is possible to encode a polarized graph into a simple CG. Let us consider the following transformation *sg*:

For each type $t \neq \top$ appearing in a type hierarchy T of the vocabulary V, replace t by the two types $+t$ and $-t$ in the same type hierarchy of the vocabulary $sg(V)$. Then for each $t \leq t' \neq \top$ in T, add $+t \leq +t'$ and $-t' \leq -t$ in the type hierarchy T of $sg(V)$. $\top$ is also the maximal element in the obtained type hierarchy.

For each signed type $(+,t)$ or $(-,t)$ appearing in a polarized graph G, replace that type by the corresponding type $+t$ or $-t$ in the simple CG $sg(G)$

Semantics. Let $G = (C, R, \gamma, \lambda)$ be a polarized CG. We translate G by a FOL formula $\Phi^*(G)$ defined as follows: for every concept type c with type $(+,t_1) \sqcap \ldots \sqcap (+,t_k) \sqcap (-,t'_1) \sqcap \ldots \sqcap (-,t'_q)$, we have the formula $\phi^*(c) = t_1(\mathit{marker}(c)) \wedge \ldots \wedge t_k(\mathit{marker}(c)) \wedge \neg t'_1(\mathit{marker}(c)) \wedge \ldots \wedge \neg t'_q(\mathit{marker}(c))$. For every relation with type $(+,t)$, we have $\phi^*(r) = \phi(r)$, and for every relation with type $(-,t)$ we have $\phi^*(r) = \neg(\phi(r))$. Then the formula $\phi^*(G)$ is built from the formulae $\phi^*(x)$ in the same way as $\Phi(G)$ is built from the formulae $\phi(x)$.

Computing Deduction. Though satisfiability of a polarized CG is easy to check in linear time, the homomorphism mechanism is insufficient to compute deduction in this formalism (a Π^2P-complete problem). One has to rely upon *completions*.

Property 1 (Satisfiability). A polarized CG G, with $nf(G) = (C, R, \gamma, \lambda)$, is unsatisfiable if and only if either there exists a concept node $c \in C$ and concept types t, t' such that $\{(+,t),(-,t')\} \subseteq \mathit{type}(c)$, and $t \leq t'$; or there exists two relations r and r' such that $\gamma(r) = \gamma(r')$, $\lambda(r) = (+,t)$, $\lambda(r') = (-,t')$ and $t \leq t'$.

Property 2. Let F and Q be two polarized CGs over a vocabulary V. Then $\Phi(sg(V))$, $\Phi(sg(F)) \vdash \Phi(sg(Q)) \;\Rightarrow\; \Phi(V), \Phi^*(F) \vdash \Phi^*(Q)$, but the converse is false in general.

We have here encoded part of the negation semantics with simple CGs, but we're still missing the axioms translating the excluded middle principle. Intuitively, let us consider fact G asserting that a blue cube A is on top of a cube B that is itself on top of a cube C that is not blue. Now our question Q is: is there a blue cube x on top of a cube y that is not blue ? Though there can be no homomorphism from Q to G, FOL asserts that the cube B is either blue or not blue. And in both cases we find an answer to the question Q, thus proving deduction. This is the kind of reasonings that led [13] to use the notion of completion in order to obtain a sound and complete deduction mechanism.

Definition 11 (Completion). *A completion of a polarized CG G is a satisfiable polarized CG G' obtained from G by a sequence of applications of the following rules. G' is said maximal if no application of one of these rules produces new information.*
AddC: *If c is a concept node of G and t is a relation type appearing in G, then replace marker(c) with marker$(c) \cup \{(+,t)\}$ or marker$(c) \cup \{(-,t)\}$;*
AddR: *If $c_1, \ldots, c_p$ are concept nodes of G and t is a relation type of arity p appearing in G, then add a relation r with $\gamma(r) = (c_1, \ldots, c_p)$ and $\lambda(r) = (+,t)$ or $(-,t)$.*

Theorem 5. *Let F and Q be two satisfiable polarized CGs defined over $\mathcal{V}$. Then $\Phi(\mathcal{V})$, $\Phi^*(F) \vdash \Phi^*(Q)$ iff, for every completion F' of F, $\Phi(sg(\mathcal{V})), \Phi(sg(F')) \vdash \Phi(sg(Q))$.*

3.2 Computing PG Deduction with Default CG Rules

Let us now show that default CG rules can handle negation of polarized graphs. We consider a set $\mathcal{D}^*$ of default rules that handles the completion mechanism. To each concept type t we can associate two default CGs whose logical translation are:

$$D^+(t) = \frac{+\top(x) : -t(x)}{+t(x)} \text{ and } D^-(t) = \frac{+\top(x) : +t(x)}{-t(x)}$$

Intuitively, the first one asserts that for any concept node c and any concept type t, we can assert that c has type t unless something else makes us deduce that c has type $\neg t$. And to each relation type t of arity k we can also associate two defaults CGs:

$$D^+(t) = \frac{+\top(x),\ldots,+\top(x_k) : -t(x_1,\ldots,x_k)}{+t(x,\ldots,x_k)} \text{ and } D^-(t) = \frac{+\top(x_1),\ldots,+\top(x_k) : +t(x_1,\ldots,x_k)}{-t(x_1,\ldots,x_k)}$$

Theorem 6. *Let F and Q be two satisfiable polarized CGs defined over $\mathcal{V}$. Then $\Phi(\mathcal{V})$, $\Phi^*(F) \vdash \Phi^*(Q) \Leftrightarrow \mathcal{K} = ((\Phi(sg(\mathcal{V})) \wedge \Phi(sg(F))); \Delta(\mathcal{D}^*)) \vdash_S \Phi(sg(Q))$.*

Proof: The proof is immediate, and based on Th. 5, since we can easily check that the fact labeling each node of DDT$(\mathcal{K})$ is a completion of F, and then that the extensions of $\mathcal{K}$ encodes all the possible maximal completions of F. □

A first remark is that encoding the completion mechanism in default CG rules seems like overkill. Indeed, a simple negation as failure mechanism asserting, for example, that for any concept node c, we can add him the type $+t$ unless $-t$ is already present would provide us with the same result, without resorting to such a complex reasoning mechanism. But that simple solution would not cope with polarized rules.

3.3 An Extension to Polarized Graphs Rules

We show here that the set of default rules $\mathcal{D}^*$ used to represent the semantics of polarized graphs is sufficient to handle those of a new CG language that combines rules and atomic negation: polarized CG rules.

Definition 12 (Polarized CG rules). *A* polarized CG rule *is a pair $R = (H, C)$ of polarized graphs where H is called the hypothesis of the rule and C its conclusion.*

Semantics. Let $R = (H, C)$ be a polarized CG rule. Then the FOL interpretation of the rule R is the formula $\Phi^*(R) = \forall x_1 \dots \forall x_k (\phi^*(H) \rightarrow (\exists y_1 \dots \exists y_q \phi^*(C)))$, where $x_1, \dots, x_k$ are all the variables appearing in $\phi(H)$ and $y_1, \dots, y_q$ are all the variables appearing in $\phi(C)$ but not in $\phi(H)$. If $\mathcal{R}$ is a set of CG rules, then $\Phi^*(\mathcal{R})$ is the conjunction of the formulae $\Phi^*(R)$, for every rule $R \in \mathcal{R}$.

Computing Deduction

Theorem 7. *Let F and Q be two polarized CGs, and $\mathcal{R}$ be a set of polarized CG rules, all defined over a vocabulary $\mathcal{V}$. Then $\Phi(\mathcal{V}), \Phi^*(F) \vdash \Phi^*(Q)$ iff either 1) there exists an unsatisfiable polarized CG U such that $\Phi(sg(\mathcal{V})), \Phi(sg(F)), \Phi(sg(\mathcal{R})) \vdash \Phi(sg(U))$; or 2) $((\Phi(sg(\mathcal{V})) \wedge \Phi(sg(F)) \wedge \Phi(sg(\mathcal{R}))); \Delta(\mathcal{D}^*)) \vdash_S \Phi(sg(Q))$.*

Proof: We provide here the reader with a sketch of proof since a complete one would require a precise introduction of the combined Tableaux/CG reasoning mechanisms of [7]. First see that if condition 1) is satisfied, then the knowledge base $(\mathcal{V}, F, \mathcal{R})$ is unsatisfiable, and everything can be deduced from it. Then see that condition 2) means that Q can be deduced from any satisfiable polarized CG that can be obtained by a sequence of completion and rule application. The possible completions correspond to the different branchings in the tableaux algorithm of [7]. □

A range restricted rule is a rule where all generic markers of the conclusion are already in the hypothesis (e.g. such that their logical interpretation Φ^* admits no existentially quantified variable in their conclusion). Finally, we can present a first decidability result for polarized CG rules.

Theorem 8. *Let $\mathcal{R}$ be a set of* range restricted *polarized CG rules. Then the deduction problem is decidable.*

Proof: Thanks to Theorem 7, we can encode that deduction problem into a skeptical deduction using a default theory whose "normal"rules are those of $\mathcal{R}$, and whose defaults are those of $\mathcal{D}^*$. Since the standard part of these defaults is range restricted, and the union of two range restricted sets of rules is range restricted, we can conclude, thanks to Theorem 9 of [1], that the deduction problem is decidable. □

4 A Two-Players Game Artificial Intelligence Using Default CG Rules

In this section, we present an original application of default reasoning based on a two players board game. To ensure a maximum readability, we have chosen a very simple board game: Tic-Tac-Toe. This model can be easily adapted for other kind of games like Four in a Row. We begin to describe how to represent the initial state of the game that is a simple CG drawing of a 3×3 grid and two players. Then using default CG rules, we give the evolution rules that permits to obtain the tree of all possible states of the game. Finally, we give some default rules that allow to find the best way to play.

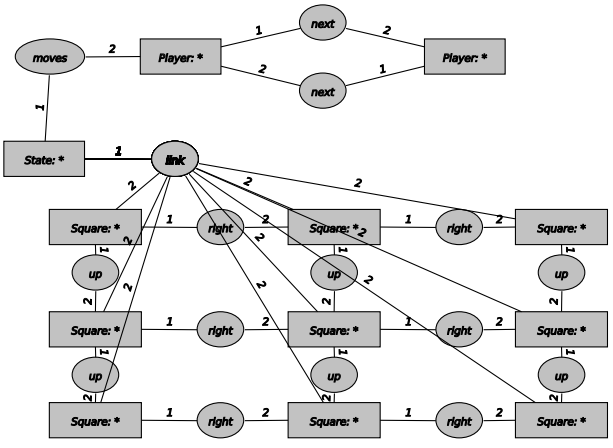


Fig. 1. Initial state of the board game

4.1 Initial Game Board

The initial game board is presented in Figure 1. Concept nodes typed `Player` represent the two who are engaged in the game. We present here the model of a two player game, but it is possible to add as many players as we want in the same way. A relation `next(Player,Player)` indicates who plays next someone. The state of the board during the game is represented by a concept node `State`. A state is linked with the player who has to play this turn by the relation $moves(State, Player)$. The `State` is also linked to several concept nodes that represent the current topology of the game board. For our Tic-Tac-Toe game, the board is constructed with 9 concept nodes `Square`, linked together by some relation `up` and `right` that indicates the relative position of each square. All the squares of a given state are related to the concept node `State` that it describes (see Figure 1).

4.2 Creating the Space of All Possible Games

Now let us see how to simulate a turn in a given state. To this purpose, we need some rules that create a new state, with a new board in which one the square has been modified. One rule is used to create the new state and duplicate the played square, and several ones are needed to rebuilt the whole board by cloning the squares of the precedent state. Figure 2.A presents the main playing rule. This default CG rule can be applied in a state in which there is an empty square. That means that the rule needs two justifications ensuring that any player already played in the square (the justifications of the defaults are represented here in dark nodes, please note that n different colors are needed to represent in a single graph n different justifications). We supposed that if a player has already won in a given state, then the state is tagged with the relation `over`. The fact that such a state can not be played is ensured by adding the needed justification. When applied,

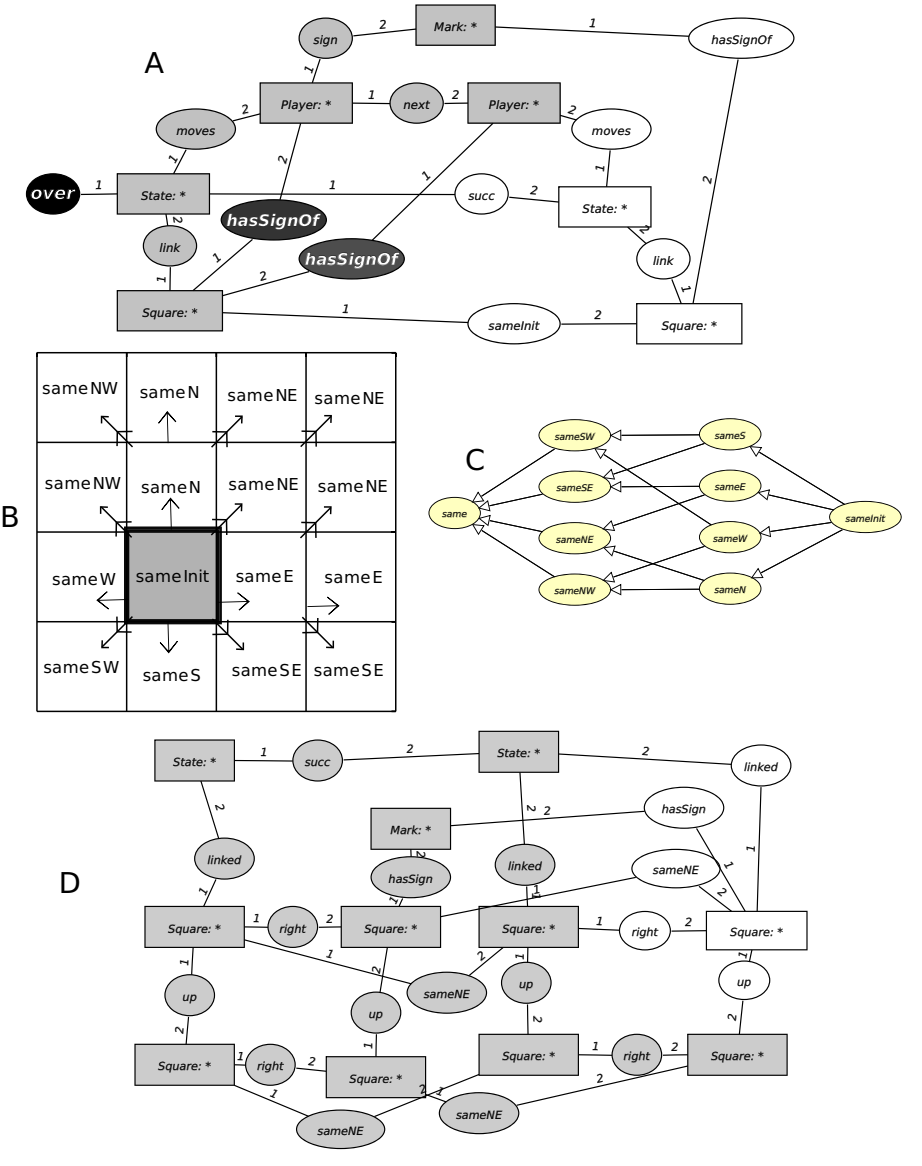


Fig. 2. Playing rules

the rule creates a new state that is indicated as a *successor* of the previous one, linked to the next player and a square linked to the player who just plays. This square is also linked to its previous instantiation by the relation `sameInit`.

Once this default rule is applied, the whole grid has to be replicated from the played square. The difficulty here is that we want to ensure that each square and relative position of the initial state is cloned once and only once, and linked to the correct player

mark. The method that we present here can be used to duplicate any kind of square-checkered board. To do that we start from the first duplicated square that is the last played one in the Tic-Tac-Toe game. This square is linked with its "father" by the special relation `sameInit`. From this square, we can duplicate the squares that are in the four cardinal directions with four different rules that use four specialized `same` relations named `sameN`, `sameS`, `sameE` and `sameW` which duplicate the squares in the north, south, east and west direction. The propagation of the cloning process is shown in Figure 2.B for a 4 by 4 grid.

Then four rules are needed to duplicate squares in each quarter of the plan. The rule that duplicates the right upper quarter plan (in the direction North-East) is given in Figure 2.D. To ensure that the duplication process runs correctly in each direction, it is sufficient to use the special hierarchy relation shown in Figure 2.C. It means that the relation `sameN` can be used for the generation of both north-east quarter plan and north west quarter plan. So all the duplication rules are directed and non–compatible (due to the relation hierarchy), and each square is duplicated once and only once.

Some rules are needed to end the game. For example if we can match 3 aligned squares (linked together by the relation `right`) played by the same player then we conclude that this player wins, the other player looses and the corresponding state is over. This rule is not a default one. Two other default rules are also needed, one to end the party tagging `over` a state over if there is no more space on the grid, and the other one to deduce that an over state is draw unless there is a winner. Applying this set of rules permit to create the complete search tree of all possible games (see Figure 3) which is the only extension of our model, and knowing who is the winner of each terminal board.

4.3 Searching the Best Way to Play

Knowing who wins each terminal state in the graph of all possible state makes easy to recursively deduce the status of each state of the game. That will allow to determine for each state if a player is ensured to win, lose or can obtain a draw.

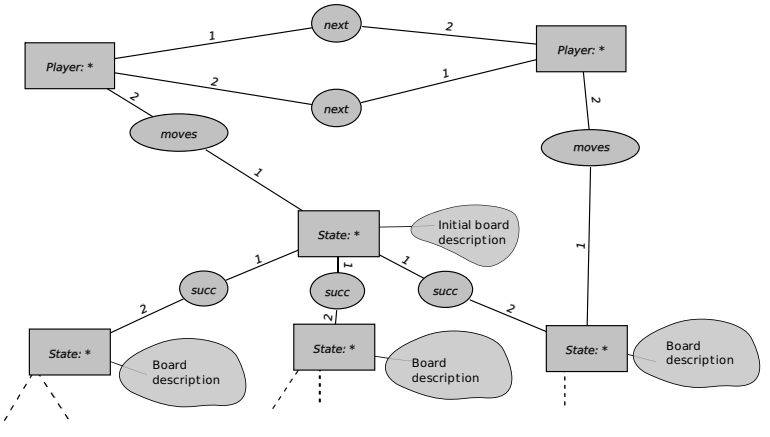


Fig. 3. General model of playing game

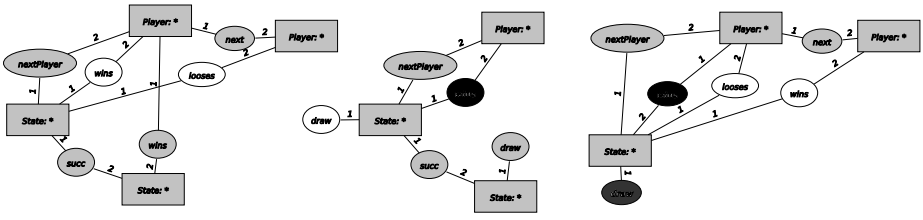


Fig. 4. Status deduction rules

To do that we need the 3 more rules shown in Figure 4 which shows the rules that permit to deduce the status of a state from the status of its successors. For a given state, the status of the game can be draw or won by a player. The status information is initialized in the terminal states, which are leaves of the tree of the possible states according to ending rules. The status of the game is then recursively computed from the leaves to the root of the possible states tree. For a state S, if there is an immediate state successor in which the current player of S is the winner, then S is a winning state for its current player because there is a way for the player to win whatever the other player does (first rule of Figure 4). Otherwise, if there is an immediate successor for which the game status is draw, then the status of the state S is draw, as it is possible for our player to ensure at least a draw (second rule of Figure 4). To apply this rule, we need to add that the current player is not already ensured to win, which is the justification of this rule. If it is not possible for the current player of S to win, nor to obtain a draw, then S is a loosing state for the current player. Note that in this presented rule, the two dark nodes represent two different justifications. That means that if one of this dark nodes can be projected, then the default can not be applied.

Finally, applying all this rules leads to one and only one extension in which we can easily find what is the best way for a player in any state of the game. As we have only one extension, the sceptical and credulous deductions are equivalent. This extension contains 26 830 different possible states of the game. This huge search tree does not take into account the fact that some states are equivalent up to reflexions and rotations (that can be computed through rules).

To know if there is a best way to play in one state, one can just try to find in the extension if the current player is linked to the initial state by the relation `wins`. If it is the case, then one of the best way to play can be found in searching one successor of the initial state in which our player is winning.

5 Conclusion

In this paper we have studied two applications of the default CG rules introduced in [1], in order to assess the expressivity of that new language for the CG family. We have first shown that default CGs allowed to express the semantics of atomic negation in FOL, with a concise and intuitive model (the set of defaults $\mathcal{D}^*$) that translated exactly the knowledge present in the polarized graph algorithm of [13]. In the same time, we have used default CGs to prove a new decidability result in an expressive CG language

that mixes CG rules with atomic negation. Then, we have exhibited a concise default CGs model of the Tic-Tac-Toe game that shows that, though default CGs are more complex than the usual CG rules, they offer possibilities to stop the generation of new consequences and thus, as can be done for the cut in Prolog, to encode reasonings in an efficient way. Our goal is now twofold: firstly, to encode a preference mechanism on the defaults, allowing for example in our game to consider first the extensions in which we can prove we will win, and only in case they are none the extensions in which we do not lose; and secondly, to study efficient reasoning mechanisms in default CG rules, building upon the results obtained for CG rules by [3].

References

1. Baget, J., Croitoru, M., Fortin, J., Thomopoulos, R.: Default conceptual graph rules: preliminary results for an agronomy application. In: Rudolph, S., Dau, F., Kuznetsov, S.O. (eds.) Conceptual Structures: Leveraging Semantic Technologies. LNCS (LNAI), vol. 5662, pp. 86–99. Springer, Heidelberg (2009)
2. Reiter, R.: A logic for default reasoning. Artificial Intelligence 13, 81–132 (1980)
3. Baget, J.F., Leclère, M., Mugnier, M.L., Salvat, E.: Extending decidable cases for rules with existential variables. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, pp. 677–682 (2009)
4. Calì, A., Gottlob, G., Lukasiewicz, T.: A general datalog-based framework for tractable query answering over ontologies. In: Proceedings of the Twenty-Eigth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS), pp. 77–86 (2009)
5. Sowa, J.F.: Conceptual graphs for a database interface. IBM Journal of Research and Development 20, 336–357 (1976)
6. Salvat, E., Mugnier, M.L.: Sound and complete forward and backward chaining of graph rules. In: Eklund, P., Mann, G.A., Ellis, G. (eds.) ICCS 1996. LNCS, vol. 1115, pp. 248–262. Springer, Heidelberg (1996)
7. Kerdiles, G.: Saying It with Pictures: a logical landscape of conceptual graphs. PhD thesis, Univ. Amsterdam (2001)
8. Tepfenhart, W., Cyre, W.: Proceedings of the 7th International Conference on Conceptual Structures, ICCS 1999, Blacksburg, VA, USA, July 12-15. Springer, Heidelberg (1999)
9. Baget, J.F.: Simple Conceptual Graphs Revisited: Hypergraphs and Conjunctive Types for Efficient Projection Algorithms. In: Ganter, B., de Moor, A., Lex, W. (eds.) ICCS 2003. LNCS (LNAI), vol. 2746, pp. 229–242. Springer, Heidelberg (2003)
10. Gottlob, G., Leone, N., Scarcello, F.: A comparison of structural CSP decomposition methods. Artificial Intelligence 124 (2000)
11. Sowa, J.F.: Conceptual Structures: Information Processing in Mind and Machine. Addison Wesley, Reading (1984)
12. Brewka, G., Eiter, T.: Prioritizing default logic: Abridged report. In: Festschrift on the occasion of Prof.Dr. W. Bibel's 60th birthday. Kluwer, Dordrecht (1999)
13. Mugnier, M., Leclere, M.: On querying simple conceptual graphs with negation. Data & Knowledge Engineering 60(3), 468–493 (2007)

On the Stimulation of Patterns

Definitions, Calculation Method and First Usages

Ryan Bissell-Siders, Bertrand Cuissart, and Bruno Crémilleux

Groupe de Recherche en Électronique, Informatique et Imagerie de Caen,
CNRS-UMR6072, Université de Caen, France
ryan.bissell-siders@info.unicaen.fr, cuissart@info.unicaen.fr,
cremilleux@info.unicaen.fr
http://www.greyc.unicaen.fr/

Abstract. We define a class of patterns generalizing the *jumping emerging patterns* which have been used successfully for classification problems but which are often absent in complex or sparse databases and which are often very specific. In supervised learning, the objects in a database are classified a priori into one class called *positive* – a *target* class – and the remaining classes, called *negative*. Each *pattern*, or set of attributes, has support in the positive class and in the negative class, and the ratio of these is the *emergence* of that pattern; the *stimulating patterns* are those patterns a, such that for many closed patterns b, adding the attributes of a to b reduces the support in the negative class much more than in the positive class. We present methods for comparing and attributing stimulation of closed patterns. We discuss the complexity of enumerating stimulating patterns. We discuss in particular the discovery of highly stimulating patterns and the discovery of patterns which capture contrasts. We extract these two types of stimulating patterns from UCI machine learning databases.

1 Introduction

We introduce *stimulation*, a new measure of interest in the classification of objects by their attributes. We suppose here that a dataset is a finite list of descriptions of objects, where the description of each object corresponds to a list of its binary attributes. A *pattern* denominates a set of binary attributes. The *extent* of a pattern is the set of objects whose descriptions contain each attribute in the pattern. The objects are classified and we aim to predict the classification of a new object from its description. The *support* of a pattern in a class is the cardinality of the extent of the pattern, restricted to the objects in that class. The classification of a pattern is a function of its supports in the classes of the classification. The *stimulation* of a pattern captures its influence on the classification of other patterns.

When we consider whether a pattern favors a certain class, we refer to that class as the *positive* class, and to the union of the remaining classes as *negative*. If a pattern stimulates the classification of other patterns to be more positive,

M. Croitoru, S. Ferré, and D. Lukose (Eds.): ICCS 2010, LNAI 6208, pp. 56–69, 2010.

by removing more (or relatively more) negative objects than positive objects from the extent of other patterns, then this is a strong correlation between the pattern and the positive class. Such a pattern stimulates a positive classification not only alone, but when mixed with any other pattern, and we observe this by considering the stimulation of a pattern on all other patterns. In addition to patterns with a constant influence, patterns that have a variable influence on the classification of other patterns are interesting too: they are useful for adding a new dimension to an existing model. Our work found inspiration in [14], which mines a dataset to find an attribute with very variable influence on a set of mutually exclusive patterns $p_0 \dots p_n$, with the intention of explaining the difference between these patterns.

The relationship between a pattern and the classification is an interesting quantitative problem. Information gain measures the amount of information in the class which is explained by a pattern, [4]. If the class and an attribute are both continuous, the Gini index measures correlation between them, [8, Chapter 9]. When we focus on a positive class, the simple odds of a positive classification, given a pattern, are called the *emergence* of that pattern [1]. Patterns with high emergence are called *emerging patterns* (or EPs), they have found wide application since their introduction in the data-mining community [2]. Mining EPs produces a flood of rules, among which may be found some rules which are valuable for constructing a classification model or for explaining the classification in a human-readable way. EPs yielded successful characterizations of biochemical properties and medical data in [11,1]. EPs are used in top-performing classifiers [3,10,18] and in commercial algorithms to find rules to explain separations between groups [21]. Creating a dataset of chemical graphs and subgraphs is in itself an interesting problem; once the dataset is constructed, extracting emerging patterns produces rules of interest to chemists [16]. We choose to use emergence to measure the relationship between a pattern and the classification because the notion is simple and powerful, and it maintains continuity with [14].

The influence of one pattern on another has been considered theoretically in statistics. Conditional probability is able to analyze the correlation between a pattern with the classification. *Naïve bayesian classification* then makes the assumption that the influence of each attribute on the pattern is independent of which patterns have gone before. When we restrict our attention to the extent of a pattern q, the correlation of the classification with a pattern p containing q is called the *odds ratio* between p and q. It expresses the influence of the larger pattern p on the classification of the subpattern q. Mining the influence of a single attribute on a set of patterns, or visa versa, has been carried out efficiently with *contrast sets* [14]. Contrast sets are similar to EPs and are mined so as to explain the difference between two classifications, They can detect a threshold or a fault. They are useful for refining a model of the classification of other patterns.

Mining the influence of all patterns on all attributes is inefficient in general; techniques to reduce the EPs to a readable and meaningful set of patterns make it efficient to study the influence of each EP on the rest [18]. In this text, we organize pairs of patterns into groups which have been stimulated in the same

way, so that for each pair in a group, it becomes clear which parts of the patterns are responsible for their classification. In one experiment, we extract groups with high and uniform stimulation. These patterns can explain why an object has positive classification, for if EPs cover some of the attributes of an object, then only the remaining attributes would oppose a positive classification. As predicted in [12], these groups conservatively extend the EPs. In another experiment, we extract groups with highly varying stimulation. These patterns are useful for extending a classification model.

This paper is organized as follows. Section 2 outlines background concepts. We define stimulation in section 3. We present an algorithm to extract the stimulation measure and describe experiments in which this measure is of interest in section 4.

2 Preliminaries

2.1 Notions of Formal Concept Analysis [6,5]

We use standard notions from Formal Concept Analysis (FCA): a formal *context* denominates the triple (M, G, I) where the binary *attributes* M and *objects* G are related by the dataset $I \subseteq G \times M$. In FCA *concepts* are the inclusion-maximal sets $a \subseteq I$ of the form $a = A \times B, A \subseteq G, B \subseteq M$. A set of attributes is named a *pattern*. The *extent* of an attribute $m \in M$ is $\{g \in G : (g, m) \in I\}$. The *extent* of a pattern p, denoted $\text{ext}(p)$, is the intersection of the extents of its attributes. Likewise, the *intent* of an object g is the set of attributes m such that $(g, m) \in I$, and the *intent* of a set $A \subseteq G$ of objects, denoted $\text{int}(A)$ is the intersection of the intents of its objects. For any two patterns a, b we write $a < b$ and say a is *more specific than* b just in case $\text{ext}(a) \subset \text{ext}(b)$.

The function taking a pattern a to the intent of its extent, denoted $\overline{a}$, is a *closure* function on patterns: for any two patterns a, b, $\overline{a} \subseteq \overline{b}$ holds whenever $a \subseteq b$ and $\overline{\overline{a}} = \overline{a}$. The set $\overline{a}$ is called *closed*. An elegant alternate notation [6] is to write x' for both $\text{int}(x)$ and $\text{ext}(x)$ and x'' for $\overline{x}$. We denote by 0 the concept that satisfies $\text{int}(0) = M$; similarly, we denote by 1 the concept that satisfies $\text{ext}(1) = G$. For any two patterns a and b, $a \vee b$ is a least upper bound – the least (most specific) closed pattern c such that $a < c$ and $b < c$. Likewise, $a \wedge b$ is a greatest lower bound – the greatest (least specific) closed pattern c below a and b. Because $\text{ext}(a \wedge b) = \text{ext}(a) \cap \text{ext}(b)$ and $\text{int}(a \vee b) = \text{int}(a) \cap \text{int}(b)$, the upper and lower bounds are unique, so the set of closed patterns inherits the structure of a lattice from the Boolean algebra of subsets of G (or of M) with $\vee$ and $\wedge$ defined as above.

The lattice structure on the set of closed patterns can be recovered from a single function, the *upper covers*. Defined on any lattice $\mathcal{L}$, the *upper covers* is the function from $a \in \mathcal{L}$ to the set of its immediate successors (those $b \in \mathcal{L}$ such that $a < b$ and there is no c such that $a < c < b$). We represent a lattice $\mathcal{L}$ by storing only its domain, also denoted $\mathcal{L}$, the extent and intent and upper covers functions.

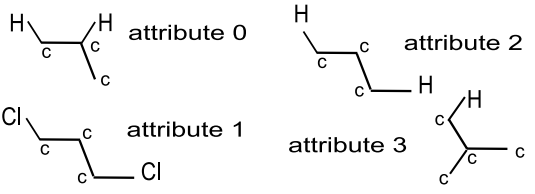


Fig. 1. The attributes used to describe the molecules in table 1 are the presence or absence of these subgraphs, each of which has 5 elements

Table 1. Polychlorinated byphenyl molecules. The first column is the molecule's descriptive name. The next four columns indicate the presence of attributes $0, 1, 2, 3$, i.e., the existence of an isomorphic copy of the corresponding subgraph from figure 1. The final column indicates which two molecules are most toxic.

The molecule's name	attributes:				toxicity
	att. 0	att. 1	att. 2	att. 3	
3,3',4,4'-TetraCB	1	0	0	0	
3,4,4',5-TetraCB	1	1	1	0	
3,3',4,4',5-PentaCB	1	1	0	0	most toxic
3,3',4,4',5,5'-HexaCB	0	1	0	0	most toxic
2,3,3',4,4'-PentaCB	1	1	0	1	
2,3,4,4',5-PentaCB	1	1	1	1	
2,3',4,4',5-PentaCB	1	1	0	1	
2',3,4,4',5-PentaCB	1	1	1	1	
2,3,3',4,4',5-HexaCB	1	1	0	1	
2,3,3',4,4',5'-HexaCB	0	1	0	1	
2,3',4,4',5,5'-HexaCB	0	1	0	1	
2,3,3',4,4',5,5'-HeptaCB	0	1	0	1	

Illustration. Figure 1 shows four chemical graphs which we use as attributes. If a molecule contains an isomorphic copy of one of these as an induced subgraph, then we say that it contains that subgraph as an attribute. The figure lists all graphs with 5 atoms which are present in at least two polychlorinated biphenyls and not present in at least two polychlorinated biphenyl molecules (PCBs). Table 1 displays (PCBs) and their subgraph attributes. These twelve PCBs are of special concern and are regulated in international law for their toxicity. Two of them are orders of magnitude more toxic than the others. Figure 2 displays the lattice of concepts for this dataset, showing the intent of the concept as a set in each circle. The edges in the diagram represent the upper covers.

2.2 Definition of Emergence[2]

Given a classification of the objects G into *positive* and *negative* classes G_0 and G_1, the *emergence* of a pattern a compares the *frequencies* of a, where the *frequency* of a pattern a within a class C indicates the portion of the objects of C that are in relation with a. If $C \neq \emptyset$, define $\text{frequency}(a, C) = \frac{|\{g \in \text{ext}(a) : g \in C\}|}{|\{g \in C\}|}$. The emergence of a pattern is the defined as:

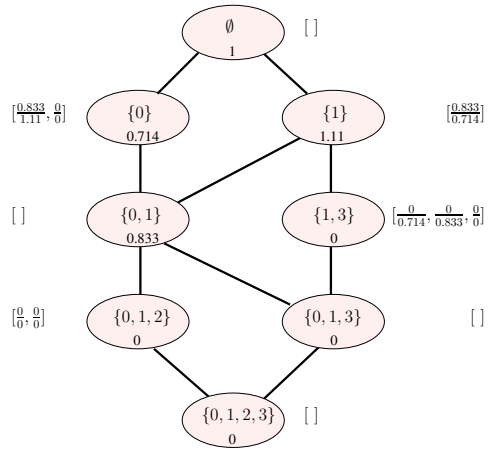


Fig. 2. The lattice of concepts of table 1, showing each concept's intent (as a set), emergence (definition 1) as a numerical value under the intent, and the stimulation set (definition 4) as a multiset $[\frac{a_0}{b_0}, \dots]$ to the side

Definition 1 (Emergence[2]). *Given positive class G_0 and negative class G_1, for any pattern $a \subseteq M$, the emergence of a is*

- $\text{emergence}(a) = \dfrac{\text{frequency}(a, G_0)}{\text{frequency}(a, G_1)}$ *if* $\text{frequency}(a, G_1) \neq 0$
- $\text{emergence}(a) = \infty$ *if* $\text{frequency}(a, G_1) = 0$ *and* $\text{frequency}(a, G_0) \neq 0$
- $\text{emergence}(a)$ *is not defined if* $\text{frequency}(a, G_1) = 0$ *and* $\text{frequency}(a, G_0) = 0$

If the role of the class G_0 and its complement G_1 were reversed in the above definition, the emergence of any pattern would become its inverse. We could specify the emergence defined above to be the emergence *into* G_0*, relative to* G_1, denoted $\text{emergence}_{(G_0,G_1)}(a)$. But for the sake of simplicity, we denote $\text{emergence}_{(G_0,G_1)}(a)$ as $\text{emergence}(a)$, supposing that we have already directed our attention towards G_0 as the *positive* class.

If the emergence of a is ∞, then a is called a *jumping emerging pattern* (JEP); if the emergence of a is 0, then we call a an *anti*-JEP. We will be careful to never refer to the emergence of a pattern with empty extent, so that our definition of emergence differs in no way from that of ([2], p.45).

Illustration. Figure 2 displays the closed patterns $\emptyset$, $\{0\}$, $\{1\}$, $\{0,1\}$, $\{1,3\}$, $\{0,1,2\}$, $\{0,1,3\}$, $\{0,1,2,3\}$. The frequency of each of these patterns among the most-toxic molecules can be observed from table 1 to be, respectively, $\frac{2}{2}$, $\frac{1}{2}$, $\frac{2}{2}$, $\frac{1}{2}$, 0, 0, 0, 0. The frequency of each of these patterns among the rest is: $\frac{10}{10}$, $\frac{7}{10}$, $\frac{9}{10}$, $\frac{6}{10}$, $\frac{8}{10}$, $\frac{3}{10}$, $\frac{5}{10}$, $\frac{2}{10}$. The emergence of each pattern is the ratio of its frequency among the most-toxic molecules to its frequency among the rest. Figure 2 shows the emergence of each pattern in the circle, under pattern (the intent of the concept).

3 Stimulation of a Pattern

3.1 Definition of Stimulation

We define the stimulation of a pattern a on b to be the ratio of the emergence of $a \wedge b$ to the emergence of b.

Definition 2 (Stimulation). *Let (a, b) be an ordered pair of patterns. If $a \wedge b \neq 0$, the* stimulation of a on b, *denoted* stimulation(a, b), *is defined to be:*

- stimulation$(a, b) = \frac{\text{emergence}(a \wedge b)}{\text{emergence}(b)}$ *if* $0 < \text{emergence}(a \wedge b) < \infty$,
- stimulation$(a, b) = \infty$ *if* emergence$(a \wedge b) = \infty$ *and* emergence$(b) < \infty$,
- stimulation$(a, b) = 0$ *if* emergence$(a \wedge b) = 0$ *and* $0 <$ emergence(b),
- stimulation$(a, b) = 1$ *if* emergence$(a \wedge b) =$ emergence$(b) = 0$ *or* emergence $(a \wedge b) =$ emergence$(b) = \infty$.

Since $\text{ext}(a \wedge b) \subseteq \text{ext}(b)$, the condition $a \wedge b \neq 0$ implies $b \neq 0$. Consequently, when $a \wedge b \neq 0$, both emergence$(a \wedge b)$ and emergence(b) are defined. In particular, if b is a JEP (resp. an anti-JEP) then $(a \wedge b)$ is a JEP (resp. an anti-JEP). If stimulation(a, b) is a finite fraction and we switch G_0 and G_1, then stimulation(a, b) becomes its own inverse.

Illustration. In figure 2 the pattern $\{0, 1\}$ appears, so it is a closed pattern and it is the intent of a concept. It has frequency $\frac{1}{2}$ among the most toxic molecules and frequency $\frac{6}{10}$ among the rest. Its emergence is, then, $\frac{1}{2}/\frac{6}{10}$. Its upper cover $\{0\}$ has frequency $\frac{1}{2}$ among the most toxic molecules and frequency $\frac{7}{10}$ among the rest. Its emergence is $\frac{1}{2}/\frac{7}{10}$. The fact that $\text{ext}(\{0\}) \setminus \text{ext}(\{0, 1\})$ contains a single molecule – the molecule which does not have attribute 1 – is reflected in stimulation$(\{1\}, \{0\}) = \frac{0.833}{0.714} = \frac{7}{6}$.

Factorization of stimulation. Stimulation allows us to factor "the odds of a and b" into "the odds of a" and "the stimulation of b on a"; likewise, we can factor the "stimulation of a_0 and a_1 on b" into "the stimulation of a_0 on b" and "the stimulation of a_1 on $(a_0 \wedge b)$." Stimulation thus transforms the interaction of patterns into multiplication:

Proposition 3. stimulation$(a, b) \times$ stimulation$(c, a \wedge b) =$ stimulation$(a \wedge c, b)$

Proof. The emergence of $a \wedge b$ is the numerator in the first multiplicand and the demoninator in second multiplicand. □

As a corollary: if a stimulates b to a degree > 1 and c stimulates $a \wedge b$ to a degree > 1, then $a \wedge c$ stimulates b to a degree > 1.

3.2 Stimulation Set

Reducing the domain of stimulation. Since the emergence of a pattern is defined from its extent, the emergence of a set of attributes is the same as the

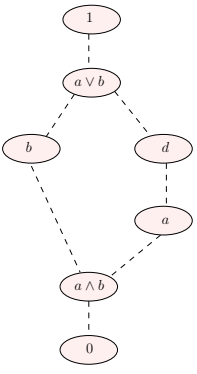


Fig. 3. a is not responsible for the change of emergence between b and $a \wedge b$

emergence of its closure. Let a and b be patterns. As $\text{ext}(a) = \text{ext}(\overline{a})$ and $\text{ext}(b) = \text{ext}(\overline{b})$, we have $\text{ext}(a \wedge b) = \text{ext}(\overline{a} \wedge \overline{b})$. Furthermore, by definition, we have $\text{ext}(a \wedge b) = \text{ext}(\overline{a \wedge b})$. It results $\overline{a} \wedge \overline{b} = \overline{a \wedge b}$. In our case, this equality implies that stimulation(a, b) is defined if stimulation$(\overline{a}, \overline{b})$ is defined. Moreover, if stimulation(a, b) is defined, we have stimulation$(a, b) =$ stimulation$(\overline{a}, \overline{b})$. *Consequently, the domain of the second argument of stimulation is naturally narrowed to closed patterns, and to the concepts which have these closed patterns as intents.* The domain of the first argument could be left as patterns, but we choose to restrict it as well to closed patterns, or their concepts.

The responsibility of a stimulation. We classify pairs of concepts $\{(b, c) : b > c\}$ so that (b, c) and (f, g) are in the same group if there exists any concept e such that $b \wedge e = c$ and $f \wedge e = g$. In this case, we say that e might be *responsible* for the difference between b and c. So, while stimulation captures a change in emergence, we want *stimulation sets* to classify stimulation(a, b) by the proper argument a which is really responsible for the stimulation from b to $c = b \wedge a$. The change in emergence from pattern b to pattern $b \wedge a$ can be attributed to different patterns, and not only to a itself. Suppose there exists $d \in \mathcal{L}$ such that: $d < a$ and $\text{ext}(d \wedge b) = \text{ext}(a \wedge b)$ (see Figure 3). Consequently $d \wedge b \neq 0$ is equivalent to $a \wedge b \neq 0$, stimulation(d, b) is defined if stimulation(a, b) is defined. Moreover, when stimulation(a, b) is defined we have stimulation$(d, b) =$ stimulation(a, b).

We decide to attribute stimulation(a, b) to a if a is the smallest pattern d such that $\text{ext}(d \wedge b) = \text{ext}(a \wedge b)$. If there exists some $d < a$ such that $\text{ext}(d \wedge b) = \text{ext}(a \wedge b)$, then we consider that the stimulation should be attributed to d rather than to a. We write $[f(a) : a \in U]$ for a multiset of values.

Definition 4 (Stimulation set). *Let $a \in \mathcal{L}$. The* stimulation set *of a, denoted $SS(a)$, is the multi-set which is a subset of* [stimulation(a, b) : $a \wedge b \neq 0$] *for which* stimulation(a, b) *belongs to $SS(a)$ just in case :*

i) $a \not\leq b$ *and*
ii) there is no $d \in \mathcal{L}$ *such that* $d \geq a$ *and* $d \wedge b = a \wedge b$.

If $b \leq a$, we suppress stimulation(a, b) when printing $SS(a)$, as in figure 2.

Illustration. The stimulation sets shown in figure 2 are in brackets to the left and right of the concepts in the lattice. Let us see why $SS(\{0,1\})$ is empty. By the first condition of definition 4, each pattern stimulates only concepts on the other side of the diagram. Further, $\{0,1\}$ can only stimulate concepts which are not related to it by $<$. This leaves only $\{1,3\}$. But taking $a = \{0,1\}$ and $b = \{1,3\}$, let $d = \{0\}$. stimulation$(\{0,1\},\{1,3\})$ = stimulation$(\{0,1\},\{1,3\})$, so by the second condition in definition 4, $\{0,1\}$ is not responsible for this stimulation value. The edges and paths in figure 2 which can be attributed to changes off the path itself are partitioned into $SS(a)$. The following lemma shows that it is always the case that $\{(a,b) : a < b\}$ is almost-partitioned into the stimulation sets. An edge will be present in two stimulation sets $SS(a), SS(b)$ if both a and b are minimal explanations responsible for that edge.

Lemma 5. *For each $b, c \in \mathcal{L}$ such that $b < c$, there is at least one $a \in \mathcal{L}$ such that $\frac{\text{emergence}(c)}{\text{emergence}(b)} \in SS(a)$.*

Proof: $b \wedge c = c$, so $\{a \in \mathcal{L} : a \wedge b = c\}$ is not empty. This set has at least one minimal element a'. For each such minimal a', $SS(a')$ contains the desired fraction. □

Now we use the stimulation values to characterize which concepts consistently stimulate other concepts. Let $MS(a)$ be the minimal stimulation of a, i.e., the minimal value in $SS(a)$. We can bound $MS(a \wedge b)$ by $MS(a)$ and $MS(b)$:

Proposition 6. *There is an injection from $SS(a \wedge b)$ into $SS(a) \times SS(b)$ such that when* stimulation$(a \wedge b, c) \mapsto (r_0, r_1)$, stimulation$(a \wedge b, c) > r_0 \times r_1$.

Proof. If $p_0 \subseteq p_1 \subseteq p_2$ are closed patterns, and $\text{ext}(a) \cap \text{ext}(p_0) = \text{ext}(p_1)$ and $\text{ext}(b) \cap \text{ext}(p_1) = \text{ext}(p_2)$, then $\text{ext}(b) \cap \text{ext}(a) \cap \text{ext}(p_0) = \text{ext}(p_2)$. If a is not a pattern of minimal intent such that $\text{ext}(a) \cap \text{ext}(p_0) = \text{ext}(p_1)$, say, $d \subseteq a$ and $\text{ext}(d) \cap \text{ext}(p_0) = \text{ext}(p_1)$, then $(\text{ext}(b) \cap \text{ext}(d)) \cap \text{ext}(p_0) = \text{ext}(p_2)$, so that $\text{ext}(b) \cap \text{ext}(a)$ is not the minimal c such that $c \cap \text{ext}(p_0) = \text{ext}(p_2)$. Likewise, if b is not the pattern of minimal intent such that $\text{ext}(b) \cap \text{ext}(p_1) = \text{ext}(p_2)$, then $\text{ext}(b) \cap \text{ext}(a)$ is not the minimal c such that $c \cap \text{ext}(p_0) = \text{ext}(p_2)$. □

Application of the lemma and proposition. Definition 4 prunes some emergence ratios from the notion of stimulation. Lemma 5 shows that the pruning is conservative, preserving the emergence ratios for any interval in the lattice. Proposition 6 shows that if the minimal value in $SS(a)$ is $MS(a)$ and the minimal value in $SS(b)$ is $MS(b)$, then the minimal value in $SS(a \cup b)$ is $MS(a \cup b) \geq MS(a) \times MS(b)$. Thus the set of patterns with uniform, high stimulation is join-closed. We can then consider only the boundary of this set, when searching for highly stimulating patterns.

In this section, we have defined a new measure of interaction for any ordered pair of patterns which captures how emergence changes under additional information or in a restricted situation. We have introduced the notion of the set of stimulation values for which a pattern a is responsible. In the next section,

Algorithm 1. Enumerate $SS(a)$ of all closed sets of objects a

Input: A Galois lattice $\mathcal{L}$ with extent, upper covers; a classification of the objects $G = G_0 \cup G_1$ into positive class G_1 and negative class G_0
Output: $\{\mathrm{SS}(a) : a \in \mathcal{L}\}$
Let T order $\mathcal{L}$ from the concept with the largest extent to the concept with the least extent, so that $a <_T b$ holds just in case the support of a in $G_0 \cup G_1$ is $\geq$ the support of b in $G_0 \cup G_1$.
Let L order $\mathcal{L}$ (arbitrarily).
foreach $b \in L$ **do**
  compute the support of $\mathrm{ext}(b)$ in both G_0 and G_1.
  foreach $a \in T$ **do**
    if $a = b$ **then**
      write $a \leq b$.
    compute the support of $\mathrm{ext}(b) \cap \mathrm{ext}(a)$ in both G_0 and G_1.
    if *for each upper cover* a' *of* a*:*
    $b \geq a'$ *fails (else, save* $b \geq a$*) and*
    $\mathrm{ext}(b) \cap \mathrm{ext}(a') = \mathrm{ext}(b) \cap \mathrm{ext}(a)$ *fails* **then**
      add stimulation(a, b), as a 4-tuple of supports, to $\mathrm{SS}(a)$.

we address the problem of calculating the stimulation sets, and we describe experiments extracting stimulation sets which mostly contain large values and stimulation sets which contain as widely varying a set of values as possible.

4 Computing Stimulation

4.1 Calculation of the Stimulation Sets

A naïve search through the lattice finds all pairs $b \geq c$, and assigns the ratio of their emergence to some other pattern a, thus computing a matrix of stimulations for $a, b \in \mathcal{L}$. See algorithm 1 for the pseudocode. For some uses of this stimulation matrix, it may be possible to achieve that use without the naïve time-complexity factor $|\mathcal{L}|^2$.

Sound and complete. This algorithm computes $SS(a)$ as in definition 4. The first condition, that $a \not\leq b$, is enforced by storing $f(a) = \{b : a \leq b\}$, which can be computed from the set of $f(a')$ for which a' is an upper cover of a, since $a \leq b$ holds just in case $a = b$ or for some upper cover a' of a we have $a < a'$ and $a' \leq y$. The second condition is enforced by comparing the support of $a \wedge b$ with the support of $a' \wedge b$ for each upper cover a' of a. If for no upper cover $a' > a$ do we have $a' \wedge b = a \wedge b$, then for no $d > a$ do we have $d \wedge b = a \wedge b$, because the function $a \mapsto \mathrm{ext}(a)$ is monotonic, taking $\leq$ to $\subseteq$, so the function $a \mapsto \mathrm{ext}(a) \cap \mathrm{ext}(b)$ is monotonic, too. Thus, this algorithm searches all triples $(a, b, a \wedge b)$ and assigns the value stimulation(a, b) to $SS(a)$ just in case the two conditions in definition 4 hold.

Table 2. For the PCB dataset in table 1, these concepts were consistently stimulating (or anti-stimulating) and had no upper cover with the same stimulation

Intent	stimulation at 10^{th} or 90^{th} percentile	stimulation set
$\{1\}$	stimulates ≥ 1.1	$SS\{1\} = [7/6]$
$\{1,3\}$	is anti-JEP	$SS(\{1,3\}) = [0, 0, \frac{0}{0}]$
$\{0\}$	stimulates $\leq 1/1.2$	$SS\{0\} = [3/4, \frac{0}{0}]$
$\{0,1,2\}$	is anti-JEP	$SS(\{0,1,2\}) = [0, 0, \frac{0}{1.11}]$

Complexity. The runtime of the algorithm as given is dominated by its obviously nested loops, and is bounded by $|\mathcal{L}|^2 \times \sup_a |\text{upper cover}(a)| \times \sup_{a,b} |\text{ext}(a) \cap \text{ext}(b)|$, where $\sup_a |f(a)| = \sup\{f(a) : a \in \mathcal{L}\}$ denotes the maximum cardinality of $f(a)$ as a varies over $\mathcal{L}$.

Lower complexity. Let L be a traversal of $\mathcal{L}$, ascending from the concept with minimal extent and stepping always from a closed pattern b_0 to a new pattern b such that $\text{ext}(b) \setminus \text{ext}(b_0)$ is minimal; then we can check whether $\text{ext}(b) \cap \text{ext}(a') \neq \text{ext}(b) \cap \text{ext}(a)$ by examining only elements of $\text{ext}(b) \setminus \text{ext}(b_0)$, which reduces the last factor in the runtime to $\sup\{|(\text{ext}(b) \setminus \text{ext}(b_0)) \cap \text{ext}(a)| : b$ is an upper cover of b_0 and $a, b, b_0 \in \mathcal{L}\}$.

4.2 Experiment

We implemented the algorithm introduced in the previous subsection. From the archive of datasets stored at UCI (`www.ics.uci.edu/~mlearn/`), we extracted concept lattices using the Galicia suite of programs [20]; we subsequently extracted $\{SS(a) : a \in \mathcal{L}\}$. We found that whenever Galicia could extract a concept lattice without overflowing memory, we were able to extract the stimulation: during our experience, the factor of $|\mathcal{L}|^2$ in the runtime is not an order of magnitude more prohibitive to computation than the decision to operate with a concept lattice.

Our goal was to discover 1. highly stimulating sets that are not JEPs, and 2. to discover non-homogeneity. Drawbacks to a purely JEP-based classification are discussed in [13]. For any dataset in which every rule $\text{ext}(p) \subseteq G_i$ has exceptions, there are no JEPs. For many datasets, there are a flood of JEPs. In some fuzzy datasets such as census and satellite images (according to a study of their second-order properties in [14]), emerging patterns fail to extract certain important properties, and logically more flexible rules are desirable. However, we chose to extract stimulating sets first in contexts where JEPs classify *well*, so as to evaluate the "border" of new information which they add to the emerging patterns.

If we discover the highly stimulating patterns in the PCB example, we get the list in table 2. The value of stimulation(a, b) is ∞/∞ when b is a JEP; the stimulation is $0/0$ when b is an anti-JEP. These values are reported as 1 in the table above, but do not count against the designation of a pattern as highly stimulating. Thus pattern $\overline{\{2\}}$ has $SS(\overline{\{2\}}) = [0, 0, 1, 1]$ and is called an anti-JEP even though 1 occurs with frequency $> 10\%$ because these values are not

Table 3. For the dataset shuttle-landing-control, these are the patterns with greatest extent which stimulated almost all other concepts, and which had no upper cover with (roughly) the same average stimulation

Intent	stimulation at 10^{th} or 90^{th} percentile	stimulation set
≥ 3	{VISIBILITY:yes}	$[1^{712}2.3^{5}3^{23}9.3^{4}12^{20}20^{6}\ldots\infty^{502}]$
$\leq 1/\infty$	{VISIBILITY:no}	$[0^{635}1^{699}]$
$\geq \infty$	{STABILITY:xstab,VISIBILITY:yes}	$[\infty^{222}]$
≥ 2	{SIGN:pp,VISIBILITY:yes}	$[0^{5}1^{9}9.3^{2}10^{3}12^{6}15^{3}22^{1}\ldots\infty^{165}]$
$\geq \infty$	{MAGNITUDE:OutOfRange,VISIBILITY:yes}	$[\infty^{134}]$
$\geq \infty$	{ERROR:XL,VISIBILITY:yes}	$[\infty^{135}]$
$\geq \infty$	{VISIBILITY:yes,ERROR:LX}	$[\infty^{135}]$
≥ 1.5	{SIGN:pp,VISIBILITY:yes,WIND:tail}	$[0^{3}1^{3}37.3^{1}48^{3}50^{2}54^{1}63^{1}\infty^{58}]$
≥ 1.5	{VISIBILITY:yes,ERROR:MM}	$[0^{9}1^{3}55^{4}60^{5}88^{1}99^{2}240^{2}\infty^{99}]$
≥ 1.5	{MAGNITUDE:Strong,VISIBILITY:yes}	$[0^{10}1^{1}39^{1}66^{6}264^{5}378^{1}456^{1}\ldots\infty^{103}]$
≥ 1.1	{MAGNITUDE:Low,VISIBILITY:yes}	$[0^{13}1^{1}66^{6}\ldots\infty^{98}]$
≥ 1.1	{MAGNITUDE:Medium,VISIBILITY:yes}	$[0^{13}1^{1}66^{6}70^{1}94.5^{1}\ldots\infty^{98}]$
≥ 1.2	{VISIBILITY:yes,WIND:tail,ERROR:MM}	$[0^{3}198^{2}210^{1}240^{2}247^{1}358.3^{1}\infty^{32}]$
$\geq \infty$	{VISIBILITY:yes,SIGN:nn,ERROR:MM}	$[1^{3}\infty^{42}]$
≥ 1.5	{MAGNITUDE:Low,VISIBILITY:yes,WIND:tail}	$[0^{4}189^{1}264^{2}273^{1}1056^{1}1075^{1}\infty^{34}]$
≥ 1.1	{MAGNITUDE:Strong,SIGN:pp,VISIBILITY:yes}	$[0^{5}189^{1}264^{2}304^{1}1056^{1}1148^{1}\infty^{33}]$
≥ 1.5	{MAGNITUDE:Medium,VISIBILITY:yes,WIND:tail}	$[0^{4}189^{1}264^{2}273^{1}1056^{1}1075^{1}\infty^{34}]$

counted against 2 in determining that it is infinitely stimulating. Ignoring the third line, we find the obvious classification that a PCB congener which is in the list of 12 toxins of concern for international control is highly toxic if: it contains attribute 1 but not attribute 2 or 3.

To test the algorithm in a "contrary" domain, we chose shuttle-landing-control from the UCI Machine Learning datasets. This dataset is presented as a set of 15 JEPS. The dataset is a rule base, where each rule has one of two forms: $\text{ext}(a) \subseteq G_0$, which indicates that if pattern a obtains, then G_0: the shuttle should be landed by a human; or or $\text{ext}(a) \subseteq G_1$, meaning that if a obtains, then the shuttle should be landed by autopilot. We expected to recover these 15 rules as highly stimulating rules as stimulating patterns. In addition, we found 7 other rules and 16 highly stimulating patterns. Table 3 displays the first 17 of these patterns, those with the smallest intent and largest extent, ranked by increasing intent and decreasing extent. The resulting 38 patterns are of interest in guiding the classification problem, as they are not numerous, and the new patterns behave as "fuzzy" JEPs.

Non-homogeneity. Even in the small dataset of shuttle-landing-control, with four multivariate attributes, one can find pairs of attributes (a, b) such that the matrix $\{\text{stimulation}(a_i, b_j) : i < n_a, j < n_b\}$ exhibits contrast behavior (it is "twisted"). In table 4 the values in each column are comparable, whereas the values in the last column are very different. Whether `STABILITY:stab` or `STABILITY:xstab` is more highly correlated with the positive classification also varies across columns. Thus, in order to explain the effect of the attribute `STABILITY` on the classification, one must discuss the attribute `ERROR`.

The runtime grows exponentially with even a trivial increase in the number of objects and attributes, a typical feature of operations on lattices. Runtimes

Table 4. Two attributes from the dataset shuttle-landing-control, and their matrix of emergences. If a is the attribute labeling the row and b is the attribute labeling the column, then the first two rows of the table list the supports of $\overline{\{a, b\}}$ in positive and negative classes, and the final two lines evaluate the emergence of the pattern $\overline{\{a, b\}}$.

	ERROR:XL	ERROR:LX	ERROR:MM	ERROR:SS
STABILITY:stab	(2025 / 2279)	(2025 / 2120)	(2025 / 848)	(4293 / 901)
STABILITY:xstab	(640 / 640)	(640 / 640)	(512 / 320)	(1280 / 0)
STABILITY:stab	0.9	1	2.4	5
STABILITY:xstab	1	1	1.6	∞

were calculated on a Dell Latitude D610: Pentium M, 800MHz, 1GB RAM. The example of 12 PCBs had 4 binary attributes, 12 objects, 8 concepts, 4 highly-stimulating concepts, and the highly stimulating concepts were extracted in < 1 second, producing a matrix $\{SS(a) : a \in \mathcal{L}\}$ with size 458b. The example of shuttle-landing-control had 16 binary attributes, 253 objects, 2040 concepts, of which 38 were extracted as highly-stimulating concepts; the highly stimulating concepts were extracted in 211 seconds, producing a matrix $\{SS(a) : a \in \mathcal{L}\}$ with size 1.98 Mb.

In conclusion, by testing the implemented algorithm on some small databases, we were able to discover homogeneously and highly stimulating patterns, which were a useful generalization of the jumping emerging patterns and yet were not too numerous. On the other hand, we were able to discover non-homogeneous interaction between attributes. The notion of stimulation captures some notions already studied in the literature. We generalize this notion to the interaction between any pair of patterns; the implementation allows for the extraction of patterns with either contrast-discriminative capacity or high, homogeneous stimulation, which should prove to be as useful as the contrast-discriminative pairs of attributes and as measures of confidence in further applications.

5 Further Work and Conclusion

There are interesting theoretical directions for further work, as well. We can discuss the stimulation of patterns on other patterns with respect to any concept-evaluating or pattern-evaluating measure, not only emergence. A second direction for further theoretical expansion is to consider how a single variable affects a pattern. If the variable a has attribute a_i and a_i is in the intent of a pattern c, then to evaluate the degree to which a_i influences c, we must consider an attribute c' as close to c as possible, and yet in which a_i is replaced by another value $a_j, j \neq i$ of the variable a. c' will be $\overline{d \cup \{a_j\}}$ for some $d > c$ for which $a_i \notin \overline{d}$. In this way, we can answer the question of how the attribute a_i in the intent of a pattern influences the pattern's emergence.

We hope to apply the stimulating patterns practically as well. First, we will study precisely the scalability of the algorithm to larger datasets. We will apply stimulating patterns to datasets where "factor interplay" is high, and the effect

of any one attribute depends on other attributes. We plan experiments on larger databases. We will explore applications to the presentation and compression of emerging patterns, to classification, and to supervision of rule-finding algorithms.

In this article we have introduced the notion of stimulating patterns in the context of formal concept analysis. Using the lattice-theoretic framework, we have generalized some notions which are known to be valuable in data mining to a new notion, the stimulation of a pattern. This notion has some attractive theoretical properties. We have implemented an algorithm to compute stimulation throughout a dataset, and we have extracted from the stimulation matrix two types of information which are already of known interest in the data-mining community.

Acknowledgement. we thank the regional council of Basse-Normandie for financial support ("programme emergence").

References

1. Dong, G., Li, J.: Applications of Emerging Patterns for Microarray Gene Expression Data Analysis. In: Liu, L., Tamer Özsu, M. (eds.) Encyclopedia of Database Systems, vol. 107. Springer, Heidelberg (2009)
2. Dong, G., Li, J.: Efficient mining of emerging patterns: discovering trends and differences. In: KDD 1999: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 43–52. ACM, New York (1999)
3. Dong, G., Zhang, X., Wong, L., Li, J.: CAEP: Classification by Aggregating Emerging Patterns. In: Arikawa, S., Furukawa, K. (eds.) DS 1999. LNCS (LNAI), vol. 1721, pp. 30–42. Springer, Heidelberg (1999)
4. Fayyad, U.M., Irani, K.B.: The Attribute Selection Problem in Decision Tree Generation. In: AAAI, pp. 104–110 (1992)
5. Ganter, B., Stumme, G., Wille, R.: Formal Concept Analysis: Foundations and Applications. LNCS (LNAI). Springer, New York (2005)
6. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. In: Trans. C. Franzke. Springer, New York (1997)
7. Harrell Jr., Frank, E.: Regression Modeling Strategies. Springer, New York (2006)
8. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer Series in Statistics (2001)
9. Huang, H.-j., Qin, Y., Zhu, X., Zhang, J., Zhang, S.: Difference Detection Between Two Contrast Sets. In: Tjoa, A.M., Trujillo, J. (eds.) DaWaK 2006. LNCS, vol. 4081, pp. 481–490. Springer, Heidelberg (2006)
10. Li, J., Dong, G., Ramamohanarao, K.: Making use of the most expressive jumping emerging patterns for classification. Knowledge and Information Systems 3(2), 131–145 (2001)
11. Li, J., Wong, L.: Emerging patterns and gene expression data. Genome Informatics 12, 3–13 (2001)
12. Li, J., Yang, Q.: Strong Compound-Risk Factors: Efficient Discovery Through Emerging Patterns and Contrast Sets. IEEE Transactions on Information Technology in Biomedicine 5(11), 544–552 (2007)

13. Loekito, E., Bailey, J.: Using Highly Expressive Contrast Patterns for Classification - Is It Worthwhile? In: Theeramunkong, T., Kijsirikul, B., Cercone, N., Ho, T.-B. (eds.) PAKDD 2009. LNCS, vol. 5476, pp. 483–490. Springer, Heidelberg (2009)
14. Loekito, E., Bailey, J.: Mining influential attributes that capture class and group contrast behaviour. In: Shanahan, J.G., Amer-Yahia, S., Manolescu, I., Zhang, Y., Evans, D.A., Kolcz, A., Choi, K.-S., Chowdhury, A. (eds.) Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM, Napa CA, USA, pp. 971–980 (2008)
15. Loekito, E., Bailey, J.: Fast mining of high dimensional expressive contrast patterns using zero-suppressed binary decision diagrams. In: KDD 2006: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 307–316, 1-59593-339-5 (2006)
16. Poezevara, G., Cuissart, B., Crémilleux, B.: Discovering Emerging Graph Patterns from Chemicals. In: Rauch, J., Raś, Z.W., Berka, P., Elomaa, T. (eds.) Foundations of Intelligent Systems. LNCS, vol. 5722, pp. 45–55. Springer, Heidelberg (2009)
17. Ramamohanarao, K., Bailey, J., Fan, H.: Efficient Mining of Contrast Patterns and Their Applications to Classification. In: ICISIP 2005: Proceedings of the 2005 3rd International Conference on Intelligent Sensing and Information Processing, Washington, DC, USA, pp. 39–47, 0-7803-9588-3. IEEE Computer Society, Los Alamitos (2005)
18. Ramamohanarao, K., Fan, H.: Patterns Based Classifiers. In: World Wide Web, Hingham, MA, USA, vol. 1(10), pp. 71–83, 1386-145X. Kluwer Academic Publishers, Dordrecht (2007)
19. Ting, R.M.H., Bailey, J.: In: Ghosh, J., Lambert, D., Skillicorn, D.B., Srivastava, J. (eds.) Proceedings of the Sixth SIAM International Conference on Data Mining, SDM, Bethesda, MD, USA, April 20-22. SIAM, Philadelphia (2006)
20. Valtchev, P., Grosser, D., Roume, C., Hacene, M.R.: Galicia: An Open Platform for Lattices. In: Using Conceptual Structures: Contributions to the 11th Intl. Conference on Conceptual Structures, pp. 241–254 (2003)
21. Webb, G., Butler, S., Newlands, D.: On detecting differences between groups. In: KDD 2003: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 256–265. ACM, New York (2003)

Ontology-Based Understanding of Natural Language Queries Using Nested Conceptual Graphs

Tru H. Cao and Anh H. Mai

Faculty of Computer Science and Engineering
Ho Chi Minh City University of Technology, Vietnam
tru@cse.hcmut.edu.vn

Abstract. In a question answering system, users always prefer entering queries in natural language and not being constrained by a rigorous grammar. This paper proposes a syntax-free method for natural language query understanding that is robust to ill-formed questions. Nested conceptual graphs are defined as a formal target language to represent not only simple queries, but also connective, superlative, and counting queries. The method exploits knowledge of an ontology to recognize entities and determine their relations in a query. With smooth mapping to and from natural language, conceptual graphs simplify conversion rules from natural language queries and can be easily converted to other formal query languages. Experimental results of the method on the QA track datasets of TREC 2002 and TREC 2007 are presented and discussed.

1 Introduction

A natural language interface is always desirable for a question answering system ([12]). Using a controlled language for the interface could ease the problem but still tights users to a restricted grammar ([10], [11]). Although performance of machine natural language understanding for the general case appears to be saturated after many years of research, limiting the domain of discourse to only questions and querying phrases can make a difference. There are various approaches to conversion of natural language queries into more formal representations regarding the two following issues.

First, it is about whether rigorous syntactic parsing is applied to a query expression before it is mapped to a target language sentence. The disadvantages of the parsing approach are time consuming and requiring grammatically correct inputs, which is thus not robust to ill-formed queries. It is also not practical to require a user to always input a question without grammatical errors. Moreover, it may still face to the problem of syntactic ambiguity, i.e., one sentence having more than one applicable syntax tree.

Second, it is about whether a knowledge base (KB) is employed in the mapping. For example, with the query *"What county is Modesto, California in?"*, given no knowledge base, *Modesto* and *California* can be tagged only as proper

M. Croitoru, S. Ferré, and D. Lukose (Eds.): ICCS 2010, LNAI 6208, pp. 70–83, 2010.

nouns and thus the implicit relation expressed by the comma between them cannot be interpreted. In contrast, with a knowledge base, they can be recognized as named entities (NE) of the types city and province, respectively, whence the relation can be mapped to one being a sub-region of the other.

For instance, [9] implemented an ontology-based search system whose queries were lists of classes and instances and translated into expressions of SeRQL. They were better than lists of normal keywords, but not as natural as human expressions. Meanwhile, accepting natural language queries, [3] followed the rigorous parsing approach using lambda calculus as an intermediate formal language for the mapping. However, the focus of that work was on efficient porting interfaces between different domains rather on the mapping itself.

The approach in [7] could be considered as closer to the syntax-free one. It used pattern matching of a natural language query to subject-property-object triples in a knowledge base, before converting the query to one of SPARQL. For the example query therein *"What is a restaurant in San Francisco that serves good French food?"*, it first searched for those triples whose subjects, properties, and objects could match with *"restaurant"*, *"in"*, and *"San Francisco"*. That method thus could not produce a mapping if the KB did not contain such a triple for the named entity *San Francisco*, although it existed in the KB. We argue that the understanding step should not be mixed up with the answering step. That is, a query can have a mapping to a target language although there is no matched answer to it in a knowledge base of discourse.

Recently, [15] also followed the syntax-free approach to convert natural language queries into SeRQL expressions. It used the named entity recognition engine of GATE ([4]) and the PROTON ontology of KIM ([8]), but supplemented it with more entity types and relation types. The method was however just tested on the authors manually collected 36 questions.

Meanwhile [2] developed a method that did not rely on a strict grammar of querying sentences but did use an ontology and knowledge base for understanding natural language queries. Knowledge was provided not only for answering queries but also for their conceptual understanding, before they could be mapped to a target language. Conceptual graphs (CG [13]) were proposed as an intermediate language to convert queries to. The method was tested on the QA track datasets of TREC 2002 and TREC 2007 with hundreds of diverse questions.

Since the root of the difficulty of machine natural language understanding is the big gap between natural language and a machine executable one, using an intermediate language like SeRQL or CGs is a way to ease the problem. We choose CGs because they could be mapped smoothly to and from natural language, and used as an interlingua for conversion to and from other formal languages ([14]). Tim Berners-Lee, the inventor of the World Wide Web, concluded in [1] that CGs could be easily integrated with the Semantic Web. It was also shown in [16] that there was a close mapping between CGs and the RDF language.

There was research on automatic generation of CGs from natural language texts in a specific domain, e.g. the rule-based method in [6] and the machine

learning-based one in [17]. However, both of the works required syntactic parsing of input sentences and were evaluated mainly on semantic roles rather than whole sentences.

The accuracy achieved in the above-mentioned work [2] was only about 78.5% and 60.5% for the TREC 2002 and TREC 2007 datasets, respectively. Part of the reason is that the CG query language therein was only simple CGs and thus not expressive enough to represent queries with connective words, superlative adjectives, and of the type *"How many"*. In this paper, the CG query language is extended with nested CGs to handle those types of queries.

Next, Section 2 summarizes the basic notions of conceptual graphs, for the paper being self-contained, and defines nested CGs to represent connective, superlative, and counting queries. Section 3 presents in detail our proposed method for converting queries in English into conceptual graphs. Section 4 evaluates the performance of the method with experimental results. Finally, Section 5 concludes the paper with some remarks.

2 Nested Conceptual Graphs

2.1 Basic Conceptual Graphs

A conceptual graph is a bipartite graph of *concept* vertices alternate with (conceptual) *relation* vertices, where edges connect relation vertices to concept vertices. For example, the CG in Figure 1 expresses the facts *"Cognac is a product. There is a province. France is a country. The province is a sub-region of France. Cognac is produced in the province."*, or briefly, *"Cognac is produced in a province in France"*.

Fig. 1. An example conceptual graph

In a textual format, concepts and relations can be respectively written in square and round brackets as follows:

[PRODUCT: *Cognac*] → (PRODUCEDIN) → [PROVINCE: *] → (SUBREGIONOF)
↓
[COUNTRY: *France*]

A CG can also be split into sub-graphs containing only one relation for each, using variable symbols to link coreferent concepts with the generic marker. For example, the above CG can be written as follows:

[PRODUCT: *Cognac*] → (PRODUCEDIN) → [PROVINCE: *x], and
[PROVINCE: *x] → (SUBREGIONOF) → [COUNTRY: *France*].

A query can be seen as expressing constraints in terms of relations between the queried entities and the known ones. Using CGs for question answering, besides individual referents and the generic referent "*", we extend them with the *queried*

referent denoted by "?", representing the named entities to be searched for. The generic referent in a query CG means that it does not care about a matched entity. Next is an extension with nested CGs of the CG query language defined in [2].

2.2 Representing Connective Queries

In the extended CG query language, a natural language connective like *"and"* or *"or"* in a query is represented by a meta-relation, labelled and or or respectively, connecting the nested query CGs that represent enclosed elementary queries. For example, Figure 2 is the query CG for *"What international leaders sent or gave congratulations?"*, where the dashed line is called a *coreference link* denoting that the two linked concepts refer to the same entity.

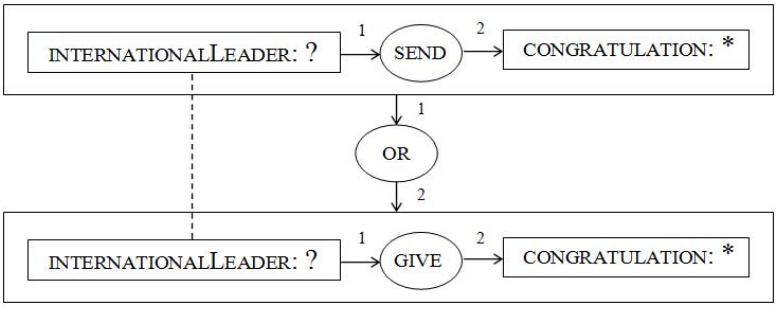


Fig. 2. The nested CG for a connective query

2.3 Representing Superlative Queries

There are queries that search for entities with the highest or lowest degree of a particular property, e.g. *"What's the tallest building in New York City?"*. For such a query, the trivial case is that the tallest building is explicitly identified in the knowledge base of discourse. However, it is often that only the height of each building is recorded. Therefore, answering such a query actually requires searching for all the entities that satisfy the constraints in the query, and then selecting the one with the highest or lowest degree of the specified property.

In the extended CG query language, such a query is represented by a meta-relation, labeled by MAX or MIN depending on the highest or lowest objective, attached to the nested CG representing the query constraints. Figure 3 illustrates the nested CG for this example query. The double line specifies the concept representing the property to be with the highest or lowest degree. It is similar to sub-queries using the aggregation MAX/MIN functions in traditional database languages like SQL.

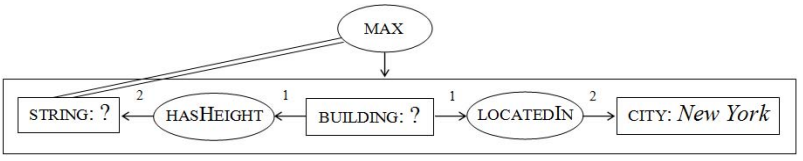


Fig. 3. The nested CG for a superlative query

2.4 Representing Counting Queries

Answering to a query *"How many"* may require counting the number of entities that satisfy the constraints specified in the query, when the sum is not trivially recorded in the knowledge base of discourse. In the extended CG query language, such a query is represented by a meta-relation, labeled by COUNT, attached to the nested CG representing the query constraints. Figure 4 illustrates the nested CG for the query *"How many languages has "Harry Potter and the Goblet of Fire" been translated to?"*. The double line specifies the concept representing the entities to be counted. It is similar to sub-queries using the aggregation COUNT function in SQL.

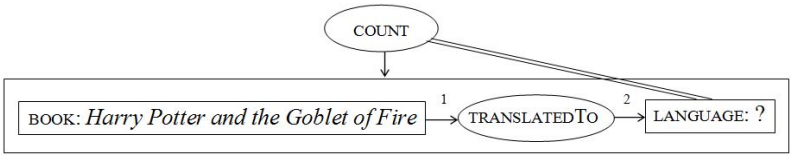


Fig. 4. The nested CG for a counting query

3 Generating Query Conceptual Graphs

Our method views a query as a sequence of entities and relations. The problem is then to determine which relation R links which entities E's, as illustrated in Figure 5. Valid relations between entities are actually constrained by an ontology of discourse. So the linking task does not depend much on the relative positions of the relations and entities in a query, and thus can accept ill-formed queries. Therefore, the main focus is only to correctly recognize entities and determine their relations expressed by a query. The method composes of the following twelve steps.

3.1 Recognizing Specified Entities

There are various tools with respective ontologies and KBs that can be used for NE recognition, such as GATE, KIM, SemTag ([5]), ESPotter ([18]). Obviously, the performance of any system relying on named entities to solve a particular problem incurs errors of the NE recognition tool employed. However, in research for models or methods, the two problems should be separated. This work is not about NE recognition and we use GATEs semantic annotation tool OCAT and KIMs ontology PROTON and KB for experiments.

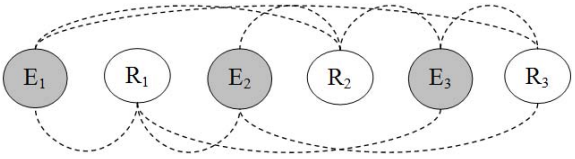


Fig. 5. A syntax-free view of a natural language query

3.2 Recognizing Unspecified Entities

Unspecified entities are those that are not expressed by names in a query. However, they are represented by phrases implying entity types and thus can be recognized. We employ the ANNIE tool of GATE for this task by building a gazetteer of phrases and their corresponding entity types in the ontology of discourse.

3.3 Extracting Relational Phrases

Words or phrases expressing relations between entities are propositional and verbal ones like *"in"*, *"on"*, *"of"*, *"born"*, *"has"*, ''is", ''located in", etc. They can also be extracted by ANNIE based on a gazetteer of phrases and their possible corresponding relation types in the ontology of discourse. For example, *"publish"* in a question can be mapped to the relation type DATEPUBLISH or HASPUBLISHER , and the suitable one depends on whether the question is about time (e.g. *"When was the first Wall Street Journal published?"*) or not (e.g. *"What company published Jasper Fforde's first book?"*).

3.4 Extracting Adjectives

The adjective associated with the noun representing an unspecified entity in a query describes either a property of the entity or a more specific type for that entity than the type expressed by the noun. For example, in the query *"What famous model was married to Billy Joel?"*, the queried entity can be assigned to the type FAMOUS_MODEL that is a subtype of the type MODEL. Meanwhile, in the query *"What is the longest suspension bridge in the U.S.?"*, the superlative adjective *"longest"* describes the property HASLENGTH of the queried suspension bridge. We build a gazetteer of adjectives and their corresponding properties or entity types, and also employ ANNIE to extract them from a query.

3.5 Splitting a Connective Query

In the spirit of the syntax-free approach, our assumption here is that users do not pose too complicated queries that are even obscure to human understanding. In particular, normally users do not use natural language connectives like *"and"* and *"or"* with arbitrary possible structures as for logical connectives. So, in our method, splitting a connective query is based on the normal patterns and their elementary queries shown in Table 1, where E_i's and R_j's represent entities and relations, respectively.

Table 1. Normal connective query patterns and their elementary queries

Query Pattern	Elementary Queries
$E_0R_1E_1$ *and/or* R_2E_2	$E_0R_1E_1$ *and/or* $E_0R_2E_2$
E_1 *and/or* $E_2R_0E_3$	$E_1R_0E_3$ *and/or* $E_2R_0E_3$
$E_1R_0E_2$ *and/or* E_3	$E_1R_0E_2$ *and/or* $E_1R_0E_3$
E_1R_1 *and/or* R_2E_2	$E_1R_1E_2$ *and/or* $E_1R_2E_2$

The remaining steps below are applied to each elementary query.

3.6 Determining the Type of Queried Entities

The type of the entity represented by the interrogative word *What* (or *Which*) is determined by the following rules:

1. If *What* is followed by an entity type, then the type of the queried entity is that entity type. For example, in the query *"What province is Montreal in?"*, the word *"province"* specifies that the type of the queried entity is province in the ontology of discourse.

2. Otherwise, the type is determined by the first NE after *What* and the relational phrase at the end of the query. For example, in the query *"What does Knight Ridder publish?"*, *Knight Ridder* is recognized as a company and the word *"publish"* entails that the queried entity is of the type PUBLISHEDMATERIAL.

The interrogative word *Who* may represent either a person or an organization. For example, in the query *"Who wrote the book Huckleberry Finn?"*, it represents a person. However, in the query *"Who provides telephone service in Orange County, California?"*, it means an organization. The appropriate entity type is determined on the basis of the involved relational phrases (e.g. *"wrote"* or *"provides"* in these examples) and the types of the entities after them (e.g. the book *"Huckleberry Finn"* or the service *"telephone"*).

Questions with the interrogative word *How* has three typical patterns:

1. The first one is with an adjective to ask about a certain property of an entity. An example query of this pattern is *"How tall is the Sears Building?"*. Values of such properties are often represented by strings of the type STRING in an ontology like PROTON. In this example, the adjective is mapped to the corresponding property type HASHEIGHT.

2. The second pattern is *"How much"* followed by an entity type, e.g. *"How much oil was spilled by the Exxon tanker Valdez?"*, or with *"cost"* at the end of the query, e.g. *"How much does an American Girl doll cost?"*. For the first case, *"How much oil"* is mapped to the oil spilling property and, for the second case, *"How much"* is mapped to the cost property of the mentioned entity.

3. The third pattern is *"How many"* followed by an entity type, e.g. *"How many counties are in Indiana?"*. Such a query is mapped to a nested CG with the meta-relation COUNT as presented in Section 2. One exception is queries asking about the population of a country, e.g. *"How many people live in Chile?"*, which is mapped to the property type POPULATIONCOUNT.

Time is also often represented by strings in data and knowledge bases. So, the interrogative word *When* in a query is mapped to a concept of the type STRING. For example, the signature of the relation type ESTABLISHMENTDATE is (ORGANIZATION, STRING).

3.7 Unifying Identical Entities

Two entities are considered as identical and unified under the following conditions:

1. One of them is an unspecified entity, and
2. The type of the unspecified entity is the same as, or a super-type of, the other entity, and
3. Between the two entities is the verb *"be"* in a particular form and tense such as *"is"*, *"are"*, *"was"*, *"were"*, etc.

For example, in the query *"Who is the president of Bolivia?"*, *Who* represents an unspecified entity of the type PERSON and *"president"* represents an entity of the type PRESIDENT, which is a subtype of PERSON. There is the relational word *"is"* between the two entities, so they are identical and can be unified.

3.8 Discovering Implicit Relations

If two entities are next to each other or separated by a comma, then there is an implicit relation between them. That relation is determined by the types of the entities and the relation types permitted for those two entity types in the ontology of discourse. For example, in the query *"In which US states has Barack Obama lived?"*, the type of US is COUNTRY and that of the unspecified entities represented by *"states"* is PROVINCE. Therefore, the appropriate type of the implicit relation between them is SUBREGIONOF.

3.9 Determining the Types of Relations

After the previous steps, the specified entities, unspecified entities, and relational phrases in a query are already recognized. The remaining task is to determine which relational phrase is between which two of the entities and what is the type of that relation. First, we present our approach to determine the appropriate relation type for a certain relational phrase in a query, with respect to the ontology of discourse. Let P_R be the relational phrase representing the relation between two entities of the types C_1 and C_2, and S_1 and S_2 be the original strings representing the two entities. We define the following sets of possible relation types:

1. R_1 is the set of possible relation types that correspond to P_R in the built-in gazetteer of relational phrases. For example, if P_R = *"publish"*, then R_1 includes DATEPUBLISH and HASPUBLISHER.
2. R_2 is the set of possible relation types between the entity types C_1 and C_2 as given in the ontology of discourse. For example, if C_1 = ORGANIZATION and C_2 = PERSON, then R_2 includes HASEMPLOYEE and HASFOUNDER.

3. R_3 is the set of possible relation types with respect to S_1 and P_R. For example, in the query *"Who is the founder of the Wal-Mart stores?"*, S_1 = *"founder"* and P_R = *"of"*, which derives hasFounder as a possible relation type between *Wal-Mart stores* and the queried entity.

4. R_4 is the set of possible relation types with respect to P_R and S_2. For example, in the query *"Who was Charles Lindberghs wife?"*, P_R = *"'s"* and S_2 = *"wife"*, which derives hasWife as a possible relation type between *Charles Lindbergh* and the queried entity.

The suitable relation types are then constrained within $R_1 \cap R_2 \cap R_3 \cap R_4$. For efficiency, we incorporate and encode all of these constraints into rules mapping relational phrases to suitable relation types in the ontology of discourse.

Second, we note that the phrase representing the relation between two entities can stand in different positions relative to those of the entities:

1. *In the middle*: for example, in the query *"Where is the location of the Orange Bowl?"*, the relational word *"of"* is in the middle of the two entities represented by *"location"* and *"Orange Bowl"*.

2. *After*: for example, in the query *"What state is the Filenes store located in?"*, the relational word *"located in"* is after the second entity represented by *"Filenes store"*.

3. *Before*: for example, in the query *"In what country is Angkor Wat?"*, the relational word *"in"* is before the first entity represented by *"country"*.

Therefore, for each pair of entities in a query, it is first checked if the relational phrase in the middle forms a proper relation between the two entities. If not, the relational phrases after and before the two entities are further checked.

3.10 Removing Improper Relations

Let E_1, E_2, , and E_N be the entities occurring in the left-to-right order in a query. We propose the following heuristic rules to remove improper relations extracted in the previous steps:

1. If E_i and E_{i+1} $(1 \leq i \leq N-1)$ are next to each other, then E_i has only a relation with E_{i+1}, and all relations if assigned to E_i and other entities will be removed. For example, in the query *"In which US states has Barack Obama lived?"* (E_1 = *"US"*, E_2 = *"states"*, E_3 = *"Barack Obama"*), there are three following possible relations extracted in the previous steps:

[province: ?x] → (subRegionOf) → [country: *US*], and
[person: *Barack Obama*] → (liveIn) → [province: ?x], and
[person: *Barack Obama*] → (liveIn) → [country: *US*],

but the last one is to be removed.

2. If E_i and E_{i+1} $(1 \leq i \leq N-1)$ are separated by a comma, then E_{i+1} has only a relation with E_i, and all relations if assigned to E_{i+1} and other entities will be removed. For example, in the query *"Who provides telephone service in Orange County, California?"* (E_1 = *"Who"*, E_2 = *"telephone service"*,

E_3 = *"Orange County"*, E_4 = *"California"*), there are four following possible relations extracted in the previous steps:

[COUNTY: *Orange*] → (SUBREGIONOF) → [PROVINCE: *California*], and
[TELEPHONESERVICE: *x] → (HASPROVIDER) → [COMPANY: ?], and
[TELEPHONESERVICE: *x] → (LOCATEDIN) → [COUNTY: *Orange*], and
[TELEPHONESERVICE: *x] → (LOCATEDIN) → [PROVINCE: *California*],

but the last one is to be removed.

3. If there is the relational symbol *"'s"* between E_i and E_{i+1} ($1 \leq i \leq N-1$), then E_i has only a relation with E_{i+1}, and all relations if assigned to E_i and other entities will be removed. For example, in the query *"What is the name of Neil Armstrong's wife?"* (E_1 = *"name"*, E_2 = *"Neil Armstrong"*, E_3 = *"wife"*), there are three following possible relations extracted in the previous steps:

[MAN: *Armstrong*] → (HASWIFE) → [WOMAN: *x], and
[WOMAN: *x] → (HASALIAS) → [ALIAS: ?y], and
[MAN: *Armstrong*] → (HASALIAS) → [ALIAS: ?y],

but the last one is to be removed.

4. If an entity is assigned relations to more than one entity standing before it, then only the relation with the nearest unspecified entity is retained. For example, in the query *"What city in Florida is Sea World in?"* (E_1 = *"city"*, E_2 = *"Florida"*, E_3 = *"Sea World"*), there are three following possible relations extracted in the previous steps:

[CITY: ?x] → (SUBREGIONOF) → [PROVINCE: *Florida*], and
[COMPANY: *Sea World*] → (LOCATEDIN) → [CITY: ?x], and
[COMPANY: *Sea World*] → (LOCATEDIN) → [PROVINCE: *Florida*].

However, since the entity Florida is already identified, the entity *Sea World* actually modifies the identity of the queried city, rather than *Florida*. Therefore, the last relation above is redundant and to be removed.

3.11 Modifying Concepts with Adjectives

An adjective modifying an entity can be in either of the following positions:

1. Right before the entity, or
2. After the entity and with the verb *"be"* in between.

An example of the first case is the query *"What is the longest suspension bridge in the U.S.?"*, and one of the second case is *"Name a tiger that is extinct?"*.

With a quantitative superlative adjective, the modified entity is represented by a nested CG with the corresponding meta-relation MAX or MIN, as presented in Section 2. Meanwhile, with the other adjectives, it is represented by a concept whose type is the corresponding subtype of the original entity type, e.g. [EXTINCT_TIGER: *] here.

4 Evaluation Experiments

We have tested the proposed method on the QA datasets of TREC 2002 and TREC 2007 with 440 and 446 queries, respectively. The test uses the PROTON ontology with about 300 entity types, 100 relation and property types, and KIM World KB with over 77,000 named entities. The correctness of each generated CG is justified by humans, with respect to the employed ontology and KB and the actual meaning of the corresponding query in natural language.

Translation errors may occur due to one of the following causes:

1. The employed NE recognition engine like GATEs does not recognize all the named entities in a query precisely and completely. We call this an R-*error*.
2. The ontology and KB of discourse lack certain entity types, relation types, or named entities mentioned in a query. We call this an O-*error*.
3. The current CG query language is not expressive enough to represent certain queries. We call this a Q-*error*.
4. The proposed method itself does not generate a correct CG. We call this an M-*error*.

On the other hand, queries are classified into those with *"How many"*, adjectives, superlative adjectives, connectives, or the others. In order to test the actual accuracy of the proposed translation method alone, we have then manually corrected the wrongly recognized NEs due to GATE, and supplemented PROTON and KIM KB with missing entity types, relation types, and named entities with respect to the testing queries.

Table 2 shows the number and percentage of each error type on the TREC 2002 dataset by our method, resulting in the overall accuracy of 87.5%. Table 3 presents the results on the TREC 2007 dataset with the overall accuracy of 73.99%. There are more O-errors and Q-errors on TREC 2007 as compared with TREC 2002. However, the translation method itself is still robust with only a few M-errors. If not counting queries with O-errors and Q-errors, then the translation accuracies are $385/(385+9) = 97.72\%$ and $330/(330+11) = 96.77\%$ for the TREC 2002 and TREC 2007 datasets, respectively.

On the basis of the experimental results, we now analyse and discuss on the above mentioned four types of translation errors and how they can be overcome. Firstly, R-errors solely depend on the accuracy of an employed NE recognition engine, whose improvement is a separate problem. Whereas, the proposed method is robust to the test datasets, so the small number of M-errors is not of primary concern now. The others, O-errors and Q-errors, are addressed below.

Non-binary relations
In practice, there are relations with arities greater than two. An example is the query *"What year did the U.S. buy Alaska?"*, where *"buy"* actually is a 3-ary relation of *U.S.*, *Alaska*, and the queried year. However, in ontology and KB languages, such as RDF and OWL, only binary relations are directly supported. All the encountered O-errors with TREC 2002 and TREC 2007 are due to non-binary relation types, which are not modelled in the used ontology. In order to

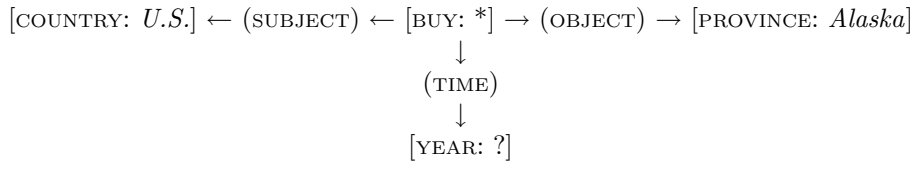
Table 2. Performance of the proposed method on TREC 2002

Query Type	**Number of Queries**	**Correct CGs**	***M*-errors**	***O*-errors**	***Q*-errors**
How many	17	8	5	3	1
with adjectives	6	6	0	0	0
with superlative adjectives	35	14	0	21	0
with conjunctions	1	1	0	0	0
others	381	356	4	20	1
Total	**440 (100%)**	**385 (87.5%)**	**9 (2.05%)**	**44 (10%)**	**2 (0.45%)**

Table 3. Performance of the proposed method on TREC 2007

Query Type	**Number of Queries**	**Correct CGs**	***M*-errors**	***O*-errors**	***Q*-errors**
How many	63	27	8	16	12
with adjectives	5	4	0	1	0
with superlative adjectives	22	6	0	16	0
with conjunctions	8	4	0	1	3
others	348	289	3	56	0
Total	**446 (100%)**	**330 (73.99%)**	**11 (2.47%)**	**90 (20.18%)**	**15 (3.36%)**

represent an n-ary relation, one way is to define a reified relation type, which is an entity type that has n binary relation types with n entity types of that relation[1]. Then, for instance, this example query can be represented by the following query CG:



Correspondingly, our proposed method needs to be extended to recognize if a relation in a query is reified or not.

Queries about relations

Among 2 Q-errors with TREC 2002 and 15 Q-errors with TREC 2007, there is one query in each dataset about relations instead of entities. For example, one of the queries is *"How was Teddy Roosevelt related to FDR?"*, where *FDR* stands for *Franklin D. Roosevelt*. If this query were converted into a CG, the question mark would be in a relation node instead of a concept node as below:

[PERSON: *Teddy Roosevelt*] → (?) → [PERSON: *FDR*]

[1] http://www.w3.org/TR/swbp-n-aryRelations/

So in order to deal with queries about relations, the CG query language needs to be extended with queried relations, and the translation method improved accordingly.

Temporal and other complicated queries
There is also one Q-error in each dataset for queries about time. For example, one of the queries is *"At Christmas time, what is the traditional thing to do under the mistletoe?"*. For representing such a temporal query, one would need to extend the CG query language with meta-relations about time to attach to a nested CG. Besides, in TREC 2007, there are 2 context-dependent queries, such as *"What was the previous world record time?"*, whose representation requires knowing the current world record time as the reference for the previous one.

5 Conclusion

We have presented a nested CG language for formal representations of natural language queries. The language is equipped with meta-relations and can represent connective, superlative, and counting queries. With smooth mapping to and from natural language, conceptual graphs simplify the rules to convert natural language queries to them. As an interlingua, conceptual graphs can also be further converted to other formal query languages.

Our proposed method of mapping natural language queries to conceptual graphs does not require grammatically correct querying sentences and exploits an ontology to identify entities and their respective relations in a query. Since the ontology constraints valid relation types between certain entity types, it makes the method robust to ill-formed queries, not too dependent on relative positions of relations and entities.

The experimental statistics show that the proposed method is robust to diverse structures and contents of questions in the test datasets, provided that the ontology and knowledge base of discourse cover well entities and relations in the domain. Still, as analysed above, to handle more query patterns, the ontology needs to be enriched to support n-ary relations. The constructed translation rules then need to be revised to recognize relations that are reified in a query. We are doing research along these lines.

References

1. Berners-Lee, T.: Conceptual Graphs and the Semantic Web, `http://www.w3.org/DesignIssues/CG.html` (Initially created: January 2001, Last change: April 2008)
2. Cao, T.H., Cao, T.D., Tran, T.L.: A Robust Ontology-Based Method for Translating Natural Language Queries to Conceptual Graphs. In: Domingue, J., Anutariya, C. (eds.) ASWC 2008. LNCS, vol. 5367, pp. 479–492. Springer, Heidelberg (2008)
3. Cimiano, P., Haase, P., Heizmann, J.: Porting Natural Language Interfaces between Domains An Experimental User Study with the ORAKEL System. In: Proceedings of the 12th ACM International Conference on Intelligent User Interfaces, pp. 180–189 (2007)

4. Cunningham, H., et al.: Developing Language Processing Components with GATE Version 3 (a User Guide). University of Sheffield (2006)
5. Dill, S., et al.: SemTag and Seeker: Bootstrapping the Semantic Web via Automated Semantic Annotation. In: Proceedings of the 12th International Conference on the World Wide Web, pp. 178–186 (2003)
6. Hensman, S., Dunnion, J.: Using Linguistic Resources to Construct Conceptual Graph Representation of Texts. In: Sojka, P., Kopeček, I., Pala, K. (eds.) TSD 2004. LNCS (LNAI), vol. 3206, pp. 81–88. Springer, Heidelberg (2004)
7. Kaufmann, E., Bernstein, A., Fischer, L.: NLP-Reduce: A "Nave" but Domain-Independent Natural Language Interface for Querying Ontologies. In: Demo-Paper at the 4th European Semantic Web Conference, pp. 1–2 (2007)
8. Kiryakov, A., Popov, B., Terziev, I., Manov, D., Ognyanoff, D.: Semantic Annotation, Indexing, and Retrieval. Journal of Web Semantics 2 (2005)
9. Lei, Y., Uren, V., Motta, E.: Semsearch: A Search Engine for the Semantic Web. In: Staab, S., Svátek, V. (eds.) EKAW 2006. LNCS (LNAI), vol. 4248, pp. 238–245. Springer, Heidelberg (2006)
10. Nelken, R., Francez, N.: Querying Temporal Databases Using Controlled Natural Language. In: Proceedings of the 18th Conference on Computational Linguistics, pp. 1076–1080 (2000)
11. Nyberg, E.H., Mitamura, T.: Controlled Language and Knowledge-Based Machine Translation: Principles and Practice. In: Proceedings of the 1st International Workshop on Controlled Language Applications, pp. 74–83 (1996)
12. Ogden, W.C., Bernick, P.: Using Natural Language Interfaces. In: Helander, M., Landauer, T.K., Prabhu, P. (eds.) Handbook of Human-Computer Interaction, pp. 137–162. Elsevier Science, Amsterdam (1997)
13. Sowa, J.F.: Conceptual Structures Information Processing in Mind and Machine. Addison-Wesley Publishing Company, Reading (1984)
14. Sowa, J.F.: Matching Logical Structure to Linguistic Structure. In: Houser, N., Roberts, D.D., Van Evra, J. (eds.) Studies in the Logic of Charles Sanders Peirce, pp. 418–444. Indiana University Press (1997)
15. Tablan, V., Damljanovic, D., Bontcheva, K.: A Natural Language Query Interface to Structured Information. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS (LNAI), vol. 5021, pp. 361–375. Springer, Heidelberg (2008)
16. Yao, H., Etzkorn, L.: Conversion from the Conceptual Graph (CG) Model to the Resource Description Framework (RDF) Model. In: Contributions of the 12th International Conference on Conceptual Structures, pp. 98–114 (2004)
17. Zhang, L., Yu, Y.: Learning to Generate CGs for Domain Specific Sentences. In: Delugach, H.S., Stumme, G. (eds.) ICCS 2001. LNCS (LNAI), vol. 2120, pp. 44–57. Springer, Heidelberg (2001)
18. Zhu, J., Uren, V., Motta, E.: ESpotter: Adaptive Named Entity Recognition for Web Browsing. In: Althoff, K.-D., Dengel, A.R., Bergmann, R., Nick, M., Roth-Berghofer, T.R. (eds.) WM 2005. LNCS (LNAI), vol. 3782, pp. 518–529. Springer, Heidelberg (2005)

An Easy Way of Expressing Conceptual Graph Queries from Keywords and Query Patterns

Catherine Comparot, Ollivier Haemmerlé, and Nathalie Hernandez

IRIT, Université de Toulouse le Mirail, Département de Mathématiques-Informatique, 5 allées Antonio Machado, F-31058 Toulouse Cedex
{catherine.comparot,ollivier.haemmerle,nathalie.hernandez}@univ-tlse2.fr

Abstract. Our goal is to hide the complexity of formulating a query expressed in a graph query language such as conceptual graphs. We propose a mechanism for querying a collection of documents previously annotated by means of conceptual graphs. Our queries are based on keywords given by the user. Our system associates the keywords with their corresponding concepts. Then it selects a set of patterns which are pre-written typical queries. The patterns are instantiated according to the initial keywords. Finally, the user can modify the pre-written query he/she selects in order to fit his/her initial needs as well as possible. The query graph obtained is projected into the annotation graphs associated with the documents. The documents which answer the query are returned to the user.

1 Introduction

Even if interrogation languages based on graphs such as conceptual graphs or, more recently, SPARQL, have been presented as a natural and intuitive way of expressing information needs, end-users do not think their queries in terms of graphs as they usually do not know these kinds of languages. End-users need simple languages that are mainly limited to keywords as it is the way they are all used to expressing their queries on the current Web. Moreover, in the framework of real applications, we often observe that the queries expressed by the users are rarely original. They tend rather to be variations around a few typical query families.

In this paper, we propose an approach in which end-users formulate their information needs by means of keywords which are automatically transformed into semantic graph queries according to typical query patterns. Our concern is both to solicit as little effort as possible from the user and to generate the query efficiently.

Our team works on automatic annotation and querying of documents. For these purposes, we use the conceptual graph model. The application presented in the paper focuses on the interrogation of annotations expressed in the CG model. Annotations are generated according to an extension of the work presented in [1]. This work has been carried out in the framework of the WebContent project [2] which consists in creating a software platform to accommodate the tools necessary to efficiently exploit and extend the Semantic Web. The main objective

M. Croitoru, S. Ferré, and D. Lukose (Eds.): ICCS 2010, LNAI 6208, pp. 84–96, 2010.

is to produce a flexible and generic platform for content management and to integrate Semantic Web technologies in order to show their effectiveness on real applications with strong economic or societal stakes. Here, we focus on economic watch in aeronautics. The input of our system is news releases which are provided by news agencies. Each of these releases has been annotated. This paper presents the interrogation mechanism we have proposed to query these annotations.

Section 2 presents related works concerning the construction of graph queries from keywords. Then we give an overview of our approach and present our ontology and the notion of pattern in section 3. Section 4 describes step by step the building and the processing of a query. The conclusion presents briefly the implementation of this work, the work in progress and some perspectives.

2 Related Works

Several works have been done on the querying of conceptual graph bases. They generally focus on information retrieval techniques, but assume that a CG query is given by the user [3]. Outside the CG community, two kinds of approaches have been proposed for facilitating the querying of information annotated semantically. The first one consists in helping the user formulate his/her query in an interrogation language adapted to the formalisation used for representing the annotations. This approach is not always adapted to end-users: to write a query, a user needs to know the syntax of the language and the representation of the data managed by the system (schema of the data to query). Some GUI systems propose an alternative to this issue. They provide an interactive graphical editing environment that combines ontology navigation capabilities with graphical query visualization techniques. For example, the approach presented in [4] allows navigation in RDF/S class and property definitions and is able to generate an RQL query which captures the cumulative effect of a user navigation session. In [5] a graphical tool for SPARQL query construction is presented. In the same way, CoGui [6] can be used to help the user construct a CG expressing his/her query. Even if these kinds of GUI are useful for end-users, the user still needs time to get used to the GUI and to formulate his/her query by means of a graph language.

Other works, such as ours, aim at automatically generating formal queries from keywords. The user can then express his/her information need in an intuitive way without knowing the interrogation language or the knowledge representation used by the system. Approaches have been proposed for generating formal queries expressed in different languages such as SeREQL [7], SPARQL [8,9].

In these systems, the generation of the query requires the following steps:

- mapping the keywords to semantic entities defined in the knowledge representation,
- building query graphs linking the entities previously detected by exploring the knowledge representation,
- ranking the built queries,

– making the user select the right one (the selection is often facilitated by the presentation of a sentence in natural language expressing the meaning of the graph).

The existing approaches focus on three main issues : optimizing the first step by using external resources (such as WordNet or Wikipedia)[7,10], optimizing the knowledge representation exploration mechanism for building the query graphs [8,9], and enhancing the query ranking score [10]. Our approach differs from existing ones in the way that we propose to enhance the effectiveness and the efficiency of the query building step by using pre-defined query patterns. The use of patterns avoids exploring the ontology for linking the semantic entities identified from the keywords since potential relations are already expressed in the patterns. The process thus benefits from the pre-established families of frequently expressed queries for which we know that real information needs exist. The main issue is being able to select the most suitable pattern and to adapt it to the initial query.

3 Pattern-Based Interrogation

3.1 Overview

As presented in Fig. 1, our approach relies on different steps. First the user expresses his/her query by means of keywords. Then our system transforms the query into a conceptualised query which consists of a set of concepts. The system identifies the patterns which are semantically close to the conceptualised query by exhibiting mappings which correspond to potential final queries. The mappings are then ranked and the user is asked to choose one by reading the sentences expressing their meaning. The final query graph is generated automatically from the pattern and used for the retrieving process.

Thanks to the use of patterns, little effort is requested from the user. He/she only has to define the initial keywords and to read and select the corresponding sentence in natural language.

The two following subsections present respectively the ontology and the kind of patterns we use.

3.2 The Support

A knowledge base expressed in the conceptual graph model is built on a support which contains the terminological knowledge. We use an extension of the classic definition of a support [11,12] which allows synonym terms to be associated with individual markers and concept types.

A support is a 5-uple $S = (T_C, Syn_{TC}, T_R, M, Syn_M)$, T_C being the partially ordered set of concept types, Syn_{TC} a set of synonyms for the concept types of the topic, T_R the partially ordered set of relation types, M the set of individual markers, which are instances of concepts, Syn_M a set of synonyms for the individual markers. Fig. 2 shows a part of our concept type set.

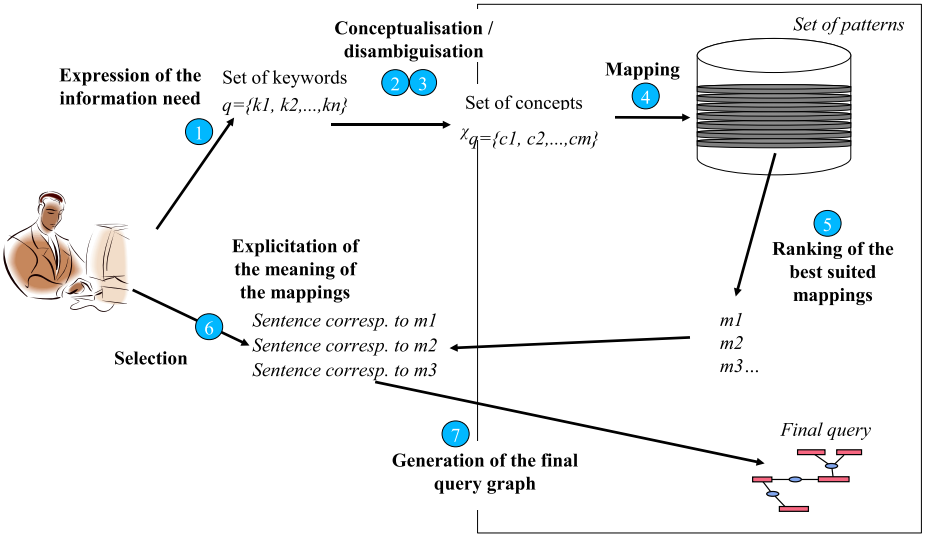


Fig. 1. Overview of our approach

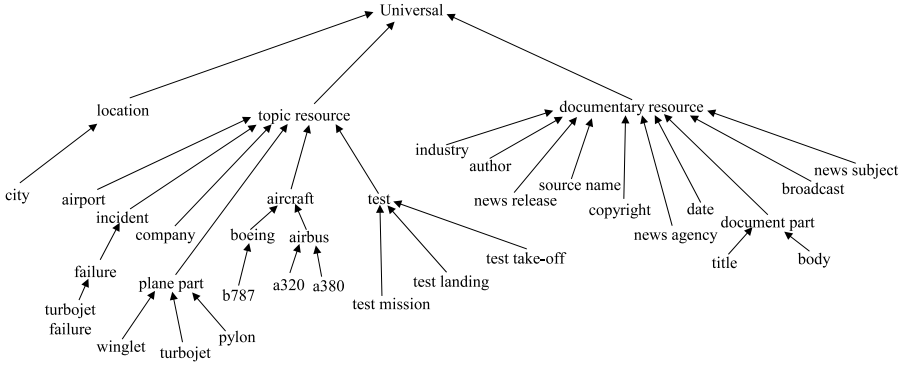


Fig. 2. A part of the concept type set of our application

In order to enable our process to generate a query graph from a set of keywords, we store for each concept type t a set of synonyms denoted $Syn_{TC}(t)$. For example we have $Syn_{TC}(A380) = \{$"A380","Airbus A380", "A380 aircraft", "Airbus superjumbo"$\}$.

The relation types belonging to T_R represent the nature of the links between concepts in the conceptual graphs: *located_in, involving, chartered_by, char, origin, dest, agt, ...* are relation types we use in our application.

The set of individual markers M contains the instances of concept types. It is partially defined *a priori* and is complemented progressively during the document annotation, for example by extracting the names of the author of a news release,

which is considered as an instance of the concept type *Author*. We store for each individual marker m known *a priori*, a set of synonyms denoted $Syn_M(m)$. For example we have $Syn_M(AFP) = \{$"AFP", "Agence France Presse"$\}$.

3.3 Pattern Definition

A pattern is composed of a conceptual graph which is the prototype of a relevant family of queries. Such a pattern is characterized by a subset of its concept vertices, called the *qualifying concept vertices*, which can be modified during the construction of the final query graph. It is also described by a sentence expressed in natural language in which a distinct substring must be associated with each qualifying concept vertex.

For now, the patterns are designed by experts who know the application domain and the general shape of the annotations of the documents. The CGs are built manually, the qualifying concept vertices are selected by the designer of the pattern who also gives the sentence describing its meaning.

Definition 1. *A task pattern p is a 3-uple $\{\mathcal{G}_p, \mathcal{C}_p, \mathcal{S}_p\}$ such that:*

- *$\mathcal{G}_p$ is a conceptual graph describing the pattern. Such a conceptual graph must be acyclic.*
- *$\mathcal{C}_p = \{c_1, c_2, \ldots c_n\}$ is a set of n distinct generic concept vertices belonging to $\mathcal{G}_p$, called the qualifying concepts of the pattern.*
- *$\mathcal{S}_p = \{s, (w_1, w_2, ...w_n)\}$ is a description of the meaning of the pattern in plain text (the sentence s) and a collection of n distinct substrings $w_1, w_2, ... w_n$ belonging to s. Note that for each $i \in 1, \ldots n$, w_i is called the term associated with the concept vertex c_i.*

Example 1. *In this article, we use in our examples 2 patterns, p_1 and p_2. Fig. 3 and Fig. 4 respectively present the conceptual graph associated with p_1 and p_2.*

$\mathcal{C}_{p_1} = \{c_{11}, c_{12}, c_{13}, c_{14}\}$. $\mathcal{S}_{p_1}$ is the sentence "A test mission$_{w_{11}}$ operated on a date$_{w_{12}}$ with an Airbus$_{w_{13}}$ at an airport$_{w_{14}}$".

$\mathcal{C}_{p_2} = \{c_{21}, c_{22}, c_{23}, c_{24}\}$. $\mathcal{S}_{p_2}$ is the sentence "A turbojet failure$_{w_{21}}$ located in an airport$_{w_{22}}$ involving an Airbus$_{w_{23}}$ chartered by a company$_{w_{24}}$".

4 Construction of an Actual Query Graph

In our approach, the user interacts with the system by expressing a query consisting of a set of keywords (which are terms or phrases). The query is then

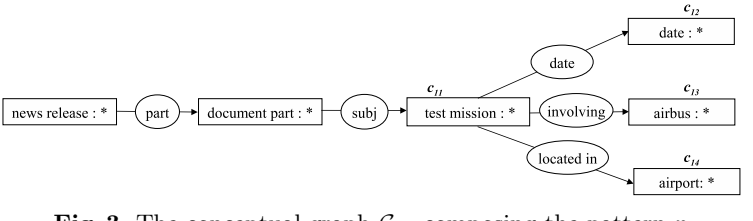


Fig. 3. The conceptual graph $\mathcal{G}_{p_1}$ composing the pattern p_1

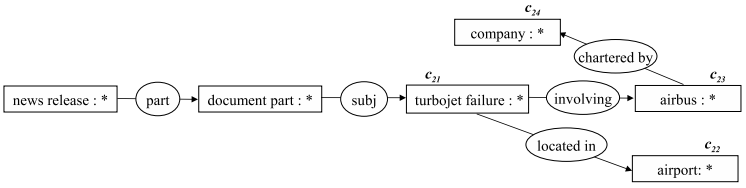


Fig. 4. The conceptual graph $\mathcal{G}_{p_2}$ composing the pattern p_2

conceptualised by translating the keywords into concept vertices which can be generic or individual. In the case of several possible conceptualisations of a user query, we propose to disambiguate the queries in order to obtain exactly one conceptualised query. That query is then mapped into the patterns of the system in order to exhibit possible relevant queries which are ranked with regard to a relevance degree. The user can select the query he/she wants to ask by reading its meaning expressed in natural language. Finally, the actual query graph is built by modifying the graph pattern with respect to the initial query of the user.

In this section, we present step by step the construction of a query graph based on the initial query asked by the user.

4.1 Step 1: Expression of a User Query

Definition 2. *A user query q is a set of strings which correspond to simple or composed keywords $q = \{k_1, k_2, \ldots k_m\}$.*

Example 2. *The following query q can be asked by a user interested in the incidents having occurred in Toulouse to the Airbus' Superjumbo reported by the Minister of transportation.*

$q = \{k_1 : \text{“}incident\text{”}, k_2 : \text{“}airbus\ superjumbo\text{”}, k_3 : \text{“}Toulouse\text{”}, k_4 : \text{“}minister\ of\ transportation\text{”}\}$

4.2 Step 2: Conceptualisation of the Query

The query conceptualisation step consists in identifying the individual marker and/or the concept type corresponding to each keyword of the initial query. This is done by looking for the keywords in the synonym set associated with each individual marker and each concept type. Note that several conceptualisations of a keyword can be made when it is ambiguous. It is also possible for the conceptualisation to fail if the term does not belong to any synonym set (in this case, the keyword is most probably outside the scope of the domain represented in the ontology).

After the conceptualisation step, each keyword k_i of the initial user query q is translated into a possibly empty set of concept vertices C_{k_i}. The C_{k_i} set contains the concept vertices corresponding to the keywords of the user query which have been identified in the ontology. The set is empty if k_i remains unrecognized.

Example 3. *The keywords of q could be conceptualised as follows:*
$C_{k_1} = \{[incident\ :\ *]\}$
$C_{k_2} = \{[a380\ :\ *]\}$
$C_{k_3} = \{[airport\ :\ Toulouse\_Blagnac], [city\ :\ Toulouse]\}$
$C_{k_4} = \emptyset$
The keyword "Toulouse" can be conceptualised as an instance of the concept type "City" (the city of Toulouse) or as an instance of the concept type "Airport" (the Toulouse-Blagnac airport). The keyword "Minister of transportation" has neither been recognized as a concept type nor as a marker of our ontology.

4.3 Step 3: Disambiguation of the Query

The following step of our process is the disambiguation step which consists in associating exactly one concept vertex corresponding to a keyword of the initial query when there is an ambiguity. This allows us to generate a conceptualised query. We limit the generation of exactly one conceptualised query since the cases of ambiguity are generally related to homonymy and, in such cases, the user is interested in exactly one meaning of the keyword.

In order to disambiguate the query, we propose to the user to choose the concept type and marker he/she wants to associate with his/her ambiguous keyword, by selecting the corresponding concept and marker labels in a list of potential meanings: the user query is disambiguated manually by the user him/herself. We have also implemented an automatic mechanism which is based on the computation of a semantic distance between the concepts involved in all the possible conceptualised queries generated from the conceptualised keywords. The distance is calculated according to the measure presented in [13]. Even if there is theoretically an exponential number of possible conceptualised queries, the number of ambiguities in a very small set of keywords is limited. However we will have to work on optimization when working on the scalability of our process.

Definition 3. *A conceptualised query $\chi_q = \{\mathcal{C}_q, \mathcal{K}_q\}$ of a user query q is composed of a set $\mathcal{C}_q$ of generic or individual concept vertices injectively mapped with keywords of q, and the set $\mathcal{K}_q$ corresponding to the keywords of q which are not images in the previous injective mapping.*

Example 4. *If we consider that the user is interested in the meaning "airport" instead of the meaning "city" of the keyword "Toulouse", we consider that the conceptualised query corresponding to q is:*
$\chi_q = \{\mathcal{C}_q : \{[incident\ :\ *], [a380\ :\ *], [airport\ :\ Toulouse\_Blagnac]\}, \mathcal{K}_q : \{\text{“}Minister\ of\ transportation''\}\}$

4.4 Step 4: Mapping between the Conceptualised Query and the Patterns

This step consists in comparing the conceptualised query obtained in the previous step to the available query patterns, in order to select the patterns which

could be used as a basis for the construction of the final query. We try to establish a relationship – called a c-mapping – between the concept vertices of the conceptualised query and the qualifying concept vertices of each pattern. We partition the set of concept vertices of the conceptualised query into 4 parts: the concepts which are mapped with a concept of the pattern which has a label which is respectively the same, or a generalisation, or a specialisation, and the fourth part which contains the concepts which are not mapped. The mapping defined in this subsection allows us to quantify the relevance degree of a pattern w.r.t. a query, but it also gives the way of modifying the pattern in order to finally obtain a query as close as possible to the user's wishes.

Definition 4. *Given a conceptualised query χ_q and a task pattern p,*

- *let $\tau_=$ be a function from $\mathcal{C}_q$ into $\mathcal{C}_p$ injective on its definition domain $\mathcal{C}_{q=}$, which maps concept vertices of $\mathcal{C}_q$ and $\mathcal{C}_p$ having exactly the same label;*
- *let $\tau_<$ be a function from $\mathcal{C}_q$ into $\mathcal{C}_p$ injective on its definition domain $\mathcal{C}_{q<}$, such that the concept vertices of $\mathcal{C}_q$ are specializations (in the narrow sense) of their images by $\tau_<$ in $\mathcal{C}_p$;*
- *let $\tau_>$ be a function from $\mathcal{C}_q$ into $\mathcal{C}_p$ injective on its definition domain $\mathcal{C}_{q>}$, such that: (i) the concept vertices of $\mathcal{C}_q$ are generalizations (in the narrow sense) of their images by $\tau_>$ in $\mathcal{C}_p$, and (ii) the generalization of a concept c in $\mathcal{C}_p$ by generalizing its label to the label of its antecedent $\tau_>^{-1}(c)$ is possible without violating the signature of its adjacent relation vertices;*
- *let $\mathcal{C}_{q<>} = \mathcal{C}_q \backslash (\mathcal{C}_{q=} \cup \mathcal{C}_{q<} \cup \mathcal{C}_{q>})$.*

$m = \{\tau_=, \tau_<, \tau_>\}$ is called a c-mapping between χ_q and p if the 4 possibly empty parts $\{\mathcal{C}_{q=}, \mathcal{C}_{q<}, \mathcal{C}_{q>}, \mathcal{C}_{q<>}\}$ form a partition of $\mathcal{C}_q$ such that $\mathcal{C}_{q<>}$ is minimal and $|\mathcal{C}_{q=}| + |\mathcal{C}_{q<}| + |\mathcal{C}_{q>}| \geq 1$.

For each pattern p, our system computes all the possible c-mappings between the conceptualised query and the considered pattern. Each c-mapping is candidate for generating a final query graph.

Example 5. *For readability reasons, the concepts are represented by their labels since there is no ambiguity in our example. There is one c-mapping $m_1 = \{\tau_=^1, \tau_<^1, \tau_>^1\}$ between the conceptualised query χ_q presented in section 4.3 and p_1:*

- $\tau_=^1 = \emptyset$
- $\tau_<^1([Airport : Toulouse\ Blagnac]) = [Airport : *]$
- $\tau_<^1([a380 : *]) = [Airbus : *]$
- $\tau_>^1 = \emptyset$

and there is one c-mapping $m_2 = \{\tau_=^2, \tau_<^2, \tau_>^2\}$ between χ_q and p_2:

- $\tau_=^2 = \emptyset$
- $\tau_<^2([Airport : Toulouse\ Blagnac]) = [Airport : *]$
- $\tau_<^2([a380 : *]) = [Airbus : *]$
- $\tau_>^2([Incident : *]) = [Turbojet\ failure : *]$

4.5 Step 5: Ranking the c-Mappings w.r.t. Their Relevance

In order to rank all the c-mappings resulting from the comparison of the conceptualised query to all the patterns, we compute a relevance degree based on the semantic proximity of the concept vertices of $\mathcal{C}_q$ and $\mathcal{C}_p$. The criteria we use for a given c-mapping are the following:

- the larger $|\mathcal{C}_{q=}|$, the more relevant is m;
- to a lesser extent, the larger $|\mathcal{C}_{q<}|$, the more relevant is m. If the query asks on a given concept, and the pattern asks on a generalization of that concept, the specialization of the pattern w.r.t. the query produces a relevant query;
- to a lesser extent, the larger $|\mathcal{C}_{q>}|$, the more relevant is m. If the query asks on a given concept, and the pattern asks on a specialization of that concept, the generalization of the pattern w.r.t. the query can produce a query which is not relevant even if it is syntactically correct;
- finally, the larger $p_{only} = \mathcal{C}_p \backslash (\tau_=(c_i) \cup \tau_<(c_i) \cup \tau_>(c_i))_{\forall c_i \in \mathcal{C}_q}$, the less relevant is m: we assume that the higher the number of concepts not mapped in p, the less relevant will be the mapping.

In order to take into account all these criteria in our ranking, we propose the following relevance degree for a given c-mapping m:

$relevance(m) = (|\mathcal{C}_{q=}| + 0.9 * |\mathcal{C}_{q<}| + 0.8 * |\mathcal{C}_{q>}|)/(1 + 0.2 * |p_{only}|)$

Note that the coefficients have been adjusted after an experimental procedure but a real experimentation will be necessary to enhance our relevance degree. For the moment, we do not integrate in our ranking a measure of the level of specialization/generalization [3] which is done. It could be useful to take such a measure into account to favor a "weak" generalization over a "high" specialization.

The most relevant c-mappings will be presented to the user, ranked according to the relevance degree.

Example 6. *With the c-mappings m_1 and m_2 computed previously, we have:*

- $relevance(m_1) = (2 * 0.9)/(1 + 0.2 * 2) = 1.29$
- $relevance(m_2) = ((2 * 0.9) + (1 * 0.8))/(1 + 0.2 * 2) = 1.86$

The query obtained by the c-mapping m_2 will be presented first to the user.

Note that we did a first experimentation on the movie domain. A panel of 16 people gave 160 keyword queries in French with their corresponding description in natural language. During the first step of the experimentation, two experts designed 13 patterns by analysing one third of the queries. They then considered the remaining queries and estimated that 73% of them are covered by at least one of the 13 patterns. This first result confirms our hypothesis that users tend to ask the same type of queries. We then evaluated the precision of our approach. For 88% of the queries, the generated query corresponding to the description in natural language is ranked among the three best queries. This first evaluation is encouraging.

4.6 Step 6: Generation of the Description Query Sentences

The sentences describing the final queries which can be generated from the c-mappings are presented to the user in order to allow him/her to choose his/her final query. For each c-mapping, a sentence is generated from the original sentence describing the involved pattern, by substituting the labels of the concepts of the conceptualised query to the words corresponding to images of the mappings $\tau_=$, $\tau_<$ and $\tau_>$.

Example 7. *In our example, the results are presented as follows:*

- *query 1: "*an incident *located in* airport:Toulouse Blagnac *involving* an a380 *chartered by [a company]. "minister of transportation" ignored."*
- *query 2: "[A test mission] operated on [a date] with* an a380 *at* airport:Toulouse Blagnac*. "minister of transportation" ignored."*

Assume the user selects the first query, which better fits his/her needs. By analyzing the sentence describing that query, the user is informed that the keyword "minister of transportation" is not taken into account in the proposed query. He/she also knows that the words in italics have been changed in the original pattern in order to fit the keywords he/she proposed. And, finally, he/she knows that the concepts written in square brackets belong to the pattern but were not filled in in the user query. Here, the concept corresponding to the company is not filled in. The user can leave the proposed query "as is" (we search for incidents occurring in Toulouse-Blagnac on an a380 chartered by any company), he/she can specify the company he/she is searching for, or he/she can remove the company criterion. Note that such a removal corresponds to the deletion of a part of the conceptual graph describing the query pattern, and it is possible under conditions which are detailed in the following paragraph.

4.7 Step 7: Construction of the Final Query Graph

The final query graph is simply built by replacing the concept vertices in the graph describing the pattern w.r.t. the injective mappings $\tau_<$ and $\tau_>$. In our example, [Airport:*] is specialised by [Airport:Toulouse Blagnac], [Airbus:*] is specialized by [a380:*], and [Turbojet failure:*] is generalised by [Incident:*], given the final query graph presented in Fig. 5.

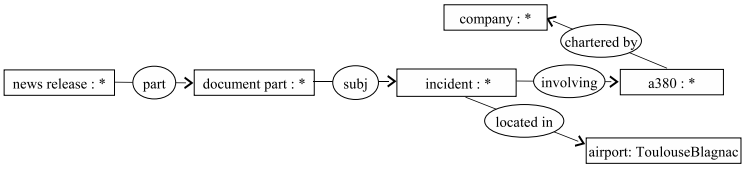


Fig. 5. The final query graph

In addition, in specific cases, the user can generalise or specialise his/her query:

Generalisation of a query by removing parts of the graph. The concepts belonging to p_{only} correspond to concepts that were not specified by the user but that belong to the pattern. We propose to the user to remove these concepts from the final query. In order to keep a coherent query graph, all the vertices belonging to p_{only} cannot be removed from the final query. Intuitively, we do not want to remove a concept vertex located on a path between two qualifying concepts of the query (it would disconnect concepts that were semantically linked in the task pattern graph).

The concept vertices of p_{only} the user wants to remove from the final query compose the set p^-. Our system computes $\mathcal{G}'_p$ the minimal connected subgraph of $\mathcal{G}_p$ which contains at least all the concept vertices of $\mathcal{C}_p \backslash p^-$. All the concept vertices belonging to that subgraph cannot be removed from the query graph even if they belong to p^-.

Finally, for the concepts belonging to p^- that can effectively be removed from the query graph, they are removed with all the vertices located on its branch until the first vertex belonging to $\mathcal{G}'_p$.

Specialization of the query by adding a concept. The concepts belonging to $\mathcal{C}_{q<>}$ correspond to concepts that are needed by the user but that have not been found in the pattern. We propose to the user to add these concepts to the final query. In the query graph, this corresponds to adding an isolated generic concept vertex labelled by the considered concept type to the query graph (which becomes non-connected). From a database point of view, it corresponds to adding a selection criterion on the query.

4.8 Step 8: Querying

The query is finally asked on every conceptual graph annotating the documentary knowledge base. We use the projection operation, and particularly the polynomial algorithm of projection of an acyclic conceptual graph [14] – all of our pattern conceptual graphs are acyclic, according to Def. 1. We return to the user the documents annotated by a conceptual graph which is more specific than his/her query.

5 Conclusion

This paper presents a way of bulding conceptual graph queries from a set of keywords. Such a simple expression of queries is better suited for end-users than query languages as SPARQL for example. The originality of our work is that the queries we build are based on query patterns which correspond to typical queries.

A prototype of our approach has been implemented using the Cogitant platform [15] and has been tested on a toy example defined from the WebContent [2] case study. This first stage of the work is really promising.

Compared to other approaches for building queries by searching semantic paths between the concepts expressed in a query, our system limits combinatorial issues by using predefined patterns: even if theoretically, there could be an exponential number of generated graph queries, in most cases, there is at most one mapping between a conceptualised query and a given pattern, and that mapping is computed trivially in polynomial time. Moreover, we use a polynomial algorithm of graph projection which ensures that the final query is run in reasonable time.

A next step will consist in working on the scalability of our approach. We are currently experimenting it on the entire WebContent case study. We will probably have to develop heuristics in order to search for the best c-mappings without generating all of them.

We will also work on the generation of the query patterns according to the queries frequently asked on a given corpus. In our first approach, we have experimented the generation of patterns by experts on two application domains: aeronautics, presented in this article, and movies. The results are encouraging since it is not that difficult for experts to generate relevant patterns from queries expressed in terms of keywords by end-users. It is obviously much easier to ask experts to generate patterns than to ask end-users to learn the CG model and the ontologies. The automatic generation of patterns is one of the issues we must address next.

Finally, we have just begun adaptating this work to semantic web languages such as SPARQL, in order to allow the querying of RDF annotations.

References

1. Comparot, C., Haemmerlé, O., Hernandez, N.: Conceptual graphs and ontologies for information retrieval. In: Priss, U., Polovina, S., Hill, R. (eds.) ICCS 2007. LNCS (LNAI), vol. 4604, pp. 480–483. Springer, Heidelberg (2007)
2. WebContent. The webcontent project. Web site (2007), `http://www.webcontent.fr`
3. Genest, D., Chein, M.: A content-search information retrieval process based on conceptual graphs. Knowl. Inf. Syst. 8(3), 292–309 (2005)
4. Athanasis, N., Christophides, V., Kotzinos, D.: Generating on the fly queries for the semantic web: The ics-forth graphical rql interface (grql). In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 486–501. Springer, Heidelberg (2004)
5. Russell, A., Smart, P.R.: Nitelight: A graphical editor for sparql queries. In: Bizer, C., Joshi, A. (eds.) International Semantic Web Conference (Posters & Demos). CEUR Workshop Proceedings, vol. 401 (2008), `CEUR-WS.org`
6. CoGui. A conceptual graph editor. Web site (2009), `http://www.lirmm.fr/cogui/`
7. Lei, Y., Uren, V.S., Motta, E.: Semsearch: A search engine for the semantic web. In: Staab, S., Svátek, V. (eds.) EKAW 2006. LNCS (LNAI), vol. 4248, pp. 238–245. Springer, Heidelberg (2006)
8. Zhou, Q., Wang, C., Xiong, M., Wang, H., Yu, Y.: Spark: Adapting keyword query to semantic search. In: Aberer, K., Choi, K.-S., Noy, N.F., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 694–707. Springer, Heidelberg (2007)

9. Tran, T., Wang, H., Rudolph, S., Cimiano, P.: Top-k exploration of query candidates for efficient keyword search on graph-shaped (rdf) data. In: ICDE, pp. 405–416. IEEE, Los Alamitos (2009)
10. Wang, H., Zhang, K., Liu, Q., Tran, T., Yu, Y.: Q2semantic: A lightweight keyword interface to semantic search. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 584–598. Springer, Heidelberg (2008)
11. Sowa, J.F.: Conceptual structures - Information processing in Mind and Machine. Addison-Wesley, Reading (1984)
12. Mugnier, M.-L., Chein, M.: Représenter des connaissances et raisonner avec des graphes. Revue d'Intelligence Artificielle 10(1), 7–56 (1996)
13. Rada, R., Mili, H., Bicknell, E., Blettner, M.: Development and application of a metric on semantic nets. IEEE Transactions on Systems Management and Cybernetics 19(1), 17–30 (1989)
14. Mugnier, M.-L., Chein, M.: Polynomial algorithms for projection and matching. In: Pfeiffer, H.D., Nagle, T.E. (eds.) Conceptual Structures: Theory and Implementation. LNCS (LNAI), vol. 754, pp. 239–251. Springer, Heidelberg (1993)
15. Genest, D.: Cogitant v-5.1 - manuel de référence (2003), `http://cogitant.sourceforge.net`

Natural Intelligence – Commonsense Question Answering with Conceptual Graphs

Fatih Mehmet Güler and Aysenur Birturk

Department of Computer Engineering, METU,
Inonu Bulvari, 06531, Ankara/Turkey
{e1285063,birturk}@ceng.metu.edu.tr

Abstract. Natural Intelligence (NI) is a question answering system based on Combinatory Categorial Grammar (CCG) as the theory of grammar and Conceptual Graphs (CG) for knowledge representation and reasoning. CCG is a lexicalized theory of grammar and very suitable for semantic analysis. Conceptual Graphs is a special kind of semantic network which can express full first-order logic. It aims to address the problem of commonsense reasoning in question answering, by using the state of the art tools such as C&C tools, Cogitant and Open Cyc. C&C tools are used for parsing natural language, Cogitant is used for Conceptual Graph operations, and Open Cyc is used for upper ontology and commonsense handling.

Keywords: Combinatory Categorial Grammar, Conceptual Graphs, Commonsense Reasoning, Open Cyc.

1 Introduction

The lack of commonsense often, required for reasoning to solve certain kinds of problems has been one of the most criticized aspects of expert systems [1]. There has not been adequate research to augment commonsense to expert systems, in spite of the existence of commonsense systems such as Cyc (`http://www.cyc.com`) which has been actively developed since 1984. On the other hand, Semantic Web (SW) is another outlook which aims to embed machine readability and computability to WWW. Although it is very promising for future web applications, it requires vast amount of manual work, and it doubles output formats resulting in increased effort. Besides, mass portion of web, in fact, knowledge in general, is represented in natural languages, and it appears to be so in foreseeable future. Therefore, in order to realize deeper natural language understanding and question answering, augmenting commonsense reasoning is required.

Phrase structure grammar is the traditional grammar formalism used for most natural language processing applications such as question answering, machine translation, and text summarization. However, meaning representation in logical form (LF) is difficult to extract from the output of such parsers, which are phrase structure trees in style of Penn Treebank [2]. Instead, because of its nature, Combinatory Categorial Grammar (CCG) is very suitable for the semantic

M. Croitoru, S. Ferré, and D. Lukose (Eds.): ICCS 2010, LNAI 6208, pp. 97–107, 2010.

composition task [3]. CCG is a lexicalized theory of grammar, and every CCG rule is coupled with a semantic interpretation. Providing compositional semantics for CCG simply consists of assigning semantic representations to lexical items and interpreting the combinatory rules. Moreover there are efficient and freely available parsers and tools available for these tasks such as C&C Tools [4].

First order logic, production rules, frames, and semantic networks are the widely used schemes for knowledge representation. In practice, mostly these schemes are used to define structures of the specific problem at hand, and a novel domain ontology is created to formalize the concepts, relations and rules. Since the ontology is specific to the domain, it is very hard, if not impossible, to scale the approach to other domains. It is possible, surely, to create a separate ontology for the new domain, but this requires the same effort to spend once more. Alternatively, a knowledge representation scheme, which is linguistically motivated from the initial design, could be better suitable for representing a diversity of concepts, not specific to a domain.

Conceptual Graphs (CG), introduced by Sowa, is developed as an intermediate language for mapping natural language questions and assertions to a relational database in the late 70's when semantic networks were very popular in computational linguistics [5]. What separates Conceptual Graphs from ordinary semantic networks is that CGs can express full first order logic. CGs are an ISO 24707 Common Logic (CL) dialect, which is a foundation for abstract model-theoretic logic-based notations. Along with the practical representation and reasoning phenomena, the Conceptual Graph community also has an interest in semiotics and epistemology, which fosters deeper natural language understanding research. Moreover, a graph based representation scheme has several advantages over linear notations such as computational efficiency in reasoning, searching, indexing and pattern matching [6]. These are the reasons for choosing Conceptual Graphs as the representation scheme of our project.

In this study, the main purpose is to address the the commonsense reasoning problem, getting closer to the ultimate goal of AI, Artificial General Intelligence (AGI) or strong AI. It is believed that the intelligence is the accumulation of knowledge, facts and rules. Moreover, since human like intelligence is based on natural language, the main theme of the study is linguistically motivated knowledge representation and reasoning, thus the name of the project is "Natural Intelligence". However, this is just a name, and we do not claim to attain AGI. In our system, natural language is parsed, utterances are represented using Conceptual Graphs, concepts and relations are mapped to Cyc equivalent commonsense counterparts, their type hierarchies are computed, and all of the knowledge is accumulated, in order to search for information and induce rules and beliefs. Whenever the input sentence is a question sentence, the system searches the missing information in the existing network. Commonsense knowledge is joined to the network by forward chaining as the sentences are entered by the user.

In the following sections, firstly we will describe the background information about the system, which includes the grammar theory and the tools used for natural language processing, the representation and reasoning scheme and tool

used for operations on it, and the commonsense system. Then we will explain our system, Natural Intelligence, in detail: how the commonsense is mapped, how the underlying tools are integrated, and how the queries are answered. Finally, we will discuss the significance of the implementation, concluding with our vision for future work.

2 Background

In this section, theories and tools about the system will be summarized. C&C tools are used to parse natural language sentences, which are based on Combinatory Categorial Grammar (CCG). Conceptual Graph (CG) notation is the representation and reasoning scheme used, suitable for natural language representation. Cogitant is the library used for CG operations. Open Cyc is the system used for commonsense ontology and reasoning.

2.1 Combinatory Categorial Grammar

Combinatory Categorial Grammar is a lexicalized theory of grammar based on categorial grammar [3]. Being lexicalized makes it an ideal framework for compositional semantics which simply involves adding semantic representations to the lexical entries and interpreting the small number of combinatory rules. Representing meaning using CCG is straightforward. In their paper Bos, Clark, Steedman, Curran and Hockenmeier [2] attempt to construct first order representations from CCG derivations using the λ-calculus and demonstrate that semantic representations can be produced for over 97% of the sentences in unseen Wall Street Journal text. Hockenmaier [7] derived a Treebank (CCGbank) of normal form CCG derivations semi-automatically from the Penn Treebank, and Clark & Curran [8] works on CCG Parsers, POS taggers and supertaggers.

2.2 C&C Tools

C&C is a wide coverage and high performance parser for CCG. C&C tools consist of a statistical parser and semantic representation generator [4]. C&C parser uses the grammar extracted from CCGbank, log linear parsing models, maximum entropy supertagger and POS tagger for a state of the art parsing. Semantic representation generator takes CCG derivation output from C&C parser and generates semantic representations in the form of Discourse Representation Structures (DRS). Figures 1 and 2 show samples of CCG derivation and semantic representation of C&C tools for the sentence "Susan knows that Bob likes Fred".

2.3 Conceptual Graphs

Conceptual graphs are a kind of assertional semantic networks and can represent first order logic [5]. Figure 3 shows the conceptual graph of the sentence "Mr. Hyde ate an apple in New York".

Similar to existential graphs of Peirce [9], rectangles represent concepts, circles are concept relations and arcs are the arguments of these relations.

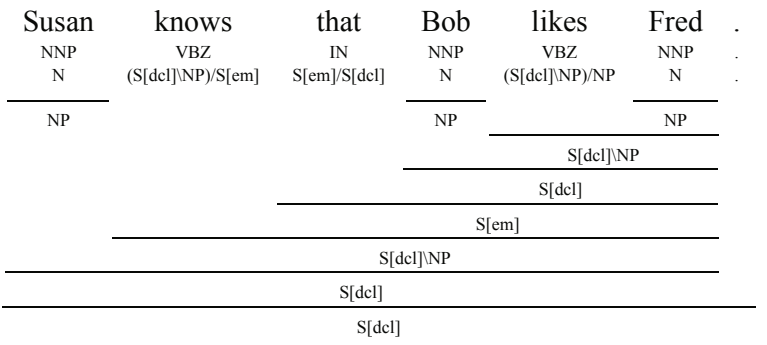


Fig. 1. Sample CCG derivation output of C&C parser

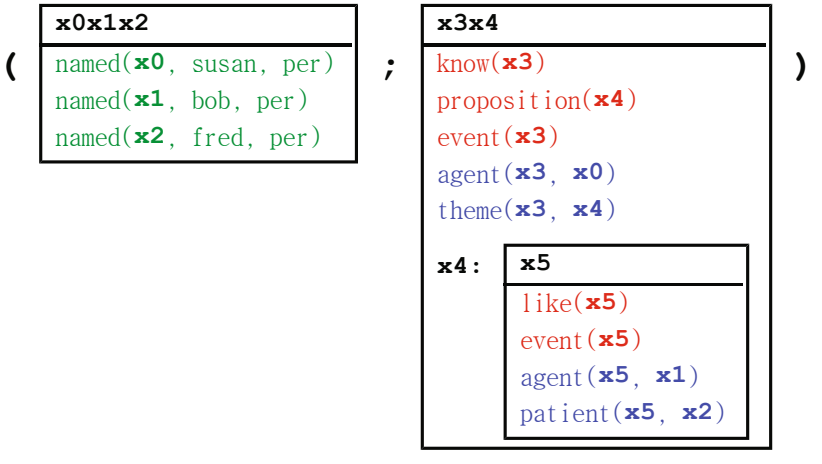


Fig. 2. Sample semantic representation

This example only uses conjunctions and existential quantifiers, but like Peirce's existential graphs, the conceptual graphs can also utilize scopes. The conceptual and existential graphs for the same sentence, "If a farmer owns a donkey, then he beats it" are given in Figure 4.

Common Logic (CL) is a foundation for abstract model theoretic logic based notations, originated from conceptual graphs and Knowledge Interchange Format. It is an ISO standard with name ISO/IEC 24707. Standard specifies three dialects which can express full CL semantics: Common Logic Interchange Format (CLIF), Conceptual Graph Interchange Format (CGIF), and XML-based notation for CL (XCL). Resource Description Framework (RDF) and Web Ontology Language (OWL) are also considered as subsets of CL. CGIF has two versions, core CGIF which is typeless, and extended CGIF which is typed and has extensions for *If, Then, Either, Or, Equivalence* and *Iff*.

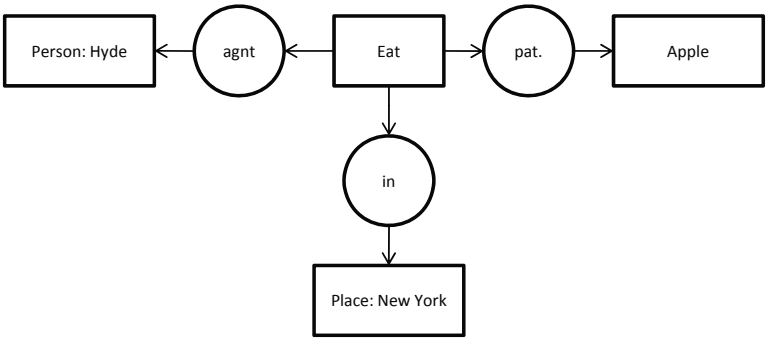


Fig. 3. Sample Conceptual Graph

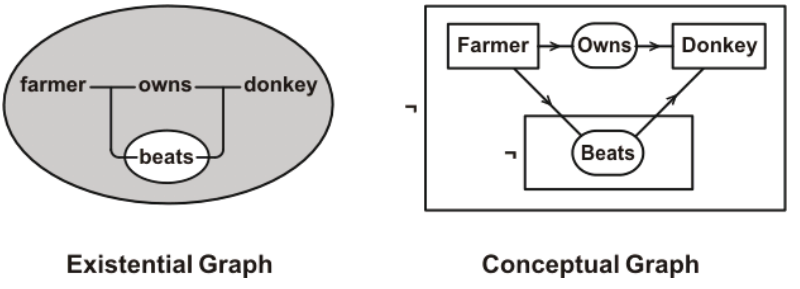


Fig. 4. The CG and EG for the sentence: "If a farmer owns a donkey, then he beats it"

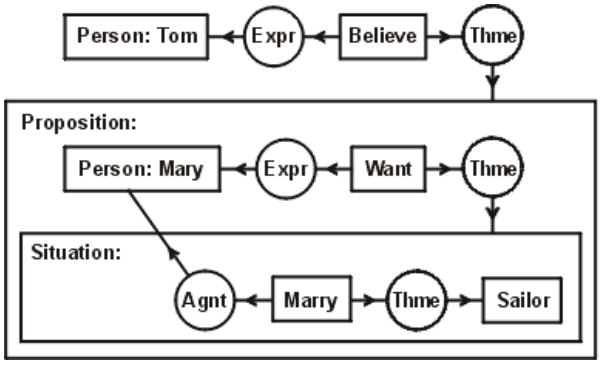


Fig. 5. Representation of the sentence "Tom believes that Mary wants to marry a sailor" using CG

The conceptual graph in Figure 3 can be written in CGIF as follows:

[Go *x] [Person: John] [City: Boston] [Bus *y]
(Agnt ?x John) (Dest ?x Boston) (Inst ?x ?y)

Here, the concepts are enclosed in square brackets and the relations are in parentheses. Variables starting with asterisk define nodes, and referencing variables start with question mark.

Conceptual graphs can be used to represent beliefs, intentions, and desires in natural language. For example the sentence "Tom believes that Mary wants to marry a sailor" can be represented as in Figure 5. The outer clause asserts that Tom has a belief which is Mary wants a situation. Belief and situation are defined in separate scopes and referenced by *believe* and *want* concepts. The sentence does not explicitly define where the sailor is though; it can be Mary's situation, Tom's proposition or in the outermost scope.

2.4 Cogitant Library

Our system uses Cogitant library for conversion from CGIF, applying commonsense rules and projection checking for query answering. Cogitant is a library which can handle Conceptual Graph operations [10]. These operations include basic graph representation, projection between graphs, rule application and constraint handling. If there is a mapping from the nodes of graph G to graph H preserving the relationships between entities there is a projection from G to H [11]. Rules can be defined using projection, such as; "if graph G exists then graph H must be asserted". The library can check whether there is a projection from the hypothesis graph of the rule to a graph and join the conclusion of the rule to the graph if there exists a projection.

2.5 Open Cyc

Open Cyc (`http://www.opencyc.org`) is the open source version of Cyc system which is a commonsense ontology and knowledge base. It aims to encode human commonsense knowledge in a machine usable form. It has been initiated by Douglas Lenat at Microelectronics and Computer Technology Corporation (MCC) in 1984. Today, Cyc has over 500,000 concepts, 26,000 relations and 5,000,000 assertions. Open Cyc, on the other hand, does not have the complex rules that are available in Cyc; instead it primarily consists of taxonomic assertions.

Assertions in Cyc are formulated in CycL language which is similar to Lisp programming language. Below is a sample assertion of the sentence: "George Bush likes Al Gore as a friend";

```
(#$likesAsFriend #$GeorgeWBush #$AlGore)
```

Predicates start with lower case, collections and individuals start with upper case. The mostly used predicates are #$isa and #$genls. #$isa means that the first argument is an individual and it is an instance of the second argument, which is a collection. #$genls means that the first argument is a subcollection of the second argument, where both arguments are collections. Below are sample assertions of #$isa and #$genls;

```
(#$isa #$GeorgeWBush #$UnitedStatesPresident)
(#$genls #$UnitedStatesPresident #$Person)
```

These assertions mean that the individual George Bush is an instance of the U.S. Presidents collection, and the U.S. President collection is a sub collection of the Person collection. Given these assertions, Cyc can conclude that George Bush is an instance of the Person collection.

Our system uses Open Cyc system to map parsed words to commonsense counterparts, such as concepts, relations and instances. Then the concept and relation hierarchy is queried from Cyc system, in order to generate support for the graph at hand. The details of this process will be further explained in the following sections.

3 Natural Intelligence

Natural Intelligence is developed as a web application which takes input as natural language, converts it to Conceptual Graphs, generates support from Open Cyc system, and whenever the input is a question sentence, tries to answer the question. The main goal of the system is to augment commonsense awareness in the process of question answering, by an existing broad commonsense knowledge base, such as Open Cyc. It also aims to design a modular infrastructure, utilizing separation of concerns, in order to employ implementation independence and to ease scalability for open domain, wide coverage question answering systems. Modules of Natural Intelligence are;

- Natural Language Processing (C&C Tools are used for implementation)
- Reasoning (Cogitant library is used for implementation)
- Commonsense (Open Cyc is used for implementation)
- Storage (Conceptual Graphs are stored in a database)

Natural Language Processing service is used to parse natural language and convert it to Conceptual Graph Interchange Format (CGIF). Reasoning service can convert CGIF to Conceptual Graphs (CGs), apply forward rules on CGs, resolve queries posed on CGs, join CGs, and use the concept hierarchy acquired from Commonsense ontology. Commonsense module, on the other hand, can find corresponding Cyc concepts of given words, in order to disambiguate the word sense, generate support for concepts and relations up to the root collection in Cyc (#$Thing). Finally, the storage module can persist CGs into database and retrieve CGs from database.

Natural Intelligence application uses these modules to create CGs from user input, augment with commonsense knowledge, merge the CGs, and answer questions based on the CGs. The system works as follows;

- User enters a sentence from web interface;
- This sentence is converted to CGIF using the NLP module;
- CGIF is converted to CGs using the reasoning module;
- Support is generated to CGs using the commonsense module;
- Commonsense rules gathered from commonsense module are applied to CGs using reasoning module;

- CGs are merged to the previous ones using reasoning module;
- If the input sentence is a question sentence, same operations take place, except the resulting graph is used to query existing CGs using the reasoning service, and if there are projections from this query graph to previous CGs, results are displayed to the user;
- CGs are persisted using the storage module.

Key points in the NI system are the mapping of commonsense, conversion of the hierarchy to Cogitant support format in addition to the process of answering questions. These points are explained in the following sections.

3.1 Commonsense Mapping

Instead of encoding the concept hierarchy by hand, NI system makes use of Cyc ontology. Commonsense module can map a given concept or relation to Cyc counterparts. This is achieved using "prettyString" predicates in the Cyc system. For every collection and relation Cyc contains assertions such as (prettyString TERM STRING), where STRING is the English word for TERM.

After mapping a given concept to corresponding Cyc collection, the full hierarchy up to the root element (#$Thing) is queried. This is achieved using "genls" predicates in the Cyc system, which means generalizations of a collection. As a result, we get a lattice, bottom most element is the concept at hand, and top most element is the root concept #$Thing. Same operation is also conducted for relations, using "genlPreds", which means generalizations of a relation. A generalization of a predicate holds whenever the specialization holds. Since Conceptual Graphs do not specify relation hierarchies, generalizations of a relation are used as forward rules. For example, #$temporallyRelated is a generalization of #$temporallyRelated, then whenever performedBy(x,y) holds, temporallyRelated(x,y) will be applied as a forward rule.

Below is the concept type hierarchy for the concept "Place" retrieved from the Cyc system;

```
#$Place ->
     #$EnduringThing-Localized ->
          #$Location-Underspecified ->
               #$Thing ->
          #$SomethingExisting ->
               #$Individual ->
                    #$Thing ^^
                    #$Trajector-Underspecified ->
                         #$Location-Underspecified ^^
               #$TemporallyExistingThing ->
                    #$TemporalThing ->
                         #$Individual ^^
          #$SpatialThing-Localized ->
               #$TemporallyExistingThing ^^
               #$SpatialThing ->
```

```
                    #$Individual ^^
               #$Boundary-Underspecified ->
                    #$Region-Underspecified ->
                         #$Location-Underspecified ^^
                    #$Landmark-Underspecified ->
                         #$Individual ^^
                         #$Location-Underspecified ^^
          #$SpatialThing-NonSituational ->
               #$SpatialThing ^^
               #$Individual ^^
     #$Location-Underspecified ^^
```

3.2 Conversion to Cogitant Support

Cogitant is the library used for reasoning module implementation. It is a set of C++ classes which can handle Conceptual Graph operations. Support and objects of the Conceptual Graph are specified as CoGXML, a specific XML format defining concept types, relation types, individuals, and graphs. Reasoning module converts concept hierarchy to Cogitant CogXML support format, and concepts to object format. Then, these XML data is loaded into Cogitant library, in order to carry out Conceptual Graph operations such as rule application, and projection finding.

In order to augment existing Conceptual Graph with concept hierarchies gathered from Cyc, first concept types are enumerated up to the root concept type. Then, relation types are enumerated, and written to CoGXML. Finally, rules obtained from Cyc relation hierarchies are converted to CoGXML rule graphs. As a result, Cogitant library becomes aware of the Cyc hierarchies and rules.

3.3 Answering Queries

Up to now, converting natural language sentences to CGIF, converting CGIF to CGs, and generating support for CGs from Cyc system are explained. As a result of these operations, we have CGs of input sentences augmented with commonsense knowledge. In order to benefit from these operations, we should use them to answer questions asked by the user. Since NLP module can differentiate the question sentences from regular sentences, we can create query graphs from question sentences and try to answer these queries. This operation is conducted using reasoning module. The system checks for projections from the query graph to existing CGs. If a projection does exist, the query graph is unified with the projected graph, and added to the list of answers. Then, these results are returned to the user.

Figure 6 illustrates a sample question answering process using the Natural Intelligence system. First, user enters a sentence, "Mr. Hyde ate two apples yesterday in New York happily". Then the user enters a second sentence, "John ate two oranges today in New Jersey sadly". The system converts these sentences to CGIF, and generates support from commonsense ontology. Finally, user enters

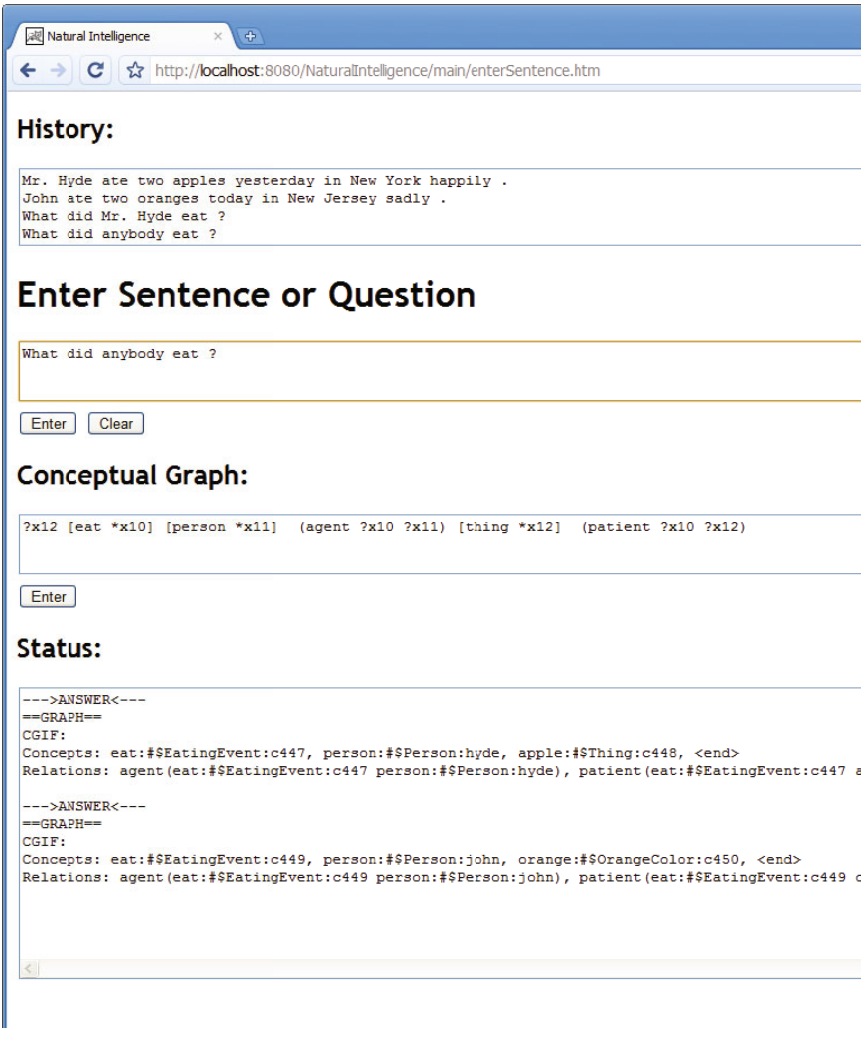


Fig. 6. Sample screenshot from Natural Intelligence System

a question sentence "What did anybody eat?" and the system lists answers found.

As a result of mapping concept types and relation types to Cyc counterparts we can make use of the implicit knowledge retrieved from the Cyc system. This means, NI system can also answer implicit questions like "Was Mr. Hyde there while eating the apples", "Does Mr. Hyde exist after eating the apples?", and "Do the apples exist after Mr. Hyde ate them?".

4 Conclusion

In this paper, Natural Intelligence, a question answering system employing OpenCyc as commonsense knowledgebase is described. Combinatory Categorial Grammar based tools are used to parse natural language and Conceptual Graphs are used to represent meaning, taking advantage of Cogitant library for CG operations. A modular framework for separating the responsibilities for NLP, reasoning, commonsense mapping and storage is defined. Finally, a simple iteration of entering sentences and answering questions is illustrated.

The significance of this work is the effort of putting together the various state of the art tools and libraries in order to build a central, integral, and scalable question answering system. This is a step towards to realize the goal of the Cyc system, to create intelligent applications. However, this is just a starting. We have a long way to go, in order to realize our long term goals. We have some restrictions due to the restrictions in the used tools. For instance, when there are problems with the NLP module, CGIF is not created in a well formed way, which causes the rest of the system to fail. Also commonsense mapping needs to be fine tuned in order to disambiguate the word senses according to the context.

As a future work, we plan to add rule induction, backward chaining capabilities, together with the improvements to NLP and commonsense modules.

References

1. McCarthy, J.: Some Expert Systems Need Common Sense. Annals of the New York Academy of Sciences 426(1), 129–137 (1984) (Computer Cult)
2. Bos, J., Clark, S., Steedman, M., Curran, J.R., Hockenmaier, J.: Wide-Coverage Semantic Representations from a CCG Parser. In: Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004), Geneva, Switzerland, pp. 1240–1246 (2004)
3. Steedman, M.: The Syntactic Process. MIT Press, Cambridge (2000)
4. Curran, J.R., Clark, S., Bos, J.: Linguistically Motivated Large-Scale NLP with C&C and Boxer. In: Proceedings of the ACL 2007 Demonstrations Session (ACL 2007 demo), pp. 29–32 (2007)
5. Sowa, J.F.: Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley, Reading (1984)
6. Sowa, J.F.: Knowledge Representation: Logical, Philosophical, and Computational Foundations. Brooks/Cole, Pacific Grove (2000)
7. Hockenmaier, J.: Data and Models for Statistical Parsing with Combinatory Categorial Grammar. Ph.D. thesis, University of Edinburgh (2003)
8. Clark, S., Curran, J.R.: Wide-Coverage Efficient Statistical Parsing with CCG and Log-Linear Models. Computational Linguistics 33(4) (2007)
9. Peirce, C.S.: Manuscripts on existential graphs. In: Collected Papers of Charles Sanders Peirce, vol. 4, pp. 320–410. Harvard University Press, Cambridge (1906)
10. Genest, D., Salvat, E.: A Platform Allowing Typed Nested Graphs: How CoGITo Became CoGITaNT. In: Mugnier, M.-L., Chein, M. (eds.) ICCS 1998. LNCS (LNAI), vol. 1453, p. 154. Springer, Heidelberg (1998)
11. Chein, M., Mugnier, M.L.: Graph-based Knowledge Representation. Springer, Heidelberg (2009)

Learning to Map the Virtual Evolution of Knowledge

Mary Keeler

mkeeler@myuw.net

Abstract. T. Deacon uses C. Peirce's theoretical perspective to examine a fundamental human predicament (the increasingly indirect linkage between the physical world and our symbolic experience). This paper argues that we might now use that same perspective to examine the evolution of knowledge in human leaning as a virtual extension of natural evolution. Peirce's pragmatic logic can then be appreciated as a method to achieve self-controlled evolution of our habits of thought. In building this argument, I address misconceptions about Peirce's logical theory (especially, that logic relies on language), and suggest his graphical forms (Existential Graphs) as an instrument for building a virtual "living ecosystem of knowledge," for conceptual structures research in the Semantic Web.

1 Introduction

As we enter the Semantic Web era and begin to cope with climate change, we must finally comprehend the "power-predicament" of *symbolic reference* (which T. Deacon brought to our attention at ICCS 2004).

> We live in a world that is both entirely physical and virtual at the same time. Remarkably, this virtual facet of the world came into existence relatively recently, as evolutionary time is measured, and it has provided human selves with an unprecedented sort of autonomy or freedom to wander from the constraints of concrete reference, and a unique power for self-determination that derives from this increasingly indirect linkage between symbolic mental representation and its grounds of reference. With it has come a more indirect linkage between mind and body, as well. [Deacon (1977): 454]].

This "increasingly indirect linkage" characterizes our most human condition, and Deacon finds in Peirce's theory useful perspective on the creative/destructive nature of symbolic reference. Can that perspective also help us appreciate the potential power of the Semantic Web to reckon better with our linkage predicament? From the Peircean vantage point, *symbolic evolution* is a virtual extension of natural evolution, through learning. In previous work [1995-2008], I have investigated Peirce's theory of how learning (or "inquiry," in his terms) can virtually evolve from instinct, and improve with habits of reasoning behavior for self-critical control. Peirce wrote, "virtual" can be ascribed to something if we can attribute to it the efficiency of something that it is actually not; "virtual" must not be confounded with "potential" [*CP* 6.372 (c. 1902)]. Virtual evolution would then never be, even potentially, actual evolution but would appear to have at least its efficiency. Might Peirce's theory of reasoning

M. Croitoru, S. Ferré, and D. Lukose (Eds.): ICCS 2010, LNAI 6208, pp. 108–124, 2010.

(and science of logic [*CP* 2.209]) assist us with our linkage power-predicament, if we re-conceive it as a theory of the virtual evolution of knowledge from natural instinct?

Deacon explains that the word instinct is a "somewhat archaic biological term" currently applied by those who refer to "some behavioral predisposition that is preformed and ready to operate prior to any experience." Moderating the *human language as instinct* theory of Chomsky, Pinker, and others, Deacon describes languages as "fuzzy collections of behaviors that happen to be reasonably well described by logic" [114]; they are abstractions that "don't just change, they evolve" [327, 109]. He contends that the evolutionary miracle of the human brain is extraordinary not just because it operates as a "flesh and blood computer," but because "the major structural and functional innovations that make human brains capable of unprecedented mental feats evolved in response to the use of something as abstract and virtual as the power of words" [321-22]. He suggests that *ideas change the brain*. James Mark Baldwin first outlined this suggestion in his subtle modification of Darwin's theory of natural selection, to explain that an animal can produce irreversible changes in the adaptive capability of future generations by adopting habits of behavior during its lifespan. "Though no new genetic change is immediately produced in the process, the change in conditions will alter which among the existing or subsequently modified genetic predispositions will be favored in the future" [in Deacon 322-23]. This *virtual evolution* of new behavior can determine actual evolution in a *co-evolutionary* process that seems to commit us to the dualistic mind/body behavior made famous by Descartes.

According to Matt Ridley, in *Genome: The Autobiography of a Species in 23 Chapters*, Baldwin pursued the question of why learning takes place at all, and why we assume it is advanced, while we consider pre-programmed instinct to be primitive. He points to the "factual mistake" that Artificial Intelligence researchers made by adopting this assumption in their goal of creating a general-purpose learning machine. Humans crawl, … stand, walk, cry and blink just as instinctively as a chick does, he explains; and we learn what we can graft onto these animal instincts. Ridley argues that the advantage of making things innate clearly has limits. If by natural selection the vocabulary of language became naturally instinctive, language would be too inflexible a tool; even birds must be able to recalibrate their star compass in every generation through learning. "The Baldwin effect is about the delicate balance between cultural and genetic evolution. They are not opposites, but comrades, trading influence with each other to get best results." "The main function of consciousness," says Baldwin, "is to enable [the child] to learn things which natural heredity fails to transmit" [in Ridley 221]. Ridley concludes: "In effect, since the process of natural selection is one of extracting useful information from the environment and encoding it in the genes, there is a sense in which you can look on the human genome as four billion years' worth of accumulated learning" [Ridley 221-22]. Peirce's theory attempts to explicate that continuity in the evolution of mind and matter, and his pragmatism is a guide to preserve that delicate balance in what Deacon considers *co-evolution.*

2 The Co-evolution of Instinct and Intelligence

Deacon assures us that, "whatever learning predispositions are responsible for the unprecedented human facility with language, they specifically cannot depend on innate symbolic information. No innate rules, no innate general principles, no innate

symbolic categories can be built in by evolution [in the genetic sense]" [339]. The unique representational functions and open-ended flexibility of symbolic representation give cultural evolution a virtual evolutionary force or dynamic, as he describes.

> The multitiered structure of living languages and our remarkably facile use of speech are both features that can only be explained as consequences of this secondary selection, produced by social functions that recruited symbolic processes after they were first introduced. These are secondary insofar as they became selection pressures only after symbolic communication was well established in other realms. They are, however, the primary causes for the extensive specialization that characterizes spoken languages and for the great gulf that now separates our abilities in these realms from those of other species. ... Our "instinct" for language is not some unitary and modular function (LAD), but co-evolutionary processes instead have produced an extensive array of perceptual, motor, learning, and even emotional predispositions [or biases], each of which in some slight way decreases the probability of failure at the language game. ... Though no one may be either indispensable or sufficient, together they guarantee the replication of language. [349-50]

From this co-evolutionary view, Deacon concludes, we can "recognize that the evolution of language took place neither inside nor outside brains, but at the interface where cultural evolutionary processes affect biological evolutionary processes" [409]. The difference between instinct and learning is only a matter of degree. "Both are, in one sense, internalizations of external correlations of events. One is built in before birth, one after birth" [67]. But the social evolution of symbolic communication created "a mode of *extrabiological inheritance* with a particularly powerful and complex character, and with a sort of autonomous life of its own." He insists that human anatomy, neurobiology, and psychology cannot be understood without recognizing that they have all been "shaped by something that could best be described as an idea: the idea of symbolic reference" [409-10 (emphasis added)].

In hindsight, we assume that symbolic reference was an advancement, but Deacon hypothesizes that it was one of many desperate responses to the environmental degradation resulting from the success of human foraging that occurred thousands of years ago, and is marked by the appearance of cave paintings and sculptures, as the first durable media. These "durable icons" indicate the beginning of a new phase, *cultural evolution*:

> one that is much more independent of individual human brains and speech, and one that has led to a modern runaway process which may very well prove to be unsustainable into the distant future. Whether or not it will be viewed in future hindsight as progress or just another short-term, irreversible, self-undermining trend in hominid evolution cannot yet be predicted. That we consider this self-undermining process *advancement*, and refer to the stable, successful, and until just recently, sustainable foraging adaptation of *Homo erectus* as "stagnation," may be the final irony to be played out by future evolution. [374]

Can we now learn to cope with our behavioral dilemma, which (Deacon points out) was increasing well before Descartes analyzed it as a dichotomy of mind and body? Peirce's theory of reasoning, with its pragmatic implications, recasts our predicament (here, expressed in Deacon's terms): "Symbolic reference is at once a function of the whole web of referential relationships and of the whole network of users extended in space and time" [452]. The common metaphor of mind as simply an algorithmic

process, Deacon says, "confuses the map with the territory, the indexical markers of points in a process with the process itself. ... The symbolic is reduced to the indexical" [446]. In the following sections, I suggest how Peirce's theory offers to clarify the usefulness of this machine metaphor, within what we can conceive as *the continuing virtual evolution of knowledge*.

3 Map versus Territory

In Jonathan Swift's story *Gulliver's Travels*, the adventurer visits a fictional land where he finds a man who carries on his back everything to which he needs to refer. We who trust language and other forms of symbolic representation to "carry the burden of reference" for us need not carry that physical weight, but we have another (and increasing) burden of bearing in mind that any symbolic expression we use cannot completely re-present what we may want to refer to. Deacon identifies the difficulty: "In order for a set of objects to serve as symbol tokens, they must be susceptible to being recoded in a consistent way. In other words, they need to be associated with one another in a pattern that maps to a closed logical system. Such lucky coincidences of object relationships very rarely occur by chance or by physical constraint. Indexical information is sufficient to adapt to the majority of complex social relationships, since most are dependent on making highly reliable assessments of others' behavioral propensities, anticipated by subtle behavioral symptoms [such as those our pet dogs can so accurately detect]" [337]**.** As our symbol-based learning progresses, we tend to lose the assurance of definite (indexical) reference to perceivable objects of experience, but we gain in power to express our feelings, thoughts, and judgments about that experience by symbolic abstraction.

The essential purpose—and virtue—of symbol use is not simply to transmit messages accurately, which is what information theory was conceived to predict, but to grow new ideas about our experience of objects, no matter how abstract or generalized those ideas (through symbolic reference) may become. Many of the recognizable objects in our experience have been created *as symbols*, entirely for the purpose of advancing the growth of abstract (or generalized) meaning. Without their symbolic purpose, these objects would represent no more than the mechanical, haphazard increase in information we now often experience as "overload." But here is our dilemma: on the one hand, if symbols were purely and accurately referential, they would be no use to us in learning more; but on the other, to the extent that symbols abstract from what they refer to in our existential experience, they may prevent us from learning any more than the rhetorical games they let us play with language. Such referent-less expressions probably now dominate our daily cultural experience, from weather forecasts and advertising to political pronouncements. The symbolic expressions we construct, within our culturally-evolved systems and forms of media, more often now have virtual *rather than actual* reference for us in communication with ourselves and with others. Although these "maps" no longer re-present the collections of objects we would otherwise have to carry around, they can still have some of the effectiveness of doing so; but only to the extent that we learn how to interpret them *pragmatically*, which depends on our remaining aware that the "territory" (of experience) is always more than what any "map" (or representation) can exhibit to us.

In our multi-cultural world, we must be aware that the many diverse language families differ radically in how they shape their speakers' thinking. Even among the Indo-European group, only English has numerous distinctive common nouns. In languages that have a verb meaning "is a man," the noun "man" becomes a superfluity. And since a noun or even combination of nouns by itself indicates nothing about objects — even the directly perceivable, Gulliverian sort — they give us no basis for interpretation or relating things, except as syntactical place-holders in symbolic formulations. When simply linked hypertextually on the Web, they are merely mechanically connected, not meaningfully related in the logical sense. Peirce explains that linguists find the roots of inference in verbs, even unspoken ones, and in the unassuming prepositions, which transparently make common nouns operate as unexpressed assertions in any language [*CP* 2.290 (1893); 2.341 (c. 1902)]. An English speaker who sees "Glass" written on a package will infer that there is glass inside, for example. We often create meaning from information by inferring possible semantic relations based on available *symbolic evidence*.

3.1 Building Semantic Relations in Representation

In defining terms (especially common nouns), we strive to make their relatedness explicit within the context of a language as a system of symbols, while establishing their referential reliability so that meaning can be relatively grounded, or normalized in usage. Terms have no meaning without these "semantic webs" we create that conventionally relate words to one another, which can be used to validate or standardize their application in specific linguistic contexts of use. These semantic relations (not the terms by themselves) — in sentences, propositions, assertions, and even in the arguments they imply — are the concepts or the *complex symbols* by which we represent what we assume or infer to be the meaningful relationships among the objects of our experience. In further inference, we use these abstracted relations to establish judgments about the world. The power of inference underlies symbolic (or conceptual) abstraction by which meaning and knowledge inevitably grow from information, often without our notice. Logic in its modern form can be an instrument for analyzing the intricacies of how inference can evolve to establish what we call knowledge from information. Peirce's theory of that instrument considers names, definitions, concepts and other features of semantic relations to be an elaborate set of hypotheses that are continually tested and improved through humans learning by experience.

Peirce credits Aquinas for re-defining logic as the science of second intentions applied to first intentions [*CP* 4.38 (1893), *CP* 3.490 (1897), *CP* 2.548 (1902)]). This definition only begins to clarify the purpose of logical analysis. *First intentions* are concepts derived from comparing percepts (these are fundamental concepts of classes, relations, etc.); *second intentions* are concepts formed by observing and comparing first intentions. Classifying objects of experience is a conscious part of our behavior only to the extent that we take notice and conceive what we are doing *as classification*. We can distinguish figments of imagination from realities, and meaningless terms from meaningful ones, only by our ability to relate second intentions in such concepts as *identity*, *otherness*, and *co-existence.* Although *meaning* has no technical status in logic, logicians have generally defined it in terms of *breadth* and *depth*: a sign stands for its denoted breadth, and it signifies its connoted depth. Depth or

signification is considered intrinsic and breadth extrinsic. Peirce's logical theory introduces a third kind of meaning: *pragmatic*. When we define an idea as a state of mind which consciously means something, we consider that it means something in the sense of intending or purposing something. "Now a purposive state of mind is one that signifies something by virtue of intending to be interpreted in a deed. Therefore, although an idea certainly has its internal and its external meaning, yet its principal meaning is of a different kind from either of those" [*CP* 8.119 (1902)].

Pragmatic meaning occurs in reference to supposed relationships *among* objects, not simply in reference to the objects themselves, as Gulliver's tale helps to illustrate; and the *argument* is the only form of representation that has the explicit purpose of determining the acceptance of its conclusion as an expressed relation. Since to call the conclusion of an argument its meaning accords quite well with general usage, Peirce designates the word "meaning" to denote the intended idea of a symbol [*CP*, 5.175 (1903)]. He then clarifies what is the focus and terminology of logical analysis. If second intentions are the objects in our understanding represented in symbols such as language, and the first intentions to which they in turn apply are the objects of those representations as we perceive or conceive them, then we can self-consciously derive *third intentions* as *representations* of second intention symbols when viewed themselves *as objects*, or *forms of argument* [*CP* 4.549 (1906)]. In these steps of building abstraction, we turn the *predicates* by which we think into *subjects* of our thought [*CP* 1.559 (c. 1893)].

3.2 Pragmatic Meaning

For Peirce, logic finds its proper phenomena of study at that third stage, by *hypostatic abstraction*, in which we use *forms* of symbolic notation to represent and analyze the conventional symbolic patterns of languages (second-stage abstraction), which express thoughts about the conceived objects of perceptual experience (first-stage abstraction). We can then say that an *argument* distinctly represents its idea as the conclusion, a *proposition* distinctly indicates the object which it denotes as its subject but leaves the interpreted idea (or meaning) to be whatever someone might interpret, and a *term* distinctly indicates only the object it denotes (it names only a particular, Gulliverian, object). Take away the subject of a proposition and you have a term called its predicate; take away the conclusion of an argument and you have a proposition called its premise (usually there is more than one) [*CP* 2.95 (1902)]. If arguments are the only forms of expression which truly relate our episodes of experience meaningfully, as hypothesized in Peirce's theory of logic, then *any fully functioning symbols must be arguments*, even if not usually expressed in that form.

The *logical forms* of argument in symbolic expression are most often not explicit. When we communicate informally, what is not explicit is assumed to be well enough understood, and without our notice can be manipulated in rhetorical style. But formal communication progresses by explicit arguments (or meaning) that can be efficiently validated by a communicating group. With his profound understanding of Aristotle's syllogistic logic (for centuries considered to be the final form of logic), Peirce proposed that a new logic be developed as a genuine science of reasoning. His *logic of relatives* (derived from earlier work of Boole and DeMorgan) has three necessarily-related stages of argument or forms of inference (abduction, deduction, and induction)

to explicate how meaning can evolve in experience. He argued that even in proposing a hypothesis to account for some facts (in abductive stage), a scientist must furnish reasons (to be argued and judged good or bad) as to why it is worthy of testing. It is the work of the logician to analyze these reasons and to discover an ideal method of investigation for pursuing *the truth* — to be understood as *the hypothetical result of indefinite inquiry* that encourages us to persist in reasoning at all.

Jonathan Swift ridiculed the syllogistic form of logic in Guliver's fictional adventure, describing there a machine for making science. We often too easily let ourselves fall into a mode of thinking which limits our conceivable choices, restricted by unexamined assumptions: a quasi-mechanical process that leaves little room for invention, a mode of behavior which is only disturbed when it clearly fails to function in response to altered environmental conditions. *Innovative response* — a property of life — is a matter of selecting previously unrecognized properties from direct life-experience (as concepts) and relating them in new forms of thought.

Deacon stresses that *to evolve* means to be "spontaneously adaptive," creating new means of "fitting with and anticipating" the environment, through "moment-to-moment natural selection of patterned information" [456]. Symbolic reference gives us "a system for representing synthetic logical relationships" [Deacon, 41] among symbols as objects, which can refer to things hypothetically, in the *form of logical patterns*. Peirce came to call this form of logic "Normative Semeotic" (he preferred the spelling "semeotic"— rather than "semiotics") [*CP* 8.377, 2.111 (1902)]. He defends what some consider "Aristotle's blunder": "If the main object of the syllogistic forms were in actual application, to test reasonings as to whose validity or invalidity we found it difficult to decide, as some logicians seem naively to suppose, then their close connection with ordinary habits of thought might be a paramount consideration. But in reality, their main function is to give us an insight into the inward structure of reasoning in general; and for that purpose systematic perfection is indispensable" [*CP* 2.458 (1893)]. Metaphorically speaking, our predicament is that each of us must "map a unique view of the territory of objects," and yet, through our pragmatic "mapping capability," we all must learn what is actually "the same territory to be mapped." Peirce's relative logic was designed for this pragmatic purpose.

4 Many Maps for the Same Territory

Alzheimer's researchers tell us that as victims lose memory they also lose a sense of the future, and so the ability to compare past to present and to conjecture about possible consequences in the future. They lose the sense of need for principle, planning, and strategy—the ability to make cause and effect or conditional judgments. These conditional inferences, of the form "if I do X, Y would happen" are a rudimentary facility in human reasoning and self-controlled behavior based on our sense of past, present, and future. As does any cultural behavior in life, intellectual behavior in the "life of thought" resides in its forms and patterns — although we can adopt intellectual norms more self-consciously as *habits of thought* in the methods, procedures, and conventions that must be explicitly learned.

With logic, we can study these forms and patterns of second-intentional representation: classify them, manipulate them, and observe how they can grow much as

scientists study the first intentional phenomena of nature. Second intentional phenomena of language and symbols *are objects* in the study of logic, called representations or signs. The first intentions to which those signs refer are the perceived objects to which logic refers only through the second-intentional phenomena of signs, which theoretically constitute the *normative conditions of interpretation*, according to Peirce, in his "Critical Analysis of Logic."

> [Logic, unlike the other sciences,] is not obliged to resort to experience for the support of the laws it discovers and enunciates, for the reason that those laws are merely conditional, not categorical. The normative character of the science consists, precisely, in that condition attached to its laws. The only purpose for which it is obliged to resort to experience is to establish a few facts, without which there could be no motive for its inquiries; and these facts are so extremely universal and atmospheric that no little acumen is required to make sure that they are anything more than empty formulae or at most hypotheses. [*CP* 2.65 (1902)]

In its modern redevelopment, logic was to serve as instrument to scrutinize the minute structural relations of symbolic expression used in the context of any formal reasoning procedure (not limited to human). Peirce, who worked as a scientist for the U.S. Coast and Geodetic Survey, led this development in America, in collaboration with European logicians. With his extensive training in logic, mathematics, philosophy, computing, and cartography, he understood the need for logic to serve as a sort of "lens" for inspecting the conduct of reasoning in science. Toward the end of his life, he assessed his 50-year effort to render it as a comprehensive analytical tool.

> I took it and melted it down, reduced it to a fluid condition. I filtered it till it was clear. I cast it in the true mold; and when it had become solid, I spared no elbow-grease in polishing it. It is now a comparatively brilliant lens, showing much that was not discernible before. I believe that it will only remain to those who come after me to perfect the processes. I am as confident as I am of death that Logic will hereafter be infinitely superior to what it is as I leave it; but my labors will have done good work toward its improvement. [*CP* 2.198 (1902)]

4.1 Pragmatic Logic

In his teaching at John Hopkins University (1879-1884), Peirce began to explain his pragmatism as the art of devising methods of research, "the method of methods" [*CP* 7.59 (1882)]. His pragmatic theory establishes logic as the *science of reasoning*, to replace traditional logic as "the art of reasoning." Logic is not a human invention, for Peirce, but simply a self-critical refinement of natural human reasoning in practice**,** making it more efficient. To accomplish that refinement, logic must consider "what reasoning ought to be" [*CP* 2.7 (1902)], not "how we do think [which is psychology]; nor how we ought to think in conformity with usage, but how we ought to think in order to think what is true" [*CP* 2.52 (1902)]. (And *truth* is what he says we must hypothesize as the theoretical limit or end of learning, giving us the hope we need to continue investigation**.)** We are morally responsible for our reasonings just as we are responsible for our behavior. His theory of logic *as normative science* is necessary to explain the directedness or *tendency of experience to grow as knowledge*, a virtual sort of experience, where "Nothing can be either logically true or morally good

without a purpose to be so. For the conclusion of an argument which is only accidentally true is not logical" [*CP* 1.575 (1902)]. Peirce's logic is the instrument for attaining clarity of thought, relying on ethics and aesthetics to determine what should be our ultimate aim; all together, these are the normative sciences we have yet to develop [*CP* 1.191 (1903), 1.573 (1906)**].** In Peirce's view, reasoning is a species of behavior that is subject to criticism: "A mental operation which is similar to reasoning in every other respect except that it is performed unconsciously cannot be called 'reasoning'," because "it is idle to criticize as good or bad that which cannot be controlled" [*CP* 2.182 (1902), 5.108 (1903)].

Because much of Peirce's advanced theory of logic (in his later work) is effectively inaccessible in some 80,000 pages of manuscript in the Houghton Library at Harvard, most modern scholars and researchers recognize his advancements only in piecemeal (often distorted) respects—if at all. For example, in his book *Things That Make Us Smart: Defending Human Attributes in the Age of the Machine*, Donald Norman recounts traditional logic's view of the human mind as a computational device (attributed to Descartes), which assumes that advances in the science and technology of computation, control, and communication are advances in the science of thought processes [228]. From this historical background, he concludes:

> Logic is most definitely not a good model of human cognition. Humans take into account both the content and the context of the problem, whereas the strength of logic and formal symbolic representation is that the content and context are irrelevant. Taking the content into account means interpreting the problem in concrete terms, mapping it back onto the known world or real actions and interactions. The point is not simply that people make internal mental models, stories, or scenarios of the problems they are attempting to solve ... People map problems back onto their own personal knowledge and experiences. [228]

Unlike the language of logic, he insists, "Human language takes into account the point of the encounter, which is to communicate" [229].

4.2 A Social Theory of Logic, the Relativity of Knowledge

Later in his life, Peirce advanced his theory of *logic as semeotic*, to explain that capability we so easily take for granted in its routine and pervasive operation: learning by experience through communication [*CP* 2.227; from a manuscript fragment, c. 1897]. With effective self-awareness of that human capability, we have been able to develop methods that improve the natural trial-and-error procedure of learning by experience. Peirce formulated his pragmatic method of logic for refining learning procedures, and he even created a graphical notation tool (the Existential Graphs), as a "topology of logic" for observing and demonstrating how the improvement of that capability can occur through the process of dialogic reasoning. He concluded that the essence of successful learning of any sort is due not primarily to the sophistication of its measuring instruments or its investigational techniques, although those are essential. Careful observation and ingenious conceptualization generate knowledge only to the extent that they are collaboratively validated by those engaged in learning.

His pragmatism identifies self-critical, collaborative learning through dialogue as the scientific method — our most highly evolved form of learning — and *science is not a body of certified truths or systematized knowledge*. Peirce even suggested that

knowledge is not the point of science at all, since knowledge though systematized may be dead memory (the hide-bound habits of thought) [*CP* 6.428 (1893)]. Scientists are members of a community who impartially pursue the truth (or "real meaning"), which none can know as absolute matter of fact and which must be conceived as an ideal or limit. The pursuit advances and is successful to the extent that members can produce testable representations. These rely on conjectures, or symbolic references to what each observer interprets as experienced evidence. *Knowing* is the tendency for the meaning of our representations to evolve reliably from increasing experience. "Does not electricity mean more now than it did in the days of Franklin? ... men and words reciprocally educate each other; each increase of a man's information involves and is involved by, a corresponding increase of a word's information" [*CP* 5.313 (1903)]. Although the ideal of scientific terminology is that each term should have a single exact meaning, Peirce explains,

> this requisite might be understood in a sense which would make it utterly impossible. For every symbol is a living thing, in a very strict sense that is no mere figure of speech. The body of the symbol changes slowly, but its meaning inevitably grows, incorporates new elements and throws off old ones. But the effort of all should be to keep the essence of every scientific term unchanged and exact; although absolute exactitude is not so much as conceivable. Every symbol is, in its origin, either an image of the idea signified, or a reminiscence of some individual occurrence, person or thing, connected with its meaning, or is a metaphor. [*CP* 2.222 (1903)]

Peirce's semeotic logical view of symbolic communication confirms the relativity of meaning, and our ultimate uncertainty as to what we actually know for sure. These are the conditions of representation that confront us: none of us will ever have “the map that can fully capture the territory" even of our experience (which continues to grow as we are constructing our “maps”); and after all, each of us can have only mortal (time-and-space-limited) experience of whatever exists as “the territory.” But pragmatism gives us this methodological hope: the more we can collaboratively “construct the maps based on increasing individual experiences," extended indefinitely in time and space by communication, the closer we can hope to approach knowing what really is the territory (that is, what might really be true). Of course, we must suppose that this semeotic process will continue indefinitely because, since we are part of “the creative evolution of the territory,” it remains beyond our reach, as our interpretations continue to contribute to its creation. Semeotic logic tells us that we can never establish *complete truth*, our representations can only indicate what progress we might make in testing evidence. Its pragmatic method says: Truth is the ideal limit for our continuing inquiry, what would be the result of indefinite learning [*CP* 5.311 (1903)].

To the extent that we uncritically believe that symbolic forms (of any sort) represent the truth, we fool ourselves that we have the only possible "map" of what truly is "the territory." We forget that our *necessarily hypothetical* view of what happens can never tell us for certain what *has, does, or will happen*, because *the truth is independent of what any person or even group of us thinks* [*CP* 2.55 (1902)]**.** In hypothesizing, we can *suppose that some surprising fact could be explained as a case of a certain general rule, and then assume that supposition on probation*. This conscious hypothesizing merely extends our natural reasoning behavior: each of us must believe something in order to make judgments, in order to direct our physical behavior — to

make our actions more than simple reactions (that is, to mediate our actions by means of inferences about what appears to be true). The urge to reach a conclusion, to take some "map" to be the truth, is a necessary part of effective behavior; but in doing so we must avoid exchanging the Gulliverian burden of reference for the burden of habit-bound thought in beliefs that cannot evolve through continuing experience. According to Peirce's social theory of logic, we must each consciously maintain a provisional view by self-critically examining the possible outcomes or implications of our beliefs as hypothetical judgments — as though we are playing a game.

An American Academy of Sciences report [A.P. story/3/6/01] concludes that those who have no intellectually challenging hobby, such as chess-playing or puzzle-solving, throughout life, are more than twice as likely to succumb to Alzheimer's disease. Strategic game-playing exercises our capability to formulate hypotheses, which does not commit us actually to do anything that has consequences beyond the conjectures as to *what would be the consequences* of doing what we conceive. The more experience we have, the closer to true our guesses will be. Peirce insists that psychological, sociological, or historical investigations alone cannot explain our refined guessing capability. Investigation of this virtual essence of human thought requires full logical analysis of the reasoning process in making conjectures, selecting and testing them. A theory of our learning capability should explain its evolution from instinct, Peirce insists, because: "All Human knowledge, up to the highest flight of science, is but the development of our inborn animal instincts" [*CP* 2.754 (1883), 6.604 (1893)]. And yet, we cannot have instincts for *every possible* circumstance: "When one's purpose lies in the line of novelty, invention, generalization, theory — in a word, improvement of the situation ... instinct and the rule of thumb manifestly cease to be applicable" [*CP* 2.178 (1902)]. Can logic help us learn to improve our linking of virtual and physical worlds?

5 Logic for Mapping

"We simply didn't evolve senses capable of detecting some of the most serious problems unaided. Knowledge of that suggests directions in which solutions might be found," observes Paul Ehrlich. In his *Human Natures: Genes, Culture, and the Human Prospect*, he concludes: "An answer to environmental misconceptions, if humanity could manage it, would be to create a conscious evolutionary process" [xi, 328]. Unfortunately, Deacon's evidence discourages hope for Erhlich's answer: particular biases in human sensory behavior and brain structure co-evolved with language development, so that human learning could possibly be considered a "pre-maladaptation." Just as certain actions or movements are impossible for us, unaided, "certain mental predispositions that serve well in some domains can get in the way of accomplishing otherwise trivial tasks that require a new perspective or logic. ... Success or failure at learning or problem solving depends on habits of attention, what we find salient and what we tend to notice, and how easily one can override and reprogram these tendencies" [48].

What sort of "detecting aid" would we need for a "conscious evolutionary process" that could augment our self-conscious capability to observe and analyze the possible consequences of the beliefs that drive our behavior? If none of us can have a

God's-eye view, and since we must effectively collaborate if we hope to create any reliable "map of the territory" for any realm of investigation, we need the "third-intentions lens" that pragmatic logic could provide. Reviewing that logical perspective: if *first intentions* are concepts that compare or relate percepts (that is, symbolic reference that categorizes and names Gulliverian objects), and *second intentions* are concepts that relate first intention concepts, then *third intentions* are concepts that relate second intention concepts (by hypostatic abstraction, or by remaining aware that these concepts relate concepts that relate percepts). When symbolic representations are considered under this logical "microscope," that is *as representations*, they can be viewed as *conceptual structures*, the forms or patterns that represent the structure of any natural language. We can then "observe" these forms as phenomenal objects (called signs), analyze their "genetic relatedness," and study how they "replicate and evolve."

5.1 A Genetic Perspective of Representation

From this *semeotic perspective*, we first notice that signs exist only in replica**.** They differ from first intentional objects in that essential respect; no object of first intentions (or object of perception) is an exact replica; in fact, we define nature by our observation of its infinite variations. But symbols function reliably for us in communication only to the extent that they are exact replicas. Take the word "man" printed on any page, it is the same word in all its occurrences. A common noun is a symbol we use to associate a conceived collection of objects, and in using its replicas we tend to develop the habit of thought (as belief or conception) that the objects in that collection are *in fact* related somehow, so that each symbol replica *can be* interpreted as referring to an object that is an instance of that *conceived* collection. When we use a noun, we normally take that association for granted, along with whatever basis there may be for relating the objects in the collection in the first place. In learning a language, we come to believe that the objects we can name in it are related in some way, using the noun to stand for that believed (or assumed) relationship. The question is: could we learn to map the structure of beliefs based on logically-defined conceptual relations, by which we could locate detailed assumptions in reasoning, those which are not noticed without that map-perspective? Could that map give us the *power* to experiment explicitly with alternative courses of thought? Could we even observe hypothetical consequences to which our conceptually-structured beliefs might lead? Such a map would diagram the *form of the relations* of the symbols we use in thought, regardless of their significance or signification, which is what deductive logic was designed to do centuries ago — and what technology does today.

As Donald Norman assumes, most logicians study only necessary reasoning (or deduction), and so confine their theories about reasoning to its "correctness," or our absolute inability to doubt the truth of the conclusion if the premises can be assumed to be true, which they explicate mathematically in two values, true and false. In his broader theory of logic, Peirce identifies deduction as the *stage of reasoning* in which we "fix our ideas as to what we shall understand by the meaning of a term" [*CP* 5.176 (1903)]. Under his theory of logic, learning can be understood as the *creating, validating, and testing of representations* — identified with the three stages of reasoning: retroduction (later renamed "abduction"), deduction, and induction [*CP*

1.65; from a manuscript of notes (c. 1896)]. Jay Zeman explains how these steps relate in genuine reasoning.

> Retroduction is educated hypothesis-formation which proposes initial organizations of figure in the problematic field. Deduction enters in a mediating way, drawing out the consequences of the abductive hypotheses. And induction consists in the return to experience which aims at confirming or refuting those hypotheses by seeing whether the deduced consequences hold or not [Zeman 1986, 12; *CP*: 2.269 (1903)]

Under close logical examination, Peirce finds that deduction is the critical link between the other two stages [*CP* 5.193 (1903)]. Retroduction essentially postulates a vaguely formulated deductive argument that might explain the facts and is capable of experimental verification (by induction). Induction and retroduction refer to the context and aim of reasoning, while deduction is its *engine*. In hypothetical inference we compose imaginary experiments and suppose their results: "If X happens, Y would result." Deduction's "lens" then exposes any assumed (not explicitly expressed) parts of the inference to critical examination and explication in formally expressed, symbolic detail. Peirce explains that the critical operations of deduction are performed by observing an argument as a diagram of formal relations, as mathematicians use formulas to find conclusions; but in logic the primary objective is to *understand the nature of the process by which the conclusion is reached* [*CP* 5.581 (1898)]. "The mathematician seeks the speediest and most abridged of secure methods; the logician wishes to make each smallest step of the process stand out distinctly, so that its nature may be understood. He wants his diagram to be, above all, as analytical as possible" [*CP* 4.533 (1906)].

5.2 Deduction as "Virtual Selection"

Even when expressed in its traditional algebraic form, deduction involves constructing a diagram (appearing as a formula) representing what we suppose is the hypothetical state of things. Even though we may not be able to formulate a precise hypothetical inference, in observing it we suspect that something might be true. In proceeding to learn whether it is true or not, the most difficult part of the operation is to form a plan of investigation. Not only do we have to select what features of the diagram are pertinent to pay attention to, but we must return to it repeatedly to check and modify certain features, based on our *inevitably growing experience* of what it refers to. Without this *human process* of improving the details of the diagram, our conclusions may be correct (or have a valid form), but they will not be the particular conclusions relevant to our purposes (or be reliable). Rule-driven deductions may even drive us to lose track of our purposes, as can any mechanism without our guidance. Under our conscious control, logical procedure driven by deduction gives inquiry its vital power of self-correction, as Peirce describes:

> [O]ne can make exact experiments upon uniform diagrams; and when one does so, one must keep a bright lookout for unintended and unexpected changes thereby brought about in the relations of different significant parts of the diagram to one another. Such operations upon diagrams, whether external or imaginary, take the place of the experiments upon real things that one performs in chemical and physical research. Chemists have ere now, I need not say, described experimentation as the putting of questions to Nature. Just

> so, experiments upon diagrams are questions put to the Nature of the relations concerned. [*CP* 4.530 (1906)]

Deductive or necessary reasoning aids in explicating the meanings of the terms in the premises of an argument, and should help us track the evolution in meaning of those terms. While the "necessary reasoning" of deduction is not infallible, any conclusions necessarily follow from the form of the relations set forth in the premises. Meanwhile, abduction furnishes possible explanations as hypotheses to test, but these are mere conjectures with no measure of certainty. Deduction is certain but only of its idealized forms or diagrams representing the explanations. It is induction that gives us the only approach to certainty concerning what we experience, but has nothing definite to test without the previous steps [*CP* 8.209 (c. 1905)]. In these stages of reasoning, Peirce's "logic of relatives" extends logic to account for the aim and context of learning by experience — all the way from hypothesis formation to experimentation. Non-relative logic gives the impression that deductive inference is simply following a rigid rule, no more than machines can do, Peirce explains. "People commonly talk of the conclusion from a pair of premises, as if there were but one inference to be drawn. But relative logic shows that from any proposition whatever, without a second, an endless series of necessary consequences can be deduced; and it very frequently happens that a number of distinct lines of inference may be taken, none leading into another" [*CP* 3.641 (c. 1902)]. Deduction has no way to select a possible inference "map" without abduction and induction to specify *what is our aim* in the search of *what territory* of experience. "Abduction seeks a theory. Induction seeks for facts. In abduction the consideration of the facts suggests the hypothesis. In induction the study of the hypothesis suggests the experiments which bring to light the very facts to which the hypothesis had pointed" [*CP* 7.218 (c. 1901)].

6 Conclusions: Logic in Virtual Evolution

According to Peirce's view, the process of learning is an iterative procedure in which the related forms of the symbol-replicas we use must function to maintain the evolution of meaning with reference to what we experience, giving us the sense of continuity in thought that makes what we call knowledge possible. Meaning, then, is a continuing inferential process of relating, rather than any permanent dyadic or arbitrary relation between sign and signified. "[N]o present actual thought (which is a mere feeling) has any meaning, any intellectual value; for this lies not in what is actually thought, but in what this thought may be connected with in representation by subsequent thoughts; so that the meaning of a thought is altogether something virtual. ... At no one instant in my state of mind is there cognition or representation, but in the relation of my states of mind at different instants there is" [*CP* 5.289 (1868)].

Peirce says he invented his Existential Graphs (EG) as formal logical notation to "put before us moving pictures of thought, I mean of thought in its essence free from physiological and other accidents" [*CP* 4.8 (1906)]. These graphs can *map* the relational evidence of arguments in as they are built [*CP* 4.512-513 (1903)], to make possible the same sort of critical control that sophisticated instruments and techniques give physical investigation in examining empirical evidence [MS 291 (1905)]). Deductive thought need not be the rigid rule-driven (algorithmic) procedure that

traditional logic conveniently assumes, if we realize its proper role in making explicit the evolution of meaning. Peirce's EGs serve as a logical instrument for observing deductive inference minutely enough in the critical testing of ideas that we can make meaning tend to become more and more reliable in reference, by an iterative practice that makes logical validity entail that reliability. "Thus," says Peirce, "the system of existential graphs is a rough and generalized diagram of the Mind, and it gives a better idea of what the mind is, from the point of view of logic, than could be conveyed by any abstract account of it" [*CP* 4.582 (1906)].

Peirce makes clear that his graphs were not intended as a calculus for "thinking machines" [*CP*: 4.581 (1906)]. Calculus seeks a solution, and by the most direct reasoning to be found; while logic must examine the possible paths reasoning can take, and any conclusion must be merely a new premise in a possible continuing argument. He further explains that mathematical treatment in measuring involves the concept of number but also the idea of continuous quantity. "Number, after all, only serves to pin us down to a precision in our thoughts which, however beneficial, can seldom lead to lofty conceptions, and frequently descends to pettiness" [*CP* 2.646 (1878)]. But the conception of continuous quantity, aside from its attempt at precision, gives us the power to generalize. Peirce agrees with Kant, "[Logic's] engine and distinction is accurate analysis; but he insists: "absolute completeness of logical analysis is no less unattainable [than] is omniscience. Carry it as far as you please, and something will always remain unanalyzed" [in *CP* Bibliography (1902)].

Logic will not tell us what data to select or what experiments to conduct, but it will tell us how to formulate a plan or procedure for *learning by experiment*. Deductive logic machines differ from other machines only in working by excessively simple principles operating in complex ways, instead of complex principles operating a monotonous ways. The result from a logic machine has a relation to the data fed in, that relation determines whether the result could be false so long as the data are true. Peirce reminds us that we often perform as a machine, turning out a written sentence expressing a conclusion, having been fed with a written statement of fact, as premise — a performance essentially no different from what a machine can do [*CP* 2.59]. To the extent that we behave that way, we are subject to the same sort of logical criticism as the procedure of a machine. Peirce stresses this point, saying, "no other in all logic, although it is a science of subtleties, is so hard to see. The confusion is embedded in language, leaving no words available to epigrammatize the error." Any instrument that performs inferences is subject to logical criticism to determine if from true premises they always yield true conclusions. Even if we decide that machines can think, we must be able to examine the logical correctness of their operations, "which we should still have to assure ourselves of in the same way we do now" [*CP* 2.56 (1902)]. Only our critical examination of it could give us that assurance and, consequently it would not strictly be a reasoning machine.

Logic can help us build maps of formal conceptual structures as abstract representations of our beliefs, ideas, and judgments, but it will not tell us how to use them reliably. Peirce designed his graphical instrument *for use in observing the deductive progress of thought*. He insisted that *when we think*, we are "conversing with another self that is just coming into life in the flow of time." But, he explains, "*When one reasons*, it is that critical self that one is trying to persuade ... The second thing to remember is that the man's circle of society (however widely or narrowly this phrase

may be understood), is a sort of loosely compacted person, in some respects of higher rank than the person of an individual organism" [*CP* 5.421 (1905), emphasis added]. If all thought is relative to our limited points of view, then communication with others is required in order for knowledge (or whatever we can tentatively agree is true) to refer increasingly reliably to the evolving world of our experience, making it possible for us to establish successful behavior in that world by virtually evolving our "maps" as part of its territory. Knowledge in a particular field can continue to progress effectively, depending on how well a communication procedure works for validating individual interpretive contributions — providing that we never forget that the "valid judgment" established by any group of inquirers is never final and infallible with respect to the evidence. Increasing validity entails improving reliability.

Erhlich emphasizes that to make much sense of human biology we must consider the context of culture, and that history is now how we refer to the evolutionary process of cultural change [see x.]. He cautions: "If *Homo sapiens* is to improve its lot by manipulating human evolution, clearly it must do so by attempting to influence the course of human cultural evolution — and doing that with great care to avoid the abuses that could so easily occur and to preserve the diversity of natures that is such an important human resource" [330]. In converting social movements into conscious evolution, he says we require "a systematic, interdisciplinary consideration of the issues involved" by a process that is "transparent to all participants" [329]. Can Conceptual Structures be developed to effect that sort of transparent view of the *logic* (rather than the languages) by which our reasoning operates, the virtual nature of which logic-based maps would reveal [see Majumdar, et al. (2008); see note, below]?

Notes and References

General Note: "MS" references are to Peirce's manuscripts archived at the Houghton Library, Harvard; for *CP* references, *Collected Papers of Charles Sanders Peirce,* 8 vols., ed. Arthur W. Burks, Charles Hartshorne, and Paul Weiss (Cambridge: Harvard University Press, 1931-58).

Note: Graphs cannot, it is true, readily be applied to cases of great complexity; but for that very reason they are less liable to serve the purposes of the logical trifler. In the opinion of some exact logicians, they lead more directly to the ultimate analysis of logical problems than any algebra yet devised. [*CP* 3.619 (1911)]

References

Deacon, T.: The Symbolic Species: The Co-evolution of Language and the Brain. W.W. Norton (1977)

Deacon, T.: Memes as Signs in the Dynamic Logic of Semiosis: Beyond Modular Science and Computation Theory. In: Wolff, K.E., Pfeiffer, H.D., Delugach, H.S. (eds.) ICCS 2004. LNCS (LNAI), vol. 3127, pp. 17–30. Springer, Heidelberg (2004)

Erhlich, P.: Human Natures: Genes Cultures, and the Human Prospect. Island Press (2000)

Keeler, M.: The Philosophical Context of Peirce's Existential Graphs. Cognito, Centro de Estudos do Pragmatismo Filosofia (2004)

Keeler, M.: Communication and Conceptual Structuring. In: Conceptual Structures: Knowledge Respresentation as Interlingua, Auxiliary Proceedings, pp. 150–164 (1996)

Keeler, M., Kloesel, C.: Communication, Semiotic Continuity, and the Margins of the Peircean Text. In: Greetham, D. (ed.) The Margins of the Text, pp. 269–322. University of Michigan Press (1997)

Keeler, M.: Pragmatically Yours. In: Ganter, B., Mineau, G. (eds.). LNCS (LNAI), vol. 1876, pp. 82–99. Springer, Heidelberg (2000)

Keeler, M.: Hegel in a Strange Costume: Reconsidering Normative Science in Conceptual Structures Research. In: Ganter, B., de Moor, A., Lex, W. (eds.) ICCS 2003. LNCS, vol. 2746, pp. 37–53. Springer, Heidelberg (2003)

Keeler, M., Pfeiffer, H.: Games of Inquiry for Collaborative Concept Structuring. In: Dau, F., Mugnier, M.-L., Stumme, G. (eds.) ICCS 2005. LNCS (LNAI), vol. 3596, pp. 396–410. Springer, Heidelberg (2005)

Keeler, M., Pfeiffer, H.: Building a Pragmatic Methodology for KR Tool Research and Development. In: Schärfe, H., Hitzler, P., Øhrstrøm, P. (eds.) ICCS 2006. LNCS (LNAI), vol. 4068, pp. 314–330. Springer, Heidelberg (2006)

Keeler, M.: Revelator Game of Inquiry: A Peircean Challenge for Conceptual Structures in Application and Evolution. In: Priss, U., Polovina, S., Hill, R. (eds.) ICCS 2007. LNCS (LNAI), vol. 4604, pp. 443–459. Springer, Heidelberg (2007)

Keeler, M., Majumdar, A.: Revelator's Complex Adaptive Reasoning Methodology for Resource Infrastructure Evolution. In: Eklund, P., Haemmerlé, O. (eds.) ICCS 2008. LNCS (LNAI), vol. 5113, pp. 88–103. Springer, Heidelberg (2008)

Majumdar, A., Sowa, J., Stewart, J.: Pursuing the Goal of Language Understanding. In: Eklund, P., Haemmerlé, O. (eds.) ICCS 2008. LNCS (LNAI), vol. 5113, pp. 21–42. Springer, Heidelberg (2008)

Norman, D.: Things that Make Us Smart. Perseus Publishing (1993)

Ridley, M.: Genome: The Autobiography of a Species in 23 Chapters. Harper Collins (1999)

Zeman, J.: Peirce's Philosophy of Logic. Transactions of the Charles S. Peirce Society 22, 1–22 (1986)

Branching Time as a Conceptual Structure

Peter Øhrstrøm, Henrik Schärfe, and Thomas Ploug

Department of Communication and Psychology, Aalborg University
Kroghstraede 3, 9220 Aalborg East, Denmark
{poe,scharfe,ploug}@hum.aau.dk

Abstract. This paper deals with the history and the philosophy of some important conceptual structures of time and modality. In particular, the focus is on the historical and philosophical background of the introduction of the notion of branching time as a useful conceptual structure in philosophical logic. It is pointed out that the idea was first suggested by Saul Kripke in a letter to A.N. Prior, dated Sept. 3, 1958. It is also shown in the paper that Prior received the idea positively and that he developed it significantly in his later writings, although he at least in the beginning met the idea with some reservation and hesitation. Prior's development of branching time may be understood as a crucial part of his attempt at the formulation of a conceptual framework integrating basic human notions of time, free choice and ethics. Finally, the paper presents some challenges regarding the significance of branching time in philosophy and in the study of information architecture.

Keywords: Branching time, tense-logic, time, A.N. Prior.

1 A.N. Prior's Tense Logic

How should reality be understood and represented? This was a key question in the logic and philosophy of A.N. Prior (1914-69). In particular, Prior was interested in the temporal and the modal aspects of reality. In fact, Prior became the founder of modern temporal logic. He presented his ideas and theories in a number of books and papers. However, we may obtain an even deeper understanding of his ideas and motivations from a study of the papers and letters he left, most of which are now kept in the Prior collection at Bodleian Library, Oxford. (For a detailed description of this collection, see the Prior-site, www.prior.aau.dk.) In this paper we will rely not only on the books and papers Prior published during his lifetime, but also on the material left in the papers in the Prior collection and elsewhere.

What does it mean for something to exist in the physical world given that the world is constantly changing? In trying to answer this question Prior strongly emphasized the importance of the tense-logical aspects of reality. According to Prior only the present is real. The past is what was real, and the future is what will become real. Prior used the term 'temporal realism'. This could in fact serve as an informative name of his ontological position. However, in order to explain what temporal realism means, Prior also had to deal with the very notion of time. In an essay entitled 'A

M. Croitoru, S. Ferré, and D. Lukose (Eds.): ICCS 2010, LNAI 6208, pp. 125–138, 2010.

Statement of Temporal Realism' he wrote: "Time is not an object, but whatever is real exists and acts in time. We can describe most of what happens in time by talking about events being earlier and later than one another, and it is possible to construct a formal calculus expressing the logical features of this earlier-later relation between events. But this earlier-later calculus is only a convenient but indirect way of expressing truths that are not really about 'events' but about *things*, and about what these things are doing, have done and will do." [1: 45]

The point here is that according to Prior the set of temporal instants and the structure of earlier and later do not constitute the primary part of the objective reality. In his view, temporal instants as well as the ordering of the temporal instants are practical constructions which we as humans have made up in order to facilitate the description of the world, but these features are just derived or secondary aspects of the objective reality. The distinction between past, present, and future, on the other hand, is in fact essential for the proper understanding of Prior's philosophy and logic. Prior's central ontological tenet was that the distinction between past, present, and future is essential for a correct understanding of the objective world. In his own words: "So far, then, as I have anything that you could call a philosophical creed, its first article is this: I believe in the reality of the distinction between past, present, and future. I believe that what we see as a progress of events *is* a progress of events, a *coming to pass* of one thing after another, and not just a timeless tapestry with everything stuck there for good and all." [2: 47]

According to Prior, Eddington is reported once to have said that "events don't happen, we merely come across them" [3: 47]. In Prior's opinion this does not make much sense, since it would be very hard not to understand "coming across an event" as a happening! In addition to his fundamental creed regarding the role of the tenses in a proper understanding of reality, Prior held that freedom of choice also is something very important and essential in the real world. He expressed this belief in real freedom in the following way: "One of the big differences between the past and the future is that once something has become past, it is, as it were, out of our reach - once a thing has happened, nothing we can do can make it not to have happened. But the future is to some extent, even though it is only to a very small extent, something we can make for ourselves. And this is a distinction which a tenseless logic is unable to express." [2: 48]

Prior maintained that the study of change in terms of past, present and future should be regarded as an important part of logic. He criticized the common approach to logic according to which only unchanging (eternal) truths are studied: "Certainly there are unchanging truths, but there are changing truths also, and it is a pity if logic ignores these, and leaves it to existentialists and contemporary informal 'dialecticians' to study the more 'dynamic' aspects of reality. There are clear, hard structures for formal logicians to discover in the world of change and temporal succession. There are practical gains to be had from this study too, for example in the representation of time-delay in computer circuits, but the greatest gain that a logic of tenses brings is the accurate philosophical description of the reality of the passage of time." [2: 46]

It is certainly interesting in this quotation to note that Prior, although he knew very little about computers, in fact hinted at the possible use of modal and temporal logic in computer science. It is very clear that he was aware of this relation. In fact, he published one of his papers [4] in *The Journal of Computing Systems*, and in *Past,*

Present and Future he stated that discrete models of tense logic are "applicable in limited fields of discourse in which we are concerned only with what happens next in a sequence of discrete states, e.g. in the workings of a digital computer" [5: 67]. These anticipations have certainly later proved to be correct (see for instance [6: 344 ff]). However, Prior's main idea regarding the importance of temporal logic was philosophical. He wanted first of all to emphasize that we have to focus on the role of tenses if we want to grasp the reality of the passage of time.

Prior's book *Time and Modality* from 1957 was the first longer presentation of his tense logical approach to the understanding and representation of reality. The book was based on his John Locke lectures in the University of Oxford in 1956. It constitutes a strong case for an understanding of reality in terms of tense logic. Prior demonstrated in the book and in the following books on the topic as well as in many papers in various journals that the development of tense logic is very important if we want to establish a deeper understanding of the temporal aspects of reality, i.e. if we want a clear representation of temporal realism. However, he also clearly showed that although the development of tense logic is a very fascinating task, it is certainly also a complicated project which gives rise to many difficult challenges.

2 Diodorean Modality and Kripke's Branching Time Model

In the early 1950s Prior had become interested in the ideas on time and modality studied by the Megarian logician, Diodorus, who was a younger contemporary of Aristotle. Diodorus had argued in favour of determinism, and he had suggested that the concept of possibility should be understood in terms of temporal notions. In fact, according to Diodorus something is possible if and only if it is true now or at some time in the future. In [7] and in [8] Prior discussed the properties of this Diodorean Modality. In [8] Prior suggested that a proposition should be represented as an infinite sequence of 'true' and 'false'. For some reason Prior used '1' for 'true' and '3' for 'false. For instance, a proposition, p, might correspond to the following sequence of 'true' and 'false':

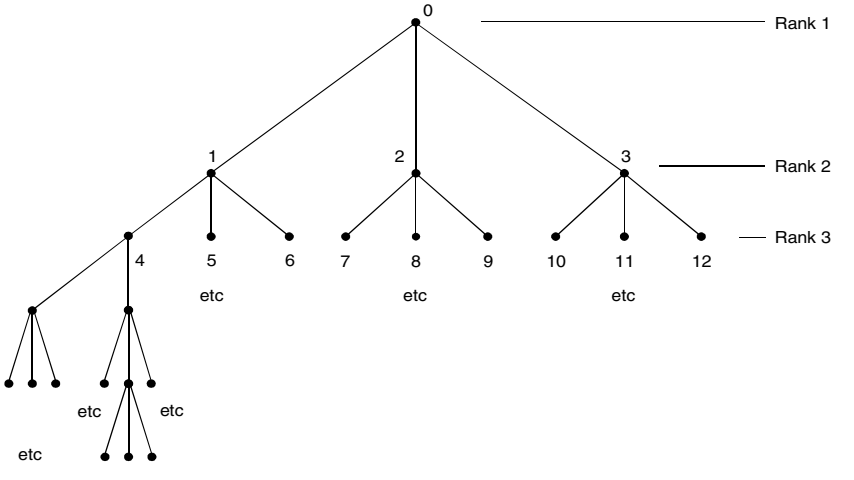
131333333133133 (and only 3's after this)

This sequence represents the present (i.e. the first '1') and the future development. The past is not taken into account in this model. Given the proposition, p, we may form the proposition, $\Diamond p$, corresponding to 'possibly p'. (Prior himself symbolized this as Mp, and 'neccesarily p' as Lp.) The sequence for $\Diamond$p can easily be constructed from the sequence corresponding to p. An element in the $\Diamond$p sequence is 1, if and only if there is a 1 at the same place or at a place to the right (i.e. later) in the p sequence. Using this Diodorean notion of possibility we can easily find the sequence corresponding to the proposition, $\Diamond$p, where p is the proposition (sequence) mentioned above:

p	1	3	1	3	3	3	3	3	3	1	3	3	1	3	3	only 3's after this
$\Diamond p$	1	1	1	1	1	1	1	1	1	1	1	1	1	3	3	only 3's after this

Given this representation we may of course wonder to which axiom system the Diodorean modality would correspond. In a rather tentative chapter of his book, Prior suggested that the system in question is simply the modal system, S4. One of the first readers to react on Prior's book, *Time and Modality*, was Saul Kripke who was only 17 years old when he wrote the following to Prior: "I have been reading your book *Time and Modality* with considerable interest. The interpretations and discussions of modality contained in your lectures are indeed very fruitful and interesting. There is, however an error in the book which ought to be pointed out, if you have not learned of it already." [L1: Sept. 3, 1958]

Young Saul Kripke then continued his letter by explaining that the formula, $\Box\Diamond p \vee \Box\Diamond\sim p$, can be verified using Prior's sequences and his definition of Diodorean modality, but that this formula can be shown not to be provable in S4. Kripke also suggested an axiomatic system corresponding to Diodorean modality. Thereby he opened a very fruitful line of research which engaged several researchers in the late 1950s and the early 1960s. (See Prior's overview in [5: 176]). Even more important was the following passage from Saul Kripke's letter in which he suggested how the semantics of S4 could be visualized. Kripke's formulation of this very original idea in the letter makes it reasonable to classify the occurrence of this letter as one of the most important events in the history of logic during the 20th century. Kripke wrote: "I have in fact obtained this infinite matrix on the basis of my own investigations on semantical completeness theorems for quantified extensions of S4 (with or without the Barcan axiom). However, I shall present it here from the point of view of your 'tensed' interpretation. (I myself was working with ordinary modal logic.) The matrix seems related to the 'indeterminism' discussed in your last chapters, although it probably cannot be identified with it. Now in an indetermined system, we perhaps should not regard time as a linear series, as you have done. Given the present moment, there are several possibilities for what the next moment may be like -- and for each possible next moment, there are several possibilities for the next moment after that. Thus the situation takes the form, not of a linear sequence, but of a 'tree':



The point 0 (or origin) is the present, and the points 1, 2, and 3 (of rank 2) are the possibilities for the next moment. If the point 1 actually does come to pass, 4, 5, and 6 are *its* possible successors, and so on. The whole tree then represents the entire set of possibilities for present and future; and every point determines a *subtree* consisting of its own present and future. Now if we let a tree sequence attach not three (as above) but a denumerable infinity of points to every point on the tree, we have a characteristic matrix for S4. An element of the matrix is a tree, with either 1 or 3 occupying each point; the designated tree contains only l's. If all points on the proper 'subtree' determined by a point on the tree p are 1's, the corresponding point on Lp is a 1; otherwise, it is a 3. (In other words, a proposition is considered "necessary" if and only if it is and definitely always will be the case.)" [L1: Sept. 3, 1958]

In this way Saul Kripke argued that S4 corresponds to a branching time system combined with the Diodorean notion of temporal modality. This presentation of branching time as a logical system is the first ever. This was clearly recognised by Prior, who in his book *Past, Present and Future* [5] discussed what he called "Kripke's branching time matrix for S4" [5: 27]. However, it should be noted that in such a model, it may in some cases be true that $F(n)p \wedge F(n)\sim p$, where $F(n)$ means 'it will be the case in n time units'. In consequence, $\sim(F(n)p \wedge F(n)\sim p)$ will not be a theorem in the system. This would certainly be intuitively acceptable, if $F(n)$ were understood as 'possibly, it will be the case in n time units'. But if we want a logic for $F(n)$ conceived as 'it will be the case in n time units' as opposed to 'possibly, it will be the case in n time units', we have to look for a further elaboration of the tense-logical system and its representation in the branching time model. Much of the recent work on branching time models has been focused on a satisfactory representation of the future operator in terms of branching time (see [9, 10]).

3 Tense Logic in Critical Perspective

Although Saul Kripke with his letter from September 3, 1958, became the first to suggest the logical idea of branching time and thereby contributed significantly to the development of tense logic, in his next letter to Prior dated October 13, 1958, he presented some doubts regarding the importance of tense logic. He wrote: "I am a little uncertain as to your own beliefs concerning the importance of tense logic. Is it your contention in Appendix A that a tenseless logic is really insufficient to represent the distinctions tense logic conveys? Do you think a tensed logic is needed for scientific discourse? I should think that, *for scientific discourse* a tenseless logic may be preferable." [L3: Oct. 13, 1958]

The Appendix A mentioned above is a separate part of Prior's *Time and Modality* [8: 102-22]. It contains a philosophical argument for the importance of tenses in logic. The argument is put into a historical context and in the appendix there is a strong emphasis on the philosophy of C.S. Peirce. In fact Prior sometimes even called himself a Peircean. To some extent, Prior conceived his work as a continuation of Peirce's philosophy and logic. In the appendix, Prior referred to the following statement made by Peirce in about 1903: "Time has usually been considered by logicians to be what is called 'extra-logical' matter. I have never shared this opinion. But I have thought that logic had not yet reached the state of development at which

the introduction of temporal modifications of its forms would not result in great confusion; and I am much of that way of thinking yet." [C.S. Peirce 4.523]

Prior wanted to take up this Peircean challenge and to carry out the task by integrating tenses in logic. He argued that we should in fact accept "tense distinctions as a proper subject of logical reflection" and that we should accept that "what is true at one time is in many cases false at another time, and vice versa" [8: 104]. Following these ideas, he showed in the book how this can be done in terms of symbolic logic. Although Prior's tense logic can be understood as a modern version of the way logic was conceived in the Middle Ages, and although this understanding of logic can be said to have been anticipated by Peirce and others, many logicians have seen Prior's logic as something quite new and different. Being brought up with the idea that logic should deal only with timeless (eternal) structures, they would expect that it should be possible to express all propositions that occur in logic using copula in the present tense, typically 'is' and 'are', and understanding these copula in a tenseless manner. Saul Kripke's questions should clearly be understood as based on this traditional understanding of logic. In his reply letter dated October 27, 1958, Prior argued that there are in fact important aspects of reality which cannot be described satisfactorily in terms of a tenseless (atemporal) logic. Prior referred to his basic belief in indeterminism, a notion which also had been mentioned by Saul Kripke in his second letter. In his letter Prior wrote: "In your paragraph about this you have a sentence beginning 'And if we accept indeterminism …', but I do not see how indeterminism can be expressed in a tenseless language at all. For indeterminism asserts a certain difference between the future and the past (that one has always APnpPnNp, but not always AFnpFnNp), which is not at all the same thing as a difference between the earlier and the later." [L4: Oct. 27, 1958]

This point is a very important point in Prior's philosophical logic. The idea is that there is a fundamental asymmetry between the past and the future, which Prior formulated in the Polish notation which he normally used in his writings. Translated into modern formalism Prior states that $P(n)p \vee P(n)\sim p$ is true in all possible cases (i.e. it is a theorem), whereas the same does not hold for $F(n)p \vee F(n)\sim p$. (Here $P(n)$ means 'it was the case n time units ago', and $F(n)$ means 'it will be the case in n time units'.) Prior's reason for denying that the latter disjunction is a theorem, is that if the proposition p depends on the free choice of some agent, then neither $F(n)p$ nor $F(n)\sim p$ should regarded as true now. In Prior's opinion, there is no truth about which future decision the agent will make (until the agent has actually made his or her decision). However, regarding the past exactly one of the propositions, $P(n)p$ and $P(n)\sim p$, is true, since only one of the propositions, p and $\sim p$, corresponds with how things were n time units ago. According to Prior, indeterminism is precisely the asymmetry between past and future expressed here. It should be mentioned that this logical representation of indeterminism depends on the choice of tense-logical system. The above representation is based on the logical system, which Prior himself preferred, i.e. the so-called Peircean system. However, Prior also considered the so-called Ockhamistic system, in which indeterminism would correspond to the rejection of the disjunction $\Box F(n)p \vee \Box F(n)\sim p$ as a theorem, whereas $F(n)p \vee F(n)\sim p$ will be accepted as an Ockhamistic theorem. More about the distinctions between the Peircean and the Ockhamistic systems can be found in [5: 113 ff, 6: 211 ff, 9].

4 Prior's Early Reactions on Kripke's Branching Time Idea

It appears that Prior found Kripke's model of branching time rather attractive since it could reflect the asymmetry between past and future, which he regarded as fundamental for a satisfactory understanding of the time. Nevertheless, he also received Saul Kripke's idea with some reservation, since the idea – at least in Kripke's presentation – was based on the notion of discrete time. Prior wrote: "An odd point I notice about the tense-logical interpretation of your 'trees' is that the passage of time, represented by the movement to new 'levels', is discrete. I don't know whether this feature is eliminable; I have sometimes myself wondered whether the notion of 'alternative futures' presupposes the discreteness of time. [L4: Oct. 27, 1958]

It seems, however, that this hesitation regarding what limitations the idea of branching time may put on other aspects of the representation of time did not last long. It appears that he rather quickly came to the conclusion that the idea of branching can be combined with ideas of continuous time as well as with ideas of discrete time. In fact, as he reported it in *Past, Present and Future* the tense-logical axioms corresponding to the denseness of time and to the linearity of time are independent. This means that when we are formulating the axiomatic system corresponding to our view of time and reality, the question of linear versus branching time can be answered independently of how the question about denseness is answered.

In *Past, Present and Future* the idea of branching time is fully accepted. In appendix B [5: 187 ff] of the book Prior when discussing the logic of world-states even used some kind of diagrammatical reasoning based on the notion of branching time. In his letter Prior also responded on another problem raised by Kripke regarding the relevance and importance of tense logic. Kripke had pointed out that the emphasis on the present (the Now) is rather problematic if we assume a scientific discourse taking relativistic physics into serious account. Prior was certainly aware of the challenge from special relativity. However, he also argued that it is in fact possible to maintain the tense-logical position without contradicting the result of relativistic physics (see [6: 197 ff]).

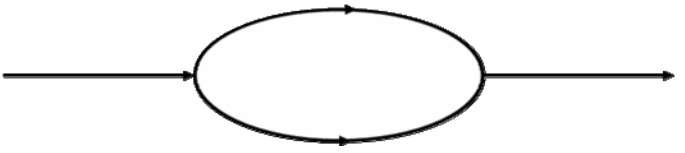
5 Prior's Consideration: Could There Be Backwards Branching?

In Saul Kripke's model there is only forward branching. It is however obvious to discuss whether backwards branching would be conceivable too. Prior discussed the problem referring to the idea that although there may be 'alternative possible immediate futures', there may only be 'one ultimate future'. In a letter to Henrik von Wright dated Feb.1, 1966, he wrote the following indirectly referring to Saul Kripke's findings: "MNp → NMp is known to be independent of S 4. And if your Np is intended to be read as "p now + for ever", S 4 is only sufficient if we allow that time may be branching (what I have said in Time and Modality about this is erroneous) if we want a logic of linear time, we shall also need MNp → NMp (which, if added to S4, gives S4.2); but even this doesn't preclude the possibility of time branching, though it does tell us that if time branches, the branches all eventually come together again ... To exclude even this, you'd need also

$$(Mp \,\&\, Mq) \rightarrow (M\,(p \,\&\, Mq) \text{ or } M\,(q \,\&\, Mp)),$$

or some equivalent formula (which, when added to S 4, would give S 4.3). And I had assumed that you did want your N to express the 'now + for ever' of ordinary linear time." [Letter 5]

Similarly, he in [5] pointed out that some theologians and some Marxists, apparently "write as if this is how things are". If that is in fact the case, time would correspond to a diagram like the following:



If this is the case there will be a future at which there is no unique past, but several 'possible pasts'. In addition, we'll have to accept that something, which was true about the past, is not true about the past anymore. However, Prior pointed out that some philosophers have been ready to accept such consequences. One of them was Łukasiewicz, who wrote: "If, of the future, only that part is real today which is causally determined by the present time; . . . then also, of the past, only that part is real today which is still active today in its effects. Facts whose effects are wholly exhausted, so that even an omniscient mind could not infer them from facts happening today, belong to the realm of possibility. We cannot say of them that they *were* but only that they were *possible.* And this is as well. In the life of each of us there occur grievous times of suffering and even more grievous times of guilt. We should be glad to wipe out these times not only from our memories but from reality. Now we are at liberty to believe that when all the consequences of those fatal times are exhausted, even if this happened only *after* our death, then they too will be erased from the world of reality and pass over to the domain of possibility." [5: 28]

According to Prior, however, this view is problematic. He pointed out that if we were to accept it, we would have to assume that *the* past "is just wiped out at the end of the day", i.e. that correct memories suddenly turn into incorrect or mistaken memories. This way of putting it, clearly illustrates why the asymmetry between past and future certainly appears reasonable. Łukasiewicz could of course answer, that he is only referring to aspects of the past which cannot be remembered by anybody anymore because all their effects "are wholly exhausted". However, a supporter of the idea of alternative pasts has to assume not only that there are elements of the past which are not remembered by anybody anymore, but also that what was true about the past may become false. Łukasiewicz may be right in claiming that such a model operating with the idea of alternative pasts can be formulated without contradiction, and that we therefore would be at liberty to believe in it. But the most interesting question is, of course, whether the temporal aspects of the real world actually include alternative pasts not only as a possible epistemological condition but also as an ontologically correct description.

6 Prior's Later Elaborations of the Notion of Branching Time

In his later writings Prior significantly contributed to the further development of the notion of branching time. One basic question regarding the branching time diagram has to do with the status of the points in the model. What is an instant? In [5 ff & 187 ff] he suggested a logic of world-states. This idea was further developed in *Papers on Time and Tense* [11]. His claim was that an instant in a branching time structure is in fact a world-proposition i.e. intuitively, an infinite conjunction of all propositions in a maximal and consistent subset of the set of all well-formed formulae in the logical language we are dealing with. In this way instant propositions are descriptions of possible states of the world which are as complete as they can be given in the language we have chosen. Following Prior's idea in [11: 128 ff] the set of instant propositions can be characterized by the following three axioms, where *a* stands for an arbitrary instant proposition, and where *p* stands for an arbitrary proposition (whether instant proposition or not):

(I1) $\Diamond a$
(I2) $\exists a: a$
(I3) $\Box(a \supset p) \vee \Box(a \supset \sim p)$

According to (I1) any instant proposition represents a possible world-state, and (I2) is the claim that there is an instant proposition describing the present state of the world. The last condition, (I3) is the claim that any instant proposition will be total in the sense that for any other proposition, *p*, it either necessarily implies *p* or it necessarily implies the negation, *~p*. It should be noted that any proposition in the logical language can be substituted for *p*. In consequence, *a* also implies which instant propositions have been, will be, could have been etc. In terms of the branching time system, this means that the whole structure of the branching time system follows from any single instant proposition. That is, the system has the nice and interesting property that the system as a whole is reflected – and in a sense even contained – in any basic part or element of the system. This insight might be conceived as a support of Prior's understanding of branching time as a conceptual structure consistent with presentism.

A basic assumption in Prior's worldview has to do with the notion of free choice. This should probably be understood as closely related to his ideas of ethics and responsibility. In general, Prior tried to establish a conceptual framework integrating fundamental notions of logic, ethics and time. In terms of branching time notions, it becomes obvious that we may not only ask which future developments are possible and which are necessary, but we may also want to investigate which of the possible futures we should choose and how this choice can be backed up by various kinds of ethical reasoning. (See [11: 65 ff].)

Through Prior's research it has become obvious that it is in fact possible to formulate a number of different branching time models. There is, however, still much work to be done if we want a full understanding of Prior's contributions and conceptual developments regarding the notion of branching time. Much of his material has still not been edited and published. In the pursuit of a deeper understanding of integrated conceptual framework of time and modality it is very

likely that we may benefit significantly from a careful study of his writings (including his Nachlass).

7 The Philosophical Significance of Branching Time

Although there is still much to obtain from a further investigation of Prior's writings regarding the understanding of branching time, it would be a misunderstanding to restrict the study of branching time to this historical perspective. It is certainly also important to discuss the philosophical significance of the idea of branching time from a more general point of view.

The notion of branching time is philosophically significant in that it, among others, seems to provide the conceptual ground for combining consistently strong intuitions concerning the existence of free will, and hence of moral responsibility, with the apparent existence of determined events or facts. In Prior's notion of branching time the future is "open" in the sense that it contains numerous possible continuations of the present state of the world, i.e. numerous possible alternative futures. As such the very structure of time allows for the exercising of free will, i.e. the exercise of the ability to shape the future through one's choices. At the same time, however, Prior's notion of branching time is consistent with the determinedness of events or facts. Thus, if determinism tentatively is defined such that a future event is determined if, and only if, its future occurrence and properties are fixed (entailed) by natural laws and a set of well-described conditions in the present, then the notion of branching time only presupposes that not *all* future events, or facts are determined. Alternatively determinism could be taken to be a global relationship holding between states of the world at different times. That is, determinism could tentatively be defined such that a future state of the world is determined if, and only if, the future state of the *entire* world is fixed (entailed) by natural laws and the present state of the *entire* world. Such a definition seems to make branching time collapse into linear time. Whether or not the notion of branching time is thought to provide an adequate model of the relationship between time, free will and determined events cannot be settled here. It is evident from these tentative considerations of different notions of determinism, though, that the notion of branching time is relevant for the longstanding philosophical attempt to understand and characterize this relationship.

A second area in which Priors notion of branching time is of more general philosophical relevance is in relation to the so-called A- and B-theories of time. The A- and B-theories hold opposing views concerning the question of how to characterize time. The A-theorist holds that positions in time are to be characterized using the tenses past, present and future as they are found in such expressions of ordinary language as 'it *was* the case', 'it *is* the case' and 'it *will* be the case'. The B-theorist, on the contrary, claims that positions in time are most adequately described using the relations 'earlier than' and/or 'later than'. The debate between the A- and B-theories has a semantical as well as an ontological dimension. On the one hand it concerns the possibility of reducing tensed expressions to tenseless ones and vice versa without a loss of meaning, and relatedly the possibility of giving tenseless truth-conditions for tensed statements and vice versa. Thus one may ask if the statement "Today is the first day of spring" is equivalent in meaning to a statement about the

day the statement was uttered, e.g. 'The 1st of March is the first day of spring', or if the latter statement lays down the truth-conditions of the former? Closely connected is the ontological dimension which concerns the question of whether the tensed or the tenseless orderings of time are the real and primary discriminations of time. Prominent in this discussion figures the notion of 'the passing of time' where this passing is taken to involve the coming into being of things and events. The point to be made here is simply that if there is a passing of time in this sense then it seems to favor the ontological priority of a tensed ordering of time simply because the tensed ordering of time straightforwardly can account for the moment of this becoming, namely the present. The B-theorist is here left with the choice of trying to provide a tenseless account of the 'now' or 'present' or simply dismiss this notion of passage of time and suggest an alternative. One such alternative is to replace the notion of becoming with that of change where change is an objects instantiation of incompatible properties at different times. This solution seems to imply that all statements claiming an entity to have a property at any given time t_n will have a truth-value although it will be dependent on n. Returning here to Prior's notion of branching time it is immediately clear that it feeds into this debate between the A- and B-theory. Thus it clearly involves the passage of time in the sense of future possibilities coming into existence in the present. Moreover, it involves becoming in the sense that contingent statements claiming an entity to have a property can only be true as and when the entity enters the present. Again, it is clear that to further advance this debate is beyond the scope of this paper, but it is also clear, however, that Prior's notion of branching time may have an important conceptual role to play in the debate between A- and B-theorist.

8 Branching Time and the User Experience

Branching time models as well as the distinction between A- and B-theoretical notations are in fact also directly applicable to the design of information spaces and user experience of such. Most information structures, such as web sites, are designed with specific goals in mind. The intention is to accommodate user's demands for navigation and overview, but also to guide users in certain directions. Typical examples include closing a sale, completing upload or download procedures, signing up for services, and finding pieces of information in accordance with the priority that information is given by the designers of the information space. Providing good navigational tool for end users has become a very important aspect of contemporary web design, displaying on the one hand the information architecture by means of menus and site maps, and on the other hand displaying dynamic information about the user's movement through the information space. For example, the path-type breadcrumbs seen on top of some web pages or lists of recently viewed items within a web site may provide the user with tools to keep track of movements and associated thoughts. Gathering and displaying information about user movement can clearly be accomplished by applying a B-theoretical perspective. A trail of breadcrumbs thus marks a route, consisting of one page visit after another, and no more than the before-after relation is needed to account for this. This procedure may be analyzed via a framework such as 'Temporal Concept Analysis' [12] and conceptualized as 'Life

Tracks'[13]. In this way, not only the structure and content of an information space but also actual user actions may be studied. This may certainly be useful, but if we want to focus on the actual user experience of moving through an information space the application of the various B-theoretical concepts will not be sufficient. This becomes very clear when we examine the cases just mentioned, such as completing a sale or the process of decision making based on interaction with a system. It is indeed a core conceptual trait of these processes that as the user moves through them, they are not yet completed. In consequence hereof, modeling the user experience must avail itself of the A-theoretical perspective in order to account for the privileged status of the 'now', situated between the past (the steps taken to get here) and the future, consisting of possible path of steps towards a final state of interaction. If we want to design a system which can convince the user to behave in a certain manner, then the communication of such a persuasive system must be based on an A-theoretical perspective. In the case of the HANDS system described in [14], the user is a teenager with an autism diagnosis who is facing a certain difficult situation. The system is supposed to persuade the user to handle the difficulty in a certain way. This means that the system should generate some sort of communication which the teenager will conceive as a strong case for choosing to follow the advice given by the system. In order to be persuasive in this manner, the communication has to be formulated in a way which is relevant for the teenager here and now, i.e. the persuasive communication has to be formulated within a tense-logical framework as seen from the current state. In other words, it must convince the teenager that by following the advice here and now he or she will obtain a desired outcome in the future.

Precisely the focus on various final states of interaction can benefit from the perspective offered in branching time models. There are two main reasons for this. In the first place, interaction design often allows for different paths through an information space – albeit with a few specific end-states as the desired ones. Closing a sale should end with a financial transaction but the items picked up by the client may vary, as does the sequence of selecting various items. In this and other decision making processes, the designer faces the challenge of presenting the user with appropriate information in all possible runs of the system. In the second place, a focus on the user experience offers a possibility of theorizing about the opportune moment for introducing new information. We refer to this as the principle of Kairos [14]: making good use of the balance between time, place, and circumstance. A good example of this is the algorithm employed by many web shops, displaying goods that are 'frequently bought with' the one you have already selected. Here, the suggestion of buying another item is presented precisely at the moment before initiating the check-out procedure. In the HANDS system [14], offering on time, on site support to mentally disadvantaged, it is imperative that the system has a fail-safe, e.g., a panic button displayed at appropriate times in every possible scenario that the user may end up in. This corresponds to saying that this solution must be present in all possible futures after initiating the sequence. This means that the possibility of getting external help must be present at every branch in the tree of possible solutions. But since offering expert guidance in real world scenarios is so vulnerable to change, it is vital to be able to incorporate dynamic information about the agents involved [15].

9 The Need for Further Investigation of Branching Time as a Conceptual Structure

Prior's ideas and findings are not fully known. There is much material from his hand which still has not been carefully studied. One of the topics which ought to be further studied in Prior's writings (including his Nachlass) would be his discussions regarding the development of the various ideas of branching time. This work should be done not only for historical reasons but also simply to obtain a better understanding of the branching time idea itself. It is certainly likely that one very attractive way to pursuit this goal would be to take advantage of the inspiration that can come from the work already done by Prior. It would certainly be very useful to have an edition of the collected works of A.N. Prior. This would, however, be a very ambitious goal given the present state of the material and the fact that many relevant papers and letters have to be collected from libraries and collections all over the world, although most of the material is included in the Prior collection at Bodleian Library in Oxford. One of the aspects of Prior's logic which needs more investigation has to do with his idea of a tensed ontology. How can we account for the existence of objects given that only the present exists? In various papers and chapters of his books Prior discussed this problem. One of the tasks he worked with during the last years before his sudden and early death in 1969 had to do with relations between his presentism and his understanding of the existence of objects. It would certainly be an interesting challenge to develop a theory of tensed ontology corresponding to Prior's tentative ideas in this regard.

Acknowledgments

We are very grateful to professor Saul Kripke for kind co-operation and for giving us access to the two letters he received from Prior in 1958 and to The National Library of Finland for giving access to Prior's letters to Henrik von Wright.

References

1. Prior, A.N.: A Statement of Temporal Realism. Published in P. Øhrstrøm's edition in Copeland 1996, pp. 45–46 (1996)
2. Prior, A.N.: Some Free thinking about Time. Published in P. Øhrstrøm's edition in Copeland 1996, pp. 47–51 (1996)
3. Copeland, J. (ed.): Logic and Reality: Essays on the Legacy of Arthur Prior. Oxford University Press, Oxford (1996)
4. Prior, A.N.: The Interpretation of two Systems of Modal Logic. The Journal of Computing Systems 2, 201–208 (1954)
5. Prior, A.N.: Past, Present and Future. Clarendon Press, Oxford (1967)
6. Øhrstrøm, P., Hasle, P.: Temporal Logic - From Ancient Ideas to Artificial Intelligence. Kluwer Academic Publishers, Dordrecht (1995)
7. Prior, A.N.: Diodoran Modalities. The Philosophical Quarterly 5, 205–213 (1955)
8. Prior, A.N.: Time and Modality, Oxford (1957)
9. Øhrstrøm, P.: In Defense of the Thin Red Line: A Case for Ockhamism. Humana Mente 8, 17–32 (2009)

10. Øhrstrøm, P.: Time and Logic: A.N. Prior's Formal Analysis of Temporal Concepts. In: Proceedings of Formal Concept Analysis: 7th International Conference, ICFCA 2009 Darmstadt, Germany, May 2009, pp. 66–81. Springer, Heidelberg (2009)
11. Prior, A.N.: Papers on Time and Tense. In: Hasle, P., et al. (eds.), Oxford University Press, Oxford (2003)
12. Wolff, K.E.: Basic Notions in Conceptual Semantic Systems. In: Gély, A., Kuznetsov, S.O., Nourine, L., Schmidt, S.E. (eds.) Contributions to ICFCA 2007, 5th International Conference on Formal Concept Analysis, Clermont-Ferrand, France, pp. 97–120 (2007)
13. Wolff, K.E.: States, Transitions, and Life Tracks in Temporal Concept Analysis. In: Ganter, B., Stumme, G., Wille, R. (eds.) Formal Concept Analysis. LNCS (LNAI), vol. 3626, pp. 127–148. Springer, Heidelberg (2005)
14. Schärfe, H., Øhrstrøm, P., Gyori, M.: A Conceptual Analysis of Difficult Situations - developing systems for teenagers with ASD. In: CEUR Workshop Proceedings of International Conference on Conceptual Structures - ICCS 2009, Moscow, Russia (2009)
15. Pertou, M.E., Schärfe, H.: Adaptive Persuasive Scripts. In: The Society for the Study of Artificial Intelligence and the Simulation of Behaviour AISB 2009. Proceedings of the Symposium Persuasive Technology and Digital Behaviour Intervention Symposium, Edinburgh, Scotland (2009)

[L1] Kripke, Saul. Letter to A.N. Prior. The Prior Collection, Bodleian Library, Oxford (September 3, 1958)
[L2] Prior, A.N. Letter to Saul Kripke. In Saul Kripke's possession (September 10, 1958)
[L3] Kripke, Saul. Letter to A.N. Prior. The Prior Collection, Bodleian Library, Oxford (October 13, 1958)
[L4] Prior, A.N. Letter to Saul Kripke. In Saul Kripke's possession (October 27, 1958)
[L5] Prior, A.N. Letter to Henrik von Wright. The Henrik von Wright collection, Helsinki, The National Library of Finland, University of Helsinki (February 1, 1966)

Formal Concept Analysis in Knowledge Discovery: A Survey

Jonas Poelmans[1], Paul Elzinga[3], Stijn Viaene[1,2], and Guido Dedene[1,4]

[1] K.U.Leuven, Faculty of Business and Economics, Naamsestraat 69, 3000 Leuven, Belgium
[2] Vlerick Leuven Gent Management School, Vlamingenstraat 83, 3000 Leuven, Belgium
[3] Amsterdam-Amstelland Police, James Wattstraat 84, 1000 CG Amsterdam, The Netherlands
[4] Universiteit van Amsterdam Business School, Roetersstraat 11 1018 WB Amsterdam, The Netherlands
{Jonas.Poelmans,Stijn.Viaene,Guido.Dedene}@econ.kuleuven.be, Paul.Elzinga@amsterdam.politie.nl

Abstract. In this paper, we analyze the literature on Formal Concept Analysis (FCA) using FCA. We collected 702 papers published between 2003-2009 mentioning Formal Concept Analysis in the abstract. We developed a knowledge browsing environment to support our literature analysis process. The pdf-files containing the papers were converted to plain text and indexed by Lucene using a thesaurus containing terms related to FCA research. We use the visualization capabilities of FCA to explore the literature, to discover and conceptually represent the main research topics in the FCA community. As a case study, we zoom in on the 140 papers on using FCA in knowledge discovery and data mining and give an extensive overview of the contents of this literature.

Keywords: Formal Concept Analysis (FCA), knowledge discovery, text mining, exploratory data analysis, systematic literature overview.

1 Introduction

Formal Concept Analysis (FCA) was invented in the early 1980s by Rudolf Wille as a mathematical theory (Wille, 1982). FCA is concerned with the formalization of concepts and conceptual thinking and has been applied in many disciplines such as software engineering, knowledge discovery and information retrieved during the last 15 years. The mathematical foundation of FCA is described by Ganter (1999).

A textual overview of part of the literature published until the year 2004 on the mathematical and philosophical background of FCA, some of the applications of FCA in the information retrieval and knowledge discovery field and in logic and AI was given in Priss (2006). An overview of available FCA software is provided by Tilley (2004). Carpineto (2004) provides an overview of FCA applications in information

M. Croitoru, S. Ferré, and D. Lukose (Eds.): ICCS 2010, LNAI 6208, pp. 139–153, 2010.

retrieval. In Tilley (2007), an overview of 47 FCA based software engineering papers was given. The authors categorized these papers according to the 10 categories as defined in the ISO 12207 software engineering standard and visualize them in a concept lattice. In Lakhal et al. (2005), a survey on FCA-based association rule mining techniques is given.

In this paper, we describe how we used FCA to create a visual overview of the existing literature on concept analysis published between the years 2003 and 2009. The core contributions of this paper are as follows. We visually represent the literature on FCA using concept lattices, in which the objects are the scientific papers and the attributes are the relevant terms available in the title, keywords and abstract of the papers. We developed a toolset with a central FCA component that we use to index the papers with a thesaurus containing terms related to FCA research and to generate the lattices. As a case study, we zoom in on the 140 papers published between 2003 and 2009 on using FCA in knowledge discovery and data mining.

The remainder of this paper is composed as follows. In section 2 we introduce the essentials of FCA theory and the knowledge browsing environment we developed to support this literature analysis. In section 3 we describe the dataset used. In section 4 we visualize the literature using FCA lattices and we summarize the papers published in the knowledge discovery field. Section 5 concludes the paper.

2 Formal Concept Analysis

Formal Concept Analysis can be used as an unsupervised clustering technique. The starting point of the analysis is a database table consisting of rows G (i.e. objects), columns M (i.e. attributes) and crosses $I \subseteq G \times M$ (i.e. relationships between objects and attributes). The mathematical structure used to reference such a cross table is called a formal context (G, M, I). In this paper, scientific papers (i.e. the objects) are related (i.e. the crosses) to a number of terms (i.e. the attributes); here a paper is related to a term if the title or abstract of the paper contains this term. Scientific papers containing terms from the same term-clusters are grouped in concepts. Given a formal context, FCA then derives all concepts from this context and orders them according to a subconcept-superconcept relation. This results in a line diagram (a.k.a. lattice). The details of using FCA for text mining can be found in (Poelmans 2009).

2.1 FCA Software

We developed a knowledge browsing environment to support our literature analysis process. One of the central components of our text mining environment is the thesaurus containing the collection of terms describing the different FCA-related research topics. The initial thesaurus was constructed based on expert prior knowledge and was incrementally improved by analyzing the concept gaps and anomalies in the resulting lattices. The thesaurus is a layered thesaurus containing multiple abstraction levels. The first and finest level of granularity contains the search terms of which most are grouped together based on their semantical meaning to form the term clusters at the second level of granularity.

The papers that were downloaded from the World Wide Web (WWW) were all formatted in pdf. These pdf files were converted to ordinary text and the abstract, title and keywords were extracted. The open source tool Lucene was used to index the extracted parts of the papers using the thesaurus. The result was a cross table describing the relationships between the papers and the term clusters or research topics from the thesaurus. This cross table was used as a basis to generate the lattices.

3 Dataset

This Systematic Literature Review (SLR) has been carried out by considering a total of 702 papers related to FCA published between 2003 and 2009 in the literature and extracted from the most relevant scientific sources. The sources that were used in the search for primary studies contain the work published in those journals, conferences and workshops which are of recognized quality within the research community. These sources are: *IEEE Computer Society, ACM Digital Library, Sciencedirect, Springerlink, EBSCOhost, Google Scholar, Repositories of the ICFCA, ICCS and CLS conference*.

Other important sources such as *DBLP* or *CiteSeer* were not explicitly included since they were indexed by some of the mentioned sources (e.g. *Google Scholar*). In the selected sources we used various search strings including "Formal Concept Analysis ", "FCA", "concept lattices", "Temporal Concept Analysis". We ensured that papers that appeared in multiple sources were only taken into account once.

4 Studying the Literature Using FCA

The 702 papers are grouped together according to a number of features within the scope of FCA research. The FCA lattices facilitate our exploration and analysis of the literature. The lattice in Fig. 1 contains 8 categories under which 53% of the 702 FCA papers can be categorized. Knowledge discovery is the most popular research theme covering 20% of the papers and will be analyzed in detail in section 4.1. Recently, improving the scalability of FCA to larger and complex datasets emerged as a new research topic covering 5% of the 702 FCA papers. In particular, we note that almost half of the papers dedicated to this topic work on issues in the KDD domain. Another important research topic in the FCA community is information retrieval covering 15% of the papers. 25 of the papers on information retrieval describe a combination with KDD approach and in 18 IR papers authors make use of ontologies. 15 IR papers deal with the retrieval of software structures such as software components. In 13% of the FCA papers, FCA is used in combination with ontologies or for ontology engineering. In 11% of the papers, the extension of FCA to deal with fuzzy attributes is investigated. The temporal variant of FCA received only minor attention, covering 1% of the papers. Other important topics are using FCA in software engineering (15%) and for classification (7%).

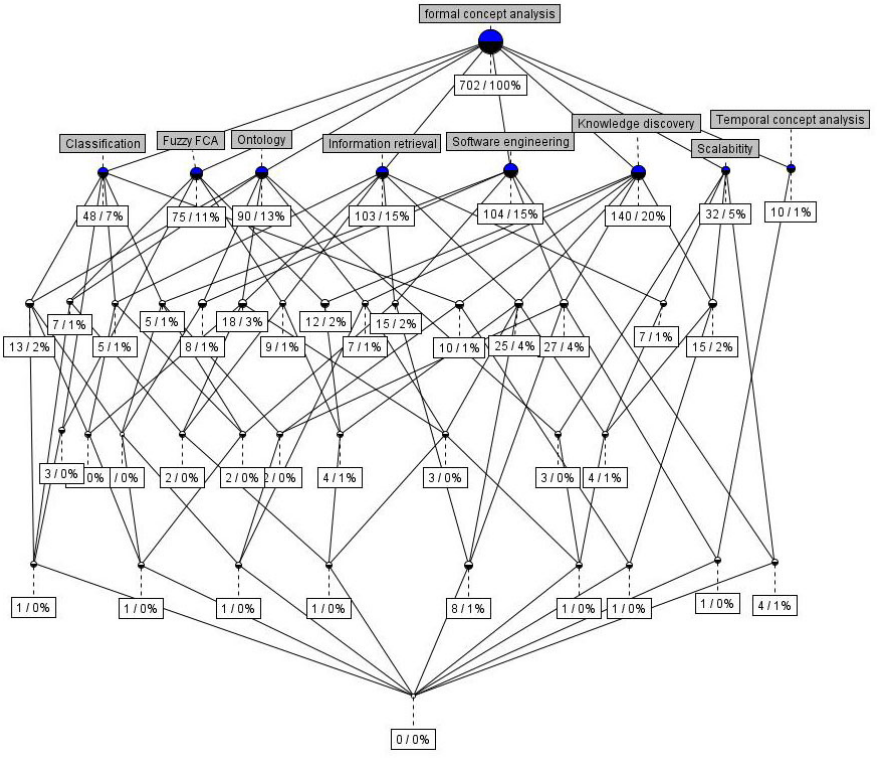


Fig. 1. Lattice containing 702 papers on FCA

4.1 Knowledge Discovery and Data Mining

In the past, the focus in knowledge discovery and data mining was on developing fully automated tools and techniques that extract new knowledge from data. Unfortunately, these techniques allowed almost no interaction between the human actor and the tool and failed at incorporating valuable expert knowledge into the discovery process (Keim 2002), which is needed to go beyond uncovering the fool's gold. These techniques assume a clear definition of the concepts available in the underlying data which is often not the case. Visual data exploration (Eidenberger 2004) and visual analytics (Thomas 2005) are especially useful when little is known about the data and exploration goals are vague. Since the user is directly involved in the exploration process, shifting and adjusting the exploration goals is automatically done if necessary.

In Conceptual Knowledge Processing (CKP) the focus lies on developing methods for processing information and knowledge which stimulate conscious reflection, discursive argumentation and human communication (Wille 2006). An important subfield of CKP is Conceptual Knowledge Discovery. FCA is particularly suited for exploratory data analysis because of its human-centeredness (Correira 2003). The generation of knowledge is promoted by the FCA representation that makes the

inherent logical structure of the information transparent. The system TOSCANA has been used as a knowledge discovery tool in various research and commercial projects (Stumme 1998).

About 74% of the FCA papers on KDD are covered by the research topics in Figure 2. In section 4.1.1 we zoom in on the 35 papers (25%) in the field of association rule mining. 19% of the KDD papers are on using FCA in the discovery of structures in software and will be described in section 4.1.2. Section 4.1.3 describes the 9% of papers on applications of FCA in web mining. Section 4.1.4 discusses some of the extensions of FCA theory for knowledge discovery (11% of papers). In section 4.1.5 we describe some of the applications of FCA in biology, chemistry and medicine covering 10% of the KDD papers. The applications on using Fuzzy FCA for KDD, covering 9% of the papers, will be discussed in section 4.1.6.

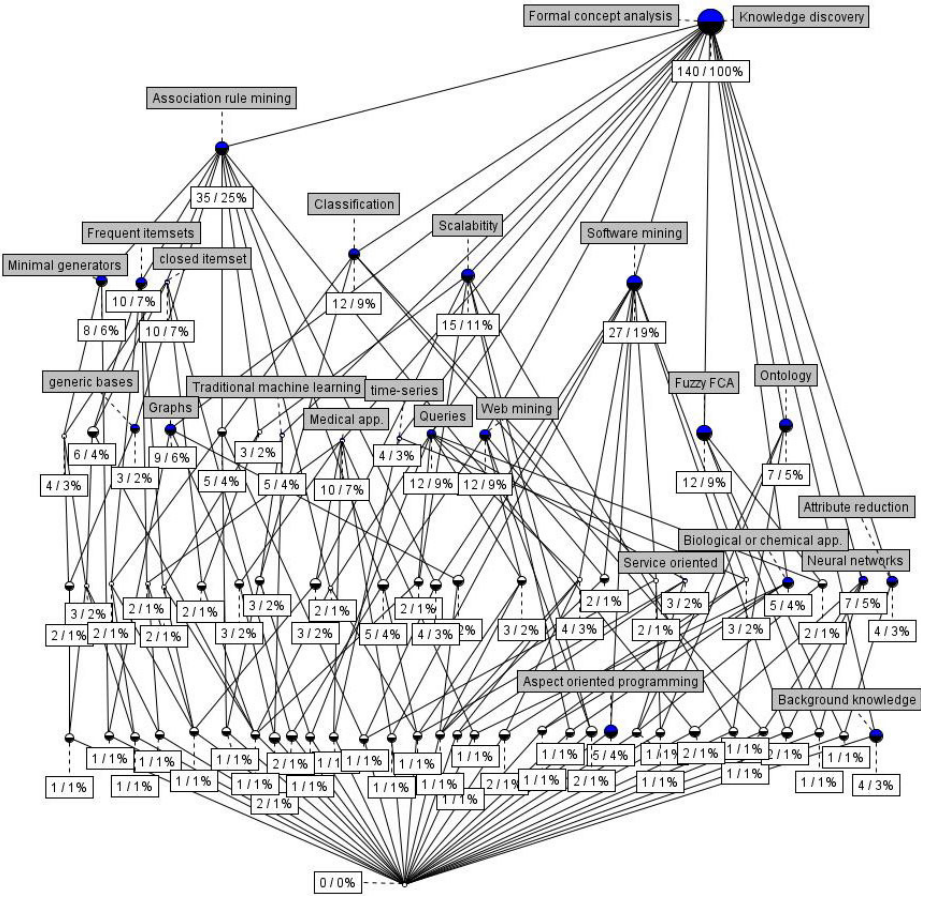


Fig. 2. Lattice containing 140 papers on using FCA in KDD

4.1.1 Association Rule Mining

Association rule mining covers 25% of the papers using FCA for KDD. Association rule mining from a transaction database requires the detection of frequently occurring patterns called frequent itemsets (FI). Recent approaches for FI mining use the closed itemset paradigm to limit the mining effort to the subset of frequent closed itemsets (FCIs). The intent of a concept *C* is called a closed itemset and consists of the maximum set of attributes that characterizes *C*. Several FCA-based algorithms were developed for mining frequent closed itemsets including CLOSE, PASCAL, CHARM, CLOSET and TITANIC (Stumme 2002) which mines frequent closed itemsets by constructing an iceberg concept lattice. In Qi et al (2004) an algorithm Closearcher is proposed based on FCA for mining frequent patterns.

The minimal generators for a concept *C* are the minimal subsets of *C*'s intent which can similarly characterize *C*. Nehmé (2005) proposes a novel method for computing the minimal generator family. Tekaya et al. (2005) propose an algorithm called GenAll to build an FCA lattice in which each concept is decorated by its minimal generators with the aim to derive generic bases of association rules. Generic bases constitute reduced sets of association rules and preserve the most relevant rules without loss of information. The GenAll algorithm further improves on the algorithm presented by Nourine et al. (1999). In Hamrouni, (2005), the extraction of reduced size generic bases of association rules is discussed to decrease the overwhelming number of association rules resulting from ARM. Hamrouni (2005a) proposes an algorithm called PRINCE which builds a minimal generator lattice from which the derivation of the generic association rules becomes straightforward. Dong et al. (2005) introduce the succinct system of minimal generators (SSMG) as a minimal representation of the minimal generators of all concepts, and gives an efficient algorithm for mining SSMGS. The SSMGS are also used for losslessly reducing the size of the representation of all minimal generators. Hamrouni et al. (2007) present a new sparseness measure for formal contexts using the framework of SSMGS. Their measure is an aggregation of two complementary measures, namely the succinctness and compactness measures of each equivalence class, induced by the closure operator. This is important for the performance of frequent closed itemset mining algorithms which is closely dependent on the type of handled extraction context, i.e. sparse or dense. Hermann et al. (2008) investigate the computational complexity of some of the problems related to generators of closed itemsets. The authors also present an incremental polynomial time algorithm that can be used for computing all minimal generators of an implication-closed set. In Yahia et al. (2004), inference axioms are presented for deriving all association rules from generic bases.

Valtchev et al. (2004), discuss the existing FCA-based data association rule mining techniques and provide guidelines for the design of novel ones to be able to apply FCA in a larger set of situations. They also propose two online methods for computing the minimal generators of a closure system. Gupta et al. (2005) discuss how classification rules based on association rules can be generated using concept lattices. Valtchev et al. (2008) show how FCIs can be mined incrementally yet efficiently whenever a new transaction is added to a database whose mining results are available. In Quan et al (2009), a new cluster-based method is proposed for mining conceptual association rules. Maddouri (2005) discusses the discovery of association rules and proposes a new approach to mine interesting itemsets as the

optimal concepts covering a binary table. Maddouri et al. (2006) summarizes many of the statistical measures introduced for selecting pertinent formal concepts. Maddouri et al (2009) present a method for building only a part of the lattice including the best concepts, which are used as classification rules.

Wollbold et al (2008) make use of FCA to construct a knowledge base consisting of a set of rules such that reasoning over temporal dependencies within gene regulatory networks is possible. Zhou et al (2005) use FCA to mine association rules from web logs, which can be used for online applications such as web recommendation and personalization. Richards et al. (2003) explores the possibilities of using FCA for mining knowledge and reorganizing this knowledge into an abstraction hierarchy and to discover higher-level concepts in the knowledge. Richards et al (2003a) discuss the discovery of multi-level knowledge from rule bases which is important to allow queries at and across different levels of abstraction. FCA is used to develop an abstraction hierarchy and the approach is applied to knowledge bases from the domain of chemical pathology. In Zarate et al. (2009), FCA is used to extract and represent knowledge in the form of a non-redundant canonical rule base with minimal implications from a trained ANN.

4.1.2 Software Mining

Software mining covers 19% of the 140 KDD papers and describes how FCA can be used to gain insight in amongst others software source code. In Cole et al. (2005), FCA is used to conceptually analyse relational structures in software source code and to detect unnecessary dependencies between software parts. In Cellier et al. (2008), FCA is used in combination with association rules for fault localization in software source code. Wermelinger (2009) uses FCA lattices to visualize the relations between the software artefacts and the developers who should fix the bugs in them. In Eisenbarth (2003), a technique is presented for reconstructing the mapping for features that are triggered by the user to the source code of the system. Mens et al (2005) use FCA to delve a system's source code for relevant concepts of interest: what concerns are addressed in the code, what patterns, coding idioms and conventions have been adopted and where and how are they implemented.

Crosscutting concerns, i.e. functionalities that are not assigned to a single modular unit in the implementation are one of the major problems in software evolution. Aspect Oriented Programming offers mechanisms to factor them out into a modular unit, called an aspect. In Tonella et al. (2004), aspect identification in existing code is supported by means of dynamic code analysis. Execution traces are generated for the use cases that exercise the main functionalities of a given application. The relationship between execution traces and executed computational units is subjected to concept analysis. In the resulting lattice, potential aspects are detected. Yang et al (2008), discuss an aspect-mining approach in which execution profiles of legacy systems are analyzed using concept lattices to identify the invoked computational units that traverse system's use case models. They can be abstracted into early-aspects for re-engineering of the legacy system with AOSD. Qu (2007) also discusses the use of FCA for aspect mining to identify crosscutting concerns in a system thereby improving the system's comprehensibility and enabling migration of existing (object-oriented) programs to aspect-oriented ones. Breu et al (2006) mined aspects from Eclipse by analyzing where developers added code to the program over time. In Del

Grosso et al. (2007), an approach is proposed to identify from database-oriented applications, pieces of functionality to be potentially exported as services.

Role Based Access Control (RBAC) is a methodology for providing users in an IT system with specific permissions like read or write. Molloy et al. (2008) use FCA for mining roles from user-permission and user-attribute information to complement the costly top-down approaches for RBAC. Dau et al. (2009) apply FCA in combination with Description Logics to capture the RBAC constraints and for deriving additional constraints.

4.1.3 Web Mining

Web mining and improving the quality of web search results is investigated in 8% of the KDD papers. Periodic web personalization aims to recommend the most relevant resources to a user during a specific time period by analyzing the periodic access patterns of the user from web usage logs. Beydoun et al. (2007) introduce a system which captures user trails as they search the internet. They construct a semantic web structure from the trails and this semantic web structure is expressed as a conceptual lattice guiding future searches. Beydoun (2009) further investigates the possibilities of FCA for processing students virtual surfing trails to express and exploit the dependencies between visited web-pages to yield subsequent and more effective focused search results. He (2007) also proposes a method for automatically mining and acquiring web user profiles using FCA. Okubo et al. (2006) show how FCA can be used for the conceptual clustering of web documents and to provide a conceptual meaning for each document cluster. Myat at al (2005) use FCA for conceptual document clustering to manage the information published on the World Wide Web. Wang et al. (2008) give a method for using FCA for developing a topic-specific web crawler for use in web data mining.

Du et al. (2009) present a method based on FCA for mining association rules that can be used to match user queries with web pages to avoid returning irrelevant web pages for search engine results. Hsieh et al. (2007) propose a knowledge acquisition system which dynamically constructs the relationships and hierarchy of concepts in a query-based ontology to provide answers for user's queries. Kim et al. (2007) discuss a novel approach using FCA to build a contextualized folksonomy and concept hierarchies from tags of blogosphere.

4.1.4 Extending FCA for Data Mining

In the last years, multiple extensions have been introduced into the literature that improve traditional FCA theory's applicability to knowledge discovery problems. Belohlavek et al. (2009) emphasizes the need for taking into account background knowledge in FCA. They present an approach for modeling background knowledge that represents user's priorities regarding attributes and their relative importance. Only those concepts that are compatible with user's priorities are considered as relevant and extracted from the data. In Pogel et al. (2008), FCA is used in combination with a tag context for formally incorporating important kinds of background knowledge. The results are Generalized Contingency Structures and Tagged Contingency structures which can be used for data summarization in epidemiology. In Poelmans et al. (2009), FCA is used in combination with Emergent Self Organising Maps for detecting domestic violence in the unstructured text of police reports.

In Besson et al. (2006), FCA is extended to cope with faults and to improve formal concepts towards fault tolerance. Pfaltz (2007) extends FCA to deal with numerical values. Valverde-Albacete et al. (2006) introduced a generalization of FCA for data mining applications called *K*-Formal Concept Analysis. This idea was further developed in Valverde-Albacete et al. (2007) where the lattice structure for such generalized contexts was introduced. This research topic was further investigated in Valverde-Albacete et al (2008). Hashemi (2004) proposes a method for efficiently creating a new lattice from an already existing one when the data granularity is changed. Lei (2007) introduces the notion of extended many-valued context to avoid the generation of a large one-valued context in document knowledge discovery. In Deogun et al (2003) FCA is complemented with Bacchus probability logic, which makes use of statistical and propositional probability inference. The authors introduce a new type of concept called "previously unknown and potentially useful" and formalize KDD as a process to find such concepts.

In its classical form FCA considers attributes as a non-ordered set. When attributes of the context are partially ordered to form a taxonomy, conceptual scaling allows the taxonomy to be taken into account by producing a context completed with all attributes deduced from the taxonomy. In Cellier et al (2008) an algorithm is proposed to learn concept-based rules in the presence of a taxonomy.

Another FCA research topic is attribute reduction. Shao et al. (2008) show how to remove redundant attributes from real set formal contexts without any loss of knowledge. Wu et al. (2009) discuss the application of viewing data at different levels of granularity to construct a granular data structure which can be used for knowledge reduction in FCA. Wang et al (2008) deal with approaches to generalized attribute reduction in a consistent decision formal context. Ganter et al (2008) describe how scaled many-valued contexts of FCA may make feature selection easier.

4.1.5 FCA Mining Applications in Biology and Medicine

10% of the KDD papers describe applications of FCA in biology, chemistry or medicine. In Sato et al (2007), FCA is used to cluster time-series medical data and to analyze these clusters. Sklenar (2005) used FCA to evaluate epidemiological questionnaire physical activity data to find dependencies between demographic data and degree of physical activity. In Kaytone et al. (2009), FCA is used for mining and clustering gene expression data. Fu (2006) applies FCA as a tool for analysis and visualization of data in a digital ecosystem. Maddouri (2004) outlines a new incremental learning approach based on FCA that supports incremental concept formation and applies it to the problem of cancer diagnosis. The incremental approach has the advantage of handling both the problem of data addition, data deletion, data update, etc.

4.1.6 Fuzzy FCA in KDD

In Fuzzy FCA, each table entry contains a truth degree to which an attribute applies to an object. 9% of the papers use Fuzzy FCA for KDD. In Chou et al. (2008), Fuzzy FCA is used for tag mining, i.e. to analyze the relationships between semantic tags and Web APIs. In Fenza et al. (2008), Fuzzy FCA is used for the discovery of semantic web services. Zhou et al. (2006) use Fuzzy FCA to construct a user behaviour model from web usage logs to identify the resources that the user is most

likely interested in during a given period. Fenza et al (2009) present a system which uses fuzzy FCA for supporting the user in the discovery of semantic web services. Through a concept-based navigation mechanism, the user discovers conceptual terminology associated to the web resources and uses it to generate an appropriate service request. Yan (2007) uses Fuzzy Set Theory to extend the many-valued context from FCA. This fuzzy many-valued context can then be used for document knowledge discovery.

5 Conclusion

We found FCA to be an interesting instrument to explore the literature on concept analysis. Over 700 papers have been published over the past 7 years on FCA and 140 zoomed in on applying FCA in KDD. The main research topics in this area are association rule mining, software mining, web mining, KDD in medicine and biology, using Fuzzy FCA for KDD and to extend the capabilities of traditional FCA theory for data mining. In the future, we will host the references and links to the article on a public interface. Further research consists of doing this same analysis for information retrieval, scalability and ontology construction.

Acknowledgements

Jonas Poelmans is aspirant of the "Research Foundation - Flanders" or "Fonds voor Wetenschappelijk Onderzoek - Vlaanderen".

References

1. Belohlavek, R., Vychodil, V.: Formal Concept Analysis With Background Knowledge: Attribute Priorities. IEEE Trans. Syst., man, & cyb. - C: App. & rev. 39(4) (2009)
2. Besson, J., Robardet, C., Boulicaut, J.F.: Mining a New Fault-Tolerant Pattern Type as an Alternative to Formal Concept Discovery. In: Schärfe, H., Hitzler, P., Øhrstrøm, P. (eds.) ICCS 2006. LNCS (LNAI), vol. 4068, pp. 144–157. Springer, Heidelberg (2006)
3. Beydoun, G.: Using Formal Concept Analysis towards Cooperative E-Learning. In: Richards, D., Kang, B.-H. (eds.) PKAW 2008. LNCS (LNAI), vol. 5465, pp. 109–117. Springer, Heidelberg (2009)
4. Beydoun, G., Kultchitsky, R., Manasseh, G.: Evolving semantic web with social navigation. Expert Systems with Applications 32, 265–276 (2007)
5. Breu, S., Zimmermann, T., Lindig, C.: Mining Eclipse for Cross-Cutting Concerns. In: MSR 2006, Shanghai, China, May 22-23 (2006)
6. Carpineto, C., Romano, G.: Concept data analysis: Theory and applications. John Wiley & Sons, Chichester (2004)
7. Cellier, P., Ferré, S., Ridoux, O., Ducasse, M.: A parameterized algorithm to explore formal contexts with a taxonomy. Int. J. Found. of Comp. Sc. 19(2), 319–343 (2008)
8. Chang, Y.H.: Automatically constructing a domain ontology for document classification. In: 6th Int. Conf. on Machine Learning and Cybernetics, Hong Kong (2007)
9. Chou, C.Y., Mei, H.: Analyzing Tag-based Mashups with Fuzzy FCA. In: IEEE Int. Symposium on Service-Oriented System Engineering (2008)

10. Cole, R., Becker, P.: Navigation Spaces for the Conceptual Analysis of Software Structure. In: Ganter, B., Godin, R. (eds.) ICFCA 2005. LNCS (LNAI), vol. 3403, pp. 113–128. Springer, Heidelberg (2005)
11. Cole, R., Becker, P.: Navigation Spaces for the Conceptual Analysis of Software Structure. In: Ganter, B., Godin, R. (eds.) ICFCA 2005. LNCS (LNAI), vol. 3403, pp. 113–128. Springer, Heidelberg (2005)
12. Cole, R., Eklund, P., Stumme, G.: Document retrieval for e-mail search and discovery using Formal Concept Analysis. App. Art. Intel. 17, 257–280 (2003)
13. Cole, R., Tilley, T., Ducrou, J.: Conceptual Exploration of Software Structure: A Collection of Examples. In: Belohlavek, R., Snasel, V. (eds.) CLA, pp. 135–148 (2005)
14. Correira, J.H., Stumme, G., Wille, R., Wille, U.: Conceptual knowledge discovery - a human-centered approach. Applied Artificial Intelligence 17, 281–302 (2003)
15. Dau, F., Knechte, M.: Access Policy Design Supported by FCA Methods. In: Rudolph, S., Dau, F., Kuznetsov, S.O. (eds.) ICCS 2009. LNCS (LNAI), vol. 5662, pp. 141–154. Springer, Heidelberg (2009)
16. Del Grosso, C., Penta, M.D., Guzman, I.G.R.: An approach for mining services in database-oriented applications. In: 11th IEEE Eur. Conf. on Software Maintenance and Reeng. (2007)
17. Deogun, J., Jiang, L., Xie, Y., Raghavan, V.: Probability Logic Modeling of Knowledge Discovery in Databases. In: Zhong, N., Raś, Z.W., Tsumoto, S., Suzuki, E. (eds.) ISMIS 2003. LNCS (LNAI), vol. 2871, pp. 402–407. Springer, Heidelberg (2003)
18. Dong, G., Jiang, C., Pei, J., Li, J., Wong, L.: Mining Succinct Systems of Minimal Generators of Formal Concepts. In: Zhou, L.-z., Ooi, B.-C., Meng, X. (eds.) DASFAA 2005. LNCS, vol. 3453, pp. 175–187. Springer, Heidelberg (2005)
19. Du, Y.J., Li, H.M.: Strategy for Mining Association Rules for Web Pages Based on Formal Concept Analysis. Elsevier, Amsterdam (2009), doi:10.1016/j.asoc.2009.09.007
20. Ducrou, J.: DVDSleuth: A Case Study in Applied Formal Concept Analysis for Navigating Web Catalogs. In: Priss, U., Polovina, S., Hill, R. (eds.) ICCS 2007. LNCS (LNAI), vol. 4604, pp. 496–500. Springer, Heidelberg (2007)
21. Eidenberger, H.: Visual Data Mining. In: Proceedings SPIE Optics East Conf., Philadelphia, October 26-28, vol. 5601, pp. 121–132 (2004)
22. Eisenbarth, T., Koschke, R., Simon, D.: Locating Features in Source Code. IEEE Transactions on Software Engineering 29(3) (March 2003)
23. Eklund, P., Ducrou, J.: Navigation and Annotation with Formal Concept Analysis. In: Richards, D., Kang, B.-H. (eds.) PKAW 2008. LNCS (LNAI), vol. 5465, pp. 118–121. Springer, Heidelberg (2009)
24. Eklund, P., Wille, R.: Semantology as Basis for Conceptual Knowledge Processing. In: Kuznetsov, S.O., Schmidt, S. (eds.) ICFCA 2007. LNCS (LNAI), vol. 4390, pp. 18–38. Springer, Heidelberg (2007)
25. Fenza, G., Loia, V., Senatore, S.: Concept Mining of Semantic Web Services By Means of Extended Fuzzy Formal Concept Analysis (FFCA). In: Int. Conf. Syst., Man & Cyb. (2008)
26. Fenza, G., Senatore, S.: Friendly web services selection exploiting fuzzy formal concept analysis. In: Soft Computing. Springer, Heidelberg (2009)
27. Fu, H.: Formal Concept Analysis for Digital Ecosystem. In: Proc. of the 5th Int. Conf. or Machine Learning and Applications (ICMLA 2006). IEEE, Los Alamitos (2006)
28. Ganter, B., Kuznetsov, S.O.: Formalizing Hypotheses with Concepts. In: Ganter, B., Mineau, G.W. (eds.) ICCS 2000. LNCS (LNAI), vol. 1867, pp. 342–356. Springer, Heidelberg (2000)

29. Ganter, B., Kuznetsov, S.O.: Scale Coarsening as Feature Selection. In: Medina, R., Obiedkov, S. (eds.) ICFCA 2008. LNCS (LNAI), vol. 4933, pp. 217–228. Springer, Heidelberg (2008)
30. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical foundations. Springer, Heidelberg (1999)
31. Gupta, A., Kumar, N., Bhatnagar, V.: Incremental Classification Rules Based on Association Rules Using Formal Concept Analysis. In: Perner, P., Imiya, A. (eds.) MLDM 2005. LNCS (LNAI), vol. 3587, pp. 11–20. Springer, Heidelberg (2005)
32. Hamrouni, T., Yahia, S.B., Nguifo, E.M.: Towards a Finer Assessment of Extraction Contexts Sparseness. In: 18th Int. Workshop on Database and Expert Systems Applications (2007)
33. Hamrouni, T., Yahia, S.B., Slimani, Y.: Avoiding the itemset closure computation pitfall. In: Belohlavek, R., Snasel, V. (eds.) CLA 2005, pp. 46–59 (2005)
34. Hamrouni, T., Yahia, S.B., Slimani, Y.: Prince: An Algorithm for Generating Rule Bases Without Closure Computations. In: Tjoa, A.M., Trujillo, J. (eds.) DaWaK 2005. LNCS, vol. 3589, pp. 346–355. Springer, Heidelberg (2005a)
35. Hashemi, R.R., De Agostino, S., Westgeest, B., Talburt, J.R.: Data Granulation and Formal Concept Analysis. In: Processing NAFIPS 2004, IEEE Annual Meeting of the Fuzzy Information, vol. 1, pp. 79–83 (2004)
36. He, H., Hai, H., Rujing, W.: FCA –Based Web User Profile Mining for Topics of Interest. In: Int. Conf. on Integation Technology, Shenzhen, China, March 20-24 (2007)
37. Hermann, M., Sertkaya, B.: On the Complexity of Computing Generators of Closed Sets. In: Medina, R., Obiedkov, S. (eds.) ICFCA 2008. LNCS (LNAI), vol. 4933, pp. 158–168. Springer, Heidelberg (2008)
38. Hsieh, T.C., Tsai, K.H., Chen, C.L., Lee, M.C., Chiu, T.K., Wang, T.: Query-based ontology approach for semantic search. In: 6th Int. Conf. on Machine Learning & Cyb. (2007)
39. Kaytone, M., Duplessis, S., Kuznetsov, S.O., Napoli, A.: Two FCA-Based Methods for Mining Gene Expression Data. In: Ferre, S., et al. (eds.) ICFCA. LNCS (LNAI), vol. 5548, pp. 251–266. Springer, Heidelberg (2009)
40. Keim, D.A.: Information visualization and visual data mining. IEEE Transactions on Visualization and Computer Graphics 8(I) (2002)
41. Kim, H.L., Hwang, S.H., Kim, H.G.: FCA-based Approach for Mining Contextualized Folksonomy. In: SAC 2007, Seoul, Korea, March 11-15 (2007)
42. Lakhal, L., Stumme, G.: Efficient Mining of Association Rules Based on Formal Concept Analysis. In: Ganter, B., Stumme, G., Wille, R. (eds.) Formal Concept Analysis. LNCS (LNAI), vol. 3626, pp. 180–195. Springer, Heidelberg (2005)
43. Lei, Y., Cao, B., Yu, J.: A formal description-based approach to extended many-valued context analysis. In: 4th Conf. Fuzzy Systems and Knowledge Discovery, vol. 1, pp. 545–549 (2007)
44. Lim, W.C., Lee, C.S.: Knowledge Discovery Through Composited Visualization, Navigation and Retrieval. In: Hoffmann, A., Motoda, H., Scheffer, T. (eds.) DS 2005. LNCS (LNAI), vol. 3735, pp. 377–379. Springer, Heidelberg (2005)
45. Maddouri, M.: Towards a machine learning approach based on incremental concept formation. Intelligent Data Analysis 8, 267–280 (2004)
46. Maddouri, M.: A Formal Concept Analysis Approach to Discover Association Rules from Data. In: Belohlavek, R., Snasel, V. (eds.) CLA, pp. 10-21 (2005)

47. Maddouri, M., Kaabi, F.: On Statistical Measures for Selecting Pertinent Formal Concepts to Discover Production Rules from Data. In: 6th Int. Conf. Data Mining Workshops (2006)
48. Meddouri, N., Maddouri, M.: Boosting Formal Concepts to Discover Classification Rules. In: Chien, B.-C., Hong, T.-P., Chen, S.-M., Ali, M. (eds.) IEA/AIE 2009. LNCS (LNAI), vol. 5579, pp. 501–510. Springer, Heidelberg (2009)
49. Mens, K., Tourwe, T.: Delving source code with formal concept analysis. Computer Languages. Systems & Stuctures 3, 183–197 (2005)
50. Molloy, I., Chen, H., Li, T., Wang, Q., Li, N., Bertino, E., Calo, S., Lobo, J.: Mining Roles with Semantic Meanings, Estes Park, Colorado, USA, June 11-13 (2008)
51. Myat, N.N., Hla, K.H.S.: A combined approach of formal concept analysis and text mining for concept based document clustering. In: Int. Conf. on Web Intelligence (2005)
52. Nehme, K., Valtchev, P., Rouane, M.H., Godin, R.: On Computing the Minimal Generator Family for Concept Lattices and Icebergs. In: Ganter, B., Godin, R. (eds.) ICFCA 2005. LNCS (LNAI), vol. 3403, pp. 192–207. Springer, Heidelberg (2005)
53. Nourine, L., Raynaud, O.: A fast algorithm for building lattices. In: Information Processing Letters, pp. 199–214 (1999)
54. Okubo, Y., Haraguchi, M.: Finding Conceptual Document Clusters with Improved Top-N Formal Concept Search. In: Int. Conf. on Web Intelligence (2006)
55. Pfaltz, J.C.: Representing numeric values in concept lattices. In: CLA (2007)
56. Poelmans, J., Elzinga, P., Viaene, S., Dedene, G.: A case of using formal concept analysis in combination with emergent self organizing maps for detecting domestic violence. In: Perner, P. (ed.) Advances in Data Mining. Applications and Theoretical Aspects. LNCS, vol. 5633, pp. 247–260. Springer, Heidelberg (2009)
57. Pogel, A., Ozonoff, D.: Contingency Structures and Concept Analysis. In: Medina, R., Obiedkov, S. (eds.) ICFCA 2008. LNCS (LNAI), vol. 4933, pp. 305–320. Springer, Heidelberg (2008)
58. Priss, U.: Formal Concept Analysis in Information Science. In: Blaise, C. (ed.) Annual Review of Information Science and Technology, ASIST, vol. 40 (2006)
59. Priss, U., Old, L.J.: Conceptual Exploration of Semantic Mirrors. In: Ganter, B., Godin, R. (eds.) ICFCA 2005. LNCS (LNAI), vol. 3403, pp. 21–32. Springer, Heidelberg (2005)
60. Qi, H., Liu, D.Y., Hu, C.Q., Lu, M., Zhao, L.: Searching for closed itemset with Formal Concept Analysis. In: 3rd Int. Conf. on Machine Learning and Cybernetics, Shanghai (2004)
61. Qu, L., Liu, D.: Extending Dynamic Aspect Mining Using Formal Concept Analysis. In: Proc. of 4th Conf. on Fuzzy Systems and Knowledge Discovery (2007)
62. Quan, T.T., Ngo, L.N., Hui, S.C.: An Effective Clustering-based Approach for Conceptual Association Rules Mining. In: Conf. Comp. & Comm. Techn., pp. 1–7 (2009)
63. Richards, D., Malik, U.: Multi level knowledge discovery from rule bases. Applied Artificial Intelligence 17, 181–205 (2003a)
64. Richards, D.: Addressing the Ontology Acquisition Bottleneck through Reverse Ontological Engineering. Knowledge and Information Systems 6, 402–427 (2004)
65. Richards, D.: Ad-Hoc and Personal Ontologies: A Prototyping Approach to Ontology Engineering. In: Hoffmann, A., Kang, B.-h., Richards, D., Tsumoto, S. (eds.) PKAW 2006. LNCS (LNAI), vol. 4303, pp. 13–24. Springer, Heidelberg (2006)
66. Richards, D., Malik, U.: Mining Propositional Knowledge Bases to Discover Multi-level Rules. In: Zaïane, O.R., Simoff, S.J., Djeraba, C. (eds.) MDM/KDD 2002 and KDMCD 2002. LNCS (LNAI), vol. 2797, pp. 199–216. Springer, Heidelberg (2003)

67. Rudolph, S.: Exploring Relational Structures Via FLE. In: Wolff, K.E., Pfeiffer, H.D., Delugach, H.S. (eds.) ICCS 2004. LNCS (LNAI), vol. 3127, pp. 196–212. Springer, Heidelberg (2004)
68. Rudolph, S.: Acquiring Generalized Domain-Range Restrictions. In: Medina, R., Obiedkov, S. (eds.) ICFCA 2008. LNCS (LNAI), vol. 4933, pp. 32–45. Springer, Heidelberg (2008)
69. Rudolph, S., Völker, J., Hitzler, P.: Supporting Lexical Ontology Learning by Relational Exploration. In: Priss, U., Polovina, S., Hill, R. (eds.) ICCS 2007. LNCS (LNAI), vol. 4604, pp. 488–491. Springer, Heidelberg (2007)
70. Sato, K., Okubo, Y., Haraguchi, M., Kunifuji, S.: Data Mining of Time-Series Medical Data by Formal Concept Analysis. In: Apolloni, B., Howlett, R.J., Jain, L. (eds.) KES 2007, Part II. LNCS (LNAI), vol. 4693, pp. 1214–1221. Springer, Heidelberg (2007)
71. Shao, M.W., Guo, Y.L.: Attribute reduction of large crisp-real concept lattices. In: Proc. of the 7th Int. Conf. on Machine Learning and Cybernetics, Kunming (2008)
72. Shi, B.S., Shen, X.J., Liu, Z.T.: A Knowledge discovery technique for heterogeneous data sources. In: Proc. of IEEE 2nd Int. Conf. on Machine Learning and Cybernetics, Xian (2003)
73. Sklenar, V., Zacpal, J., Sigmund, E.: Evaluation of IPAQ questionnaire by FCA. In: Belohlavek, R., Snasel, V. (eds.) CLA, pp. 60–69 (2005)
74. Stumme, G., Bestride, Y., Taouil, R., Lakhal, L.: Computing Iceberg Concept Lattices with TITANIC. Data and Knowledge Engineering 42(2), 189–222 (2002)
75. Stumme, G., Wille, R., Wille, U.: Conceptual knowledge discovery in databases using Formal Concept Analysis Methods. In: Żytkow, J.M. (ed.) PKDD 1998. LNCS, vol. 1510, pp. 450–458. Springer, Heidelberg (1998)
76. Tekaya, S.B., Yahia, S.B., Slimani, Y.: GenAll Algorithm: Decorating Galois lattice with minimal generators. In: Belohlavek, R., Snasel, V. (eds.) CLA, pp. 166–178 (2005)
77. Thomas, J., Cook, K.: Illuminating the path: research and development agenda for visual analytics. National Visualization and Analytics Ctr. (2005)
78. Tilley, T.: Tool support for FCA. In: Eklund, P. (ed.) ICFCA 2004. LNCS (LNAI), vol. 2961, pp. 104–111. Springer, Heidelberg (2004)
79. Tilley, T., Eklund, P.: Citation analysis using Formal Concept Analysis: A case study in software engineering. In: 18th Int. Conf. on Database and Expert Systems Applications (2007)
80. Tonella, P., Ceccato, M.: Aspect Mining through the Formal Concept Analysis of Execution Traces. In: Proc. of the 11th Working Conf. on Reverse Engineering (2004)
81. Valtchev, P., Missaoui, R., Godin, R.: Formal Concept Analysis for Knowledge Discovery and Data Mining: The New Challenges. In: Eklund, P. (ed.) ICFCA 2004. LNCS (LNAI), vol. 2961, pp. 352–371. Springer, Heidelberg (2004)
82. Valtchev, P., Missaoui, R., Godin, R.: A framework for incremental generation of closed itemsets. Discrete Applied Mathematics 156, 924–949 (2008)
83. Valverde-Albacete, F.J., Pelaez-Moreno, C.: Galois Connections Between Semimodules and Applications in Data Mining. In: Kuznetsov, S.O., Schmidt, S. (eds.) ICFCA 2007. LNCS (LNAI), vol. 4390, pp. 181–196. Springer, Heidelberg (2007)
84. Valverde-Albacete, F.J., Pelaez-Moreno, C.: Spectral Lattices of Rmax,+-Formal Contexts. In: Medina, R., Obiedkov, S. (eds.) ICFCA 2008. LNCS (LNAI), vol. 4933, pp. 124–139. Springer, Heidelberg (2008)
85. Valverde-Albacete, F.J., Pelaez-Moreno, C.: Towards a generalization of Formal concept analysis for data mining purposes. In: Missaoui, R., Schmidt, J. (eds.) ICFCA 2006. LNCS (LNAI), vol. 3874, pp. 161–176. Springer, Heidelberg (2006)

86. Volker, J., Rudolph, S.: Fostering Web Intelligence by Semi-automatic OWL Ontology Refinement. In: Int. Conf. on Web Intelligence and Intelligent Agent Technology (2008)
87. Volker, J., Rudolph, S.: Lexico-Logical Acquisition of OWL DL Axioms: An Integrated Approach to Ontology Refinement. In: Medina, R., Obiedkov, S. (eds.) ICFCA 2008. LNCS (LNAI), vol. 4933, pp. 62–77. Springer, Heidelberg (2008)
88. Wang, H., Zhang, W.X.: Generalized attribute reduction in consistent decision formal context. In: 7th Int. Conf. on Machine Learning and Cybernetics, Kunming, July 12-15 (2008)
89. Wermelinger, M., Yu, Y., Strohmaier, M.: Using Formal Concept Analysis to Construct and Visualize Hierarchies of Socio-Technical Relations. In: ICSE 2009, Vancouver (2009)
90. Wille, R.: Methods of conceptual knowledge processing. In: Missaoui, R., Schmidt, J. (eds.) ICFCA 2006. LNCS (LNAI), vol. 3874, pp. 1–29. Springer, Heidelberg (2006)
91. Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: Rival, I. (ed.) Ordered sets, pp. 445–470. Reidel, Dordrecht (1982)
92. Wollbold, J., Guthke, R., Ganter, B.: Constructing a Knowledge Base for Gene Regulatory Dynamics by Formal Concept Analysis Methods. In: Horimoto, K., Regensburger, G., Rosenkranz, M., Yoshida, H. (eds.) AB 2008. LNCS, vol. 5147, pp. 230–244. Springer, Heidelberg (2008)
93. Wu, W.Z., Leung, Y., Mi, J.S.: Granular Computing and Knowledqe Reduction in Formal Contexts. IEEE Transactions on Knowledge & Data Engineering 21(10) (October 2009)
94. Yahia, S.B., Xguifo, E.M.: Revisiting Generic Bases of Association Rules. In: Kambayashi, Y., Mohania, M., Wöß, W. (eds.) DaWaK 2004. LNCS, vol. 3181, pp. 58–67. Springer, Heidelberg (2004)
95. Yan, W., Baoxiang, C.: Fuzzy Many-Valued Context Analysis Based on Formal Description. In: 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (2007)
96. Yang, S.Z., Hou, X.W., Zhang, M.Q.: Approach on Aspect-Oriented Software Reverse Engineering at Requirements Level. In: Int. Conf. on Computer Science and Software Engineering. IEEE, Los Alamitos (2008)
97. Yang, Y., Du, Y., Sun, J., Hai, Y.: A Topic-Specific Web Crawler with Concept Similarity Context Graph Based on FCA. In: Huang, D.-S., Wunsch II, D.C., Levine, D.S., Jo, K.-H. (eds.) ICIC 2008. LNCS (LNAI), vol. 5227, pp. 840–847. Springer, Heidelberg (2008)
98. Zarate, L.E., Dias, S.M.: Qualitative behavior rules for the cold rolling process extracted from trained ANN via the FCANN method. Engineering Applications of Artificial Intelligence 22, 718–731 (2009)
99. Zhou, B., Hui, S.C., Chang, K.: A formal concept analysis approach for web usage mining. Intelligent Information Processing II 163, 437–441 (2005)
100. Zhou, B., Hui, S.C., Fong, A.C.M.: An Effective Approach for Periodic Web Personalization. In: Proc. of the IEEE/WIC/ACM Int. Conf. on Web Intelligence (2006)

Granular Reduction of Property-Oriented Concept Lattices

Ling Wei[1], Xiao-Hua Zhang[1], and Jian-Jun Qi[2]

[1] Department of Mathematics, Northwest University, Xi'an, 710069, PR China
wl@nwu.edu.cn, zhangxiaohua1980@163.com
[2] School of Computer Science & Technology, Xidian University,
Xi'an, 710071, PR China
qjj426@gmail.com

Abstract. Konwledge reduction on concept lattices is an important research topic in formal concept analysis, and there are some different meanings and methods. This paper mainly studies the granular reduction of property-oriented concept lattices which can preserve the property-oriented concepts, which is called granules in this paper. Firstly, the granules of a property-oriented concept lattice is presented, and it is proven that each extension of property-oriented concepts can be constructed by the granules. Then, the granular reduction of a property-oriented concept lattice is proposed, and the granular discernibility attribute matrix and the granular discernibility attribute function are respectively employed to determine granular consistent sets and calculate granular reducts. Finally, the relation between a granular consistent set and a lattice consistent set of a property-oriented concept lattice is discussed.

1 Introduction

The theory of concept lattices, also called formal concept analysis, was proposed by Wille in 1982 [3,11]. A concept lattice is a hierarchical structure defined by a binary relation between an object set and an attribute set, called a formal context, which reflects the relationship of specialization and generalization among concepts. As an efficient tool for data analysis, the theory of concept lattices has been extensively applied to various fields, such as machine learning, data mining and software engineering [1,5,6,7,9,18].

The attribute reduction theory of concept lattices is one important issue in knowledge representation and data mining. Attribute reduction is to find a subset of the attribute set to keep some characteristics for a certain formal context. Such idea is to delete some redundant attributes while not change its potential information. Recently, much attention has been paid to knowledge reduction in formal concept analysis [10,12,15,16]. For example, based on the discernibility matrix and the discernibility function, an approach to attribute reduction in concept lattices was presented in literature [15,16], which preserves all the concepts and their hierarchy in a formal context. In [10], based on meet-irreducible elements in concept lattices, a new approach to attribute reduction was developed,

M. Croitoru, S. Ferré, and D. Lukose (Eds.): ICCS 2010, LNAI 6208, pp. 154–164, 2010.

this approach is to find the minimal set of attributes, which can preserve meet-irreducible elements of the concept lattice. Granular computing and knowledge reduction in formal context was discussed in [12]. Compared to the studies on the attribute reduction theory of classical concept lattices, there has less investigation on the reduction theory of property-oriented concept lattice. Duntsch and Gediga presented the property-oriented concept lattice using a pair of approximation operators [2,3,13,14]. Yao introduced the object-oriented concept lattice and proved that object-oriented concept lattice is isomorphic to the property-oriented concept lattice for a same formal context [13,14]. Liu [8] discussed the reduction of object-oriented concept lattice and property-oriented concept lattice using the lattice-preserving reduction theory proposed in [15], and proposed judgment approaches of consistent sets. Wang [10] studied relations of attribute reduction between the object-oriented concept lattice and property-oriented concept lattice from viewpoint of irreducible elements, and obtained that reducts and attribute characteristics are identical in the two concept lattices.

Suggested by the reduction theory of classical concept lattices, we discuss the granular reduction of property-oriented concept lattices. Basic definitions of formal contexts and property-oriented concept lattices are recalled in Section 2. In Section 3, a granule is defined and the relation between the granules and extensions are investigated. Section 4 gives definitions of a granular consistent set and a granular reduct of a property-oriented concept lattice firstly, an approach to granular reduction of property-oriented concept lattice is then developed based on discernibility matrix, and the relation between granular consistent set and lattice consistent set of [8] is discussed. Finally, the paper is concluded by Section 5.

2 Preliminaries

To make this paper self-contained, we introduce the involved notions in the theory of concept lattices in this section [3,11].

Definition 1. *A **formal context** (G, M, I) consists of two sets G and M and a relation I between G and M. The elements of G are called the objects and the elements of M are called the attributes of the context. In order to express that an object g is in a relation I with an attribute m, we write gIm or $(g, m) \in I$ and read it as "the object g has the attribute m".*

In [11], Wille defined operators $*$ and $'$ for every$X \subseteq G, B \subseteq M$:

$$X^* := \{a \in M | \forall x \in X, xRa\},$$
$$B' := \{x \in G | \forall a \in B, xRa\}.$$

In [13], there are other denotations. $\forall x \in U, \forall a \in A$, xR is an attribute set that x possesses: $xR := \{a \in M | xRa\} = \{x\}^*$, Ra is an object set in which each object possesses a: $Ra := \{x \in G | xRa\} = \{a\}'$. For simplicity, a single-point set will be denoted by its symbol in this paper. For example, we will write x^* instead of $\{x\}^*$ and a' instead of $\{a\}'$ in the sequence.

In this paper, we assume that all the formal contexts are regular, that is, for every $x \in G, x^* \neq \emptyset, x^* \neq M$, and for every $a \in M, a' \neq \emptyset, a' \neq G$. And also, we assume that all the formal contexts are finite, that is, G and M are finite sets.

With respect to a formal context (G, M, I), a pair of dual approximation operators, $\diamondsuit : 2^G \to 2^M$ and $\square : 2^M \to 2^G$, are defined as follows [2,4,13,14]:

$X^{\diamondsuit} := \{a \in M | \exists x \in G(xRa \wedge x \in X)\} = \{a \in M | Ra \cap X \neq \emptyset\} = \bigcup_{x \in X} xR = XR$.

$B^{\square} := \{x \in G | \forall a \in M(xRa \Rightarrow a \in B)\} = \{x \in G | xR \subseteq B\}$.

It's easy to see that, $\forall x \in G$, $x^{\diamondsuit} = x^* = xR$.

The approximation operators have the following properties [13,14]: $\forall X, X_1, X_2 \subseteq G, \forall B, B_1, B_2 \subseteq M$,

(1). $X_1 \subseteq X_2 \Rightarrow X_1^{\diamondsuit} \subseteq X_2^{\diamondsuit}$; $B_1 \subseteq B_2 \Rightarrow B_1^{\square} \subseteq B_2^{\square}$.
(2). $X \subseteq X^{\diamondsuit\square}; B^{\square\diamondsuit} \subseteq B$.
(3). $X^{\diamondsuit\square\diamondsuit} = X^{\diamondsuit}; B^{\square\diamondsuit\square} = B^{\square}$.
(4). $(X_1 \cup X_2)^{\diamondsuit} = X_1^{\diamondsuit} \cup X_2^{\diamondsuit}; (B_1 \cap B_2)^{\square} = B_1^{\square} \cap B_2^{\square}$.

Definition 2. *[2,4] Suppose (G, M, I) is a formal context. A pair (X, B), $X \subseteq G, B \subseteq M$, is called a* ***property-oriented concept****, if $X = B^{\square}$ and $B = X^{\diamondsuit}$. The object set X and the attribute set B are called the* ***extension*** *and the* ***intension*** *of (X, B) respectively.*

The set of all property-oriented concepts of a formal context (G, M, I) is denoted by $L_P(G, M, I)$.

$\forall (X_1, B_1), (X_2, B_2) \in L_P(G, M, I)$, we say that (X_1, B_1) is a sub-concept of (X_2, B_2) and (X_2, B_2) is a super-concept of (X_1, B_1) if and only if $X_1 \subseteq X_2$, or equivalently, $B_1 \subseteq B_2$.

$L_P(G, M, I)$ is a complete lattice called property-oriented concept lattice, where, the infimum and supremum of the property-oriented concepts are given by

$(X_1, B_1) \wedge (X_2, B_2) = (X_1 \cap X_2, (X_1 \cap X_2)^{\diamondsuit}) = (X_1 \cap X_2, (B_1 \cap B_2)^{\square\diamondsuit})$,
$(X_1, B_1) \vee (X_2, B_2) = ((B_1 \cup B_2)^{\square}, B_1 \cup B_2) = ((X_1 \cup X_2)^{\diamondsuit\square}, B_1 \cup B_2)$.

3 Relation between the Granules and Extensions of Property-Oriented Concepts

In this section, we first define the granule of a property-oriented lattice induced by a formal context, and then study the relation between granules and extensions.

Suppose (G, M, I) is a formal context. For every $x \in G$, $(x^{\diamondsuit\square}, x^{\diamondsuit})$ is a property-oriented concept, which is called a granule of the property-oriented concept lattice $L_P(G, M, I)$. Let $E_G = \{x^{\diamondsuit\square} | x \in G\}$. It is easy to see that E_G forms a covering of the universe of discourse G.

Example 1. A formal context (G, M, I) is shown in Table 1, where $G = \{1, 2, 3, 4, 5, 6\}$ is an object set, $M = \{a, b, c, d, e\}$ is an attribute set.

To facilitate our discussion, any set is represented by its element string. For example, an object set $\{1, 2, 3\}$ is represented by 123, an attribute set $\{a, b, c\}$ is

Table 1. A formal context (G, M, I)

G	a	b	c	d	e
1	×		×	×	×
2	×		×		
3		×			×
4		×			×
5	×				
6	×	×			×

represented by *abc*. Then, the granules of the property-oriented concept lattice $L_P(G, M, I)$ can be calculated as follows: $(1^{\diamond\Box}, 1^{\diamond}) = (125, acde), (2^{\diamond\Box}, 2^{\diamond}) = (25, ac), (3^{\diamond\Box}, 3^{\diamond}) = (34, be) = (4^{\diamond\Box}, 4^{\diamond}), (5^{\diamond\Box}, 5^{\diamond}) = (5, a), (6^{\diamond\Box}, 6^{\diamond}) = (3456, abe)$. Thus, $E_G = \{\{1, 2, 5\}, \{2, 5\}, \{3, 4\}, \{5\}, \{3, 4, 5, 6\}\}$.

Lemma 1. *Let (G, M, I) be a formal context. Any element in E_G cannot be expressed by the join of the other ones.*

Proof. Let $G = \{x_1, ..., x_n\}$ and $E_G = \{x_1^{\diamond\Box}, \ldots, x_s^{\diamond\Box}\}, s \leq n$.

Suppose there exists one element in E_G can be denoted by join of other ones. Without loss of generality, we suppose $x_1^{\diamond\Box} = \bigcup_{\substack{l\in\tau \\ \tau\subseteq\{2,\cdots,s\}}} x_l^{\diamond\Box}$. Thus, $\forall l \in \tau, x_l^{\diamond\Box} \subseteq x_1^{\diamond\Box}$. Since x_1 must belong to one of $x_l^{\diamond\Box}, l \in \tau$, for example $x_1 \in x_k^{\diamond\Box}, k \in \tau$, then we have $x_1^{\diamond} \subseteq x_k^{\diamond}$, thus, $x_1^{\diamond\Box} \subseteq x_k^{\diamond\Box}$. Therefore, $x_1^{\diamond\Box} = x_k^{\diamond\Box}$, which contradicts the fact that elements in a set should not be the same. The proof is completed.

The lemma shows that every $(x^{\diamond\Box}, x^{\diamond})$ is a join-irreducible element of the property-oriented concept lattice.

Lemma 2. *Suppose (G, M, I) is a formal context, for every $(X, B) \in L_P(G, M, I)$, $X = \bigcup_{x\in X} x^{\diamond\Box}$ and $B = \bigcup_{x\in X} x^{\diamond}$ hold.*

Proof. For every $x \in X$, we have $x \in x^{\diamond\Box}$, then $X \subseteq \bigcup_{x\in X} x^{\diamond\Box}$. On the other hand, since $(X, B) \in L_P(G, M, I)$, we have $X = X^{\diamond\Box} = (\bigcup_{x\in X} x)^{\diamond\Box} = (\bigcup_{x\in X} x^{\diamond})^{\Box}$, $x^{\diamond} \subseteq \bigcup_{x\in X} x^{\diamond} \Rightarrow x^{\diamond\Box} \subseteq (\bigcup_{x\in X} x^{\diamond})^{\Box} \Rightarrow \bigcup_{x\in X} x^{\diamond\Box} \subseteq (\bigcup_{x\in X} x^{\diamond})^{\Box} = X$. Combining above two aspects, we have $X = \bigcup_{x\in X} x^{\diamond\Box}$.

For every $(X, B) \in L_P(G, M, I)$, we have $X = B^{\Box}$ and $B = X^{\diamond}$. According to property (3) $X^{\diamond\Box\diamond} = X^{\diamond}$, property (4) $(X_1 \cup X_2)^{\diamond} = X_1^{\diamond} \cup X_2^{\diamond}$, and $X = \bigcup_{x\in X} x^{\diamond\Box}$, we can obtain that $B = X^{\diamond} = (\bigcup_{x\in X} x^{\diamond\Box})^{\diamond} = \bigcup_{x\in X} x^{\diamond\Box\diamond} = \bigcup_{x\in X} x^{\diamond}$.

From Lemma 1 and Lemma 2, we know that any extension of a property-oriented concept can be constructed by the granules, and the granule can be taken as

the minimum unit in $L_P(U, A, R)$. Because every property-oriented concept can be determined by its extension, the approach to construct property-oriented concepts by extensions based on E_G can be researched.

Corollary 1. *Suppose (G, M, I) is a formal context, for every $(X, B) \in L_P(G, M, I)$, we have $(X, B) = \underset{x\in X}{\vee}(x^{\Diamond\Box}, x^{\Diamond})$.*

Proof. For every $(X, B) \in L_P(G, M, I)$, we have $X = B^{\Box}, B = X^{\Diamond}$. Since $\underset{x\in X}{\vee}(x^{\Diamond\Box}, x^{\Diamond}) = ((\bigcup_{x\in X} x^{\Diamond\Box})^{\Diamond\Box}, \bigcup_{x\in X} x^{\Diamond}) = ((\bigcup_{x\in X} x^{\Diamond\Box\Diamond})^{\Box}, \bigcup_{x\in X} x^{\Diamond}) = ((\bigcup_{x\in X} x^{\Diamond})^{\Box}, X^{\Diamond}) = (X^{\Diamond\Box}, X^{\Diamond}) = (X, B)$, we have $(X, B) = \underset{x\in X}{\vee}(x^{\Diamond\Box}, x^{\Diamond})$.

4 The Granular Reduction of Property-Oriented Concept Lattices

In this section, we first define the notions of a granular consistent set and a granular reduct, and propose a granular reduction method based on discernibility attribute matrix. Then, we discuss the attribute characteristics of granular core. Finally, relations between granular reduction and lattice reduction of a property-oriented concept lattice are analyzed.

4.1 The Granular Reduction of Property-Oriented Concept Lattices

Firstly, we introduce two operators $\Diamond B$ and $\Box B$, which are defined on an object set $X \subseteq G$ and an attribute set $B \subseteq M$, respectively.

$\Diamond B : 2^G \to 2^B$ and $\Box B : 2^B \to 2^G$ are a pair of dual operators similar with $\Diamond$ and $\Box$. $\forall X \subseteq G, C \subseteq B \subseteq M$:

$X^{\Diamond B} := \{a \in B | Ra \cap X \neq \emptyset\}$.

$C^{\Box B} := \{x \in G | \forall a \in B(xRa \Rightarrow a \in C)\} = \{x \in G | xR \cap B \subseteq C\}$.

The following properties give the relation between $\Diamond B$ and $\Diamond$, $\Box B$ and $\Box$, which can be proven by definitions.

Property 1. Suppose (G, M, I) is a formal context. Then, $\forall X \subseteq G, \forall x \in G, \forall C \subseteq M, \forall B \subseteq M, \forall a \in M$,

(1). $X^{\Diamond B} = X^{\Diamond} \cap B \subseteq X^{\Diamond}$. Especially, $x^{\Diamond B} \subseteq x^{\Diamond}$.

(2). $C^{\Box} \subseteq C^{\Box B}$. Especially, $a^{\Box} \subseteq a^{\Box B}$.

Definition 3. *Suppose (G, M, I) is a formal context. If there exists $B \subseteq M (B \neq \emptyset)$ such that $x^{\Diamond\Box} = x^{\Diamond B\Box B}$ for every $x \in G$, then B is called a granular consistent set of the property-oriented concept lattice $L_P(G, M, I)$; If B is a **granular consistent set** of $L_P(G, M, I)$ and there is no proper subset $C \subset B$ such that C is a granular consistent set of $L_P(G, M, I)$, then B is called a **granular reduct** of $L_P(G, M, I)$; The intersection of all granular reducts is called the **granular core** of $L_P(G, M, I)$.*

Example 2. In example 1, if $B = \{a, b, c, d\}$, then, we have $(1^{\Diamond B\Box B}, 1^{\Diamond B}) = (125, acd)$, $(2^{\Diamond B\Box B}, 2^{\Diamond B}) = (25, ac)$, $(3^{\Diamond B\Box B}, 3^{\Diamond B}) = (34, b) = (4^{\Diamond B\Box B}, 4^{\Diamond B})$, $(5^{\Diamond B\Box B}, 5^{\Diamond B}) = (5, a)$, $(6^{\Diamond B\Box B}, 6^{\Diamond B}) = (3456, ab)$. So, $E_B = E_G$. Namely, B is a granular consistent set of $L_P(G, M, I)$. Furthermore, since there is no proper subset $C \subset B$ such that C is a granular consistent set of $L_P(G, M, I)$, B is a granular reduct of $L_P(G, M, I)$.

Definition 4. *Suppose (G, M, I) is a formal context. $\forall x, y \in G$, we define the* ***granular discernibility attribute set*** *between property-oriented concepts $(x^{\Diamond\Box}, x^{\Diamond})$ and $(y^{\Diamond\Box}, y^{\Diamond})$ as $D(x, y) = x^{\Diamond} - y^{\Diamond}$. And Λ is a matrix over index sets G and M with $\Lambda_{(x,y)} := D(x, y)$, $x, y \in G$, which is called the* ***granular discernibility matrix*** *of the property-oriented concept lattice $L_P(G, M, I)$.*

Theorem 1. *Suppose (G, M, I) is a formal context. $\forall x, y, z \in G$, we have the following properties:*
(1). $D(x, x) = \emptyset$;
(2). $D(x, y)$ and $D(y, x)$ may be different;
(3). $D(x, y) \subseteq D(x, z) \cup D(z, y)$.

Proof. (1) and (2). They are obvious.

(3). Suppose $a \in D(x, y)$, that means $a \in x^{\Diamond}, a \notin y^{\Diamond}$. For any other $z \in U$, the relation between a and z is either $a \in z^{\Diamond}$ or $a \notin z^{\Diamond}$. If $a \in z^{\Diamond}$, that is, $a \in x^{\Diamond}, a \notin y^{\Diamond}$, $a \in z^{\Diamond}$, we have $a \in D(z, y)$; if $a \notin z^{\Diamond}$, that is, $a \in x^{\Diamond}, a \notin y^{\Diamond}$, $a \notin z^{\Diamond}$, we have $a \in D(x, z)$. Therefore, (3) is proven.

Theorem 2. *Suppose (G, M, I) is a formal context, and $B \subseteq M, B \neq \emptyset$. Then the following propositions are equivalent:*
(1). B is a granular consistent set of $L_P(G, M, I)$;
(2). For every $D(x, y) \neq \emptyset$, $B \cap D(x, y) \neq \emptyset$;
(3). For every $C \subseteq M$, if $C \cap B = \emptyset$, then $C \notin \Lambda$.

Proof. (1)⇒(2). If B is a granular consistent set of $L_P(G, M, I)$, then we have $y^{\Diamond B\Box B} = y^{\Diamond\Box}$ for all $y \in G$. Combining the fact $y^{\Diamond B\Box B} = \{x \in G | xR \cap B \subseteq y^{\Diamond B}\} = \{x \in G | x^{\Diamond B} \subseteq y^{\Diamond B}\}$ and $y^{\Diamond\Box} = \{x \in G | xR \subseteq y^{\Diamond}\} = \{x \in G | x^{\Diamond} \subseteq y^{\Diamond}\}$, we have that $x \in y^{\Diamond B\Box B} \Leftrightarrow x^{\Diamond B} \subseteq y^{\Diamond B}$; $x \in y^{\Diamond\Box} \Leftrightarrow x^{\Diamond} \subseteq y^{\Diamond}$. So, we can obtain that $x^{\Diamond B} \subseteq y^{\Diamond B} \Rightarrow x^{\Diamond} \subseteq y^{\Diamond}$. That means $x^{\Diamond} \nsubseteq y^{\Diamond} \Rightarrow x^{\Diamond B} \nsubseteq y^{\Diamond B}$. For every $D(x, y) \neq \emptyset$, we can find $a \in M$ such that $a \in x^{\Diamond}, a \notin y^{\Diamond}$, that means $x^{\Diamond} \nsubseteq y^{\Diamond}$, then $x^{\Diamond B} \nsubseteq y^{\Diamond B}$. It means, there exists an attribute $b \in x^{\Diamond B} = x^{\Diamond} \cap B$ and $b \notin y^{\Diamond B}$, therefore, $b \in B, b \in x^{\Diamond}, b \notin y^{\Diamond}$, that is, $b \in B, b \in D(x, y)$, therefore, $B \cap D(x, y) \neq \emptyset$.

(2)⇒(1). For every $D(x, y) \neq \emptyset$, $B \cap D(x, y) \neq \emptyset$, there exists $b \in B \cap D(x, y)$, that is, $b \in B, b \in x^{\Diamond}, b \notin y^{\Diamond}$, i.e., $b \in x^{\Diamond B}, b \notin y^{\Diamond B}$, thus, $x^{\Diamond B} \nsubseteq y^{\Diamond B}$.

Assume that B is not a granular consistent set of $L_P(G, M, I)$. According to Definition 3, there exists $y \in G$ such that $y^{\Diamond B\Box B} \neq y^{\Diamond\Box}$. That means, there exists an object $x \in G$ such that $x \in y^{\Diamond B\Box B}$, $x \notin y^{\Diamond\Box}$ or $x \in y^{\Diamond\Box}$, $x \notin y^{\Diamond B\Box B}$. If $x \in y^{\Diamond B\Box B}$, $x \notin y^{\Diamond\Box}$, then $x^{\Diamond B} \subseteq y^{\Diamond B}$, which contradicts the given condition. If $x \in y^{\Diamond\Box}$, $x \notin y^{\Diamond B\Box B}$, we have $x^{\Diamond} \subseteq y^{\Diamond}$ and $x^{\Diamond B} \nsubseteq y^{\Diamond B}$, which is obviously wrong. Therefore, B is a granular consistent set of $L_P(G, M, I)$.

(2)⇔(3). It is obvious.

Lemma 3. *For every $a \in M$, there must exist $x, y \in G$ such that $a \in D(x, y)$.*

Proof. Assume that there doesn't exist $x, y \in G$ such that $a \in D(x, y)$ for $a \in M$. That is, for every $x, y \in G, x \neq y$, we have $a \notin D(x, y)$. According to Definition 4, we can obtain $a \in y^{\diamondsuit}$ or $a \notin x^{\diamondsuit}$, which can be classified into 4 cases: (1) $a \in y^{\diamondsuit}, a \in x^{\diamondsuit}$; (2) $a \in y^{\diamondsuit}, a \notin x^{\diamondsuit}$; (3) $a \notin y^{\diamondsuit}, a \notin x^{\diamondsuit}$; (4) $a \notin y^{\diamondsuit}, a \in x^{\diamondsuit}$. It's easy to see (2) and (4) contradict with the assumption. We only consider the case (1) and case (3). Since $x, y \in G$ are arbitrary, (1) means each object in G has property a, i.e. $Ra = a' = G$. (3) means each object in G has no property a, i.e. $Ra = a' = \emptyset$. Both of them contradict the fact that the formal context (G, M, I) is regular. Therefore, the assumption is wrong, and the proposition is proven.

This Lemma shows that each attribute must appear in at least one element of the discernability matrix.

4.2 Attribute Characteristics of Granular Core

From the granular discernibility matrix of $L_P(G, M, I)$, we can discuss the attribute characteristics of granular core as follows.

Theorem 3. *Suppose (G, M, I) is a formal context, then the following propositions are equivalent:*
(1). a is a granular core attribute of $L_P(G, M, I)$;
(2). There exist $x, y \in G$ such that $D(x, y) = \{a\}$;
(3). $\forall x \in G$, $x^{\diamondsuit\square} \neq x^{\diamondsuit(M-\{a\})\square(M-\{a\})}$.

Proof. (1)⇒(2). For $x, y \in G$ satisfying $D(x, y) \neq \emptyset$, suppose $a \in D(x, y)$ and $|D(x, y)| \geq 2$. Let $B = \cup\{D(x, y) - \{a\} | x, y \in G\}$, then $B \cap D(x, y) \neq \emptyset$. From Theorem 2, we know that B is a granular consistent set of $L_P(G, M, I)$. There must exists $C \subseteq B$ such that C is a granular reduct of $L_P(G, M, I)$. Since $a \notin B \Rightarrow a \notin C$, a is not a core attribute, which contradicts the known condition.

(2)⇒(1). If there exists $x, y \in G$ such that $D(x, y) = \{a\}$, we have $a \in x^{\diamondsuit}$, $a \notin y^{\diamondsuit}$. If $x^{\diamondsuit} = \{a\}$, then $x^{\diamondsuit(M-\{a\})} = \emptyset$, $x^{\diamondsuit\square} \neq x^{\diamondsuit(M-\{a\})\square(M-\{a\})}$. It means that $M - \{a\}$ is not a consistent set, and a is a core attribute. If $|x^{\diamondsuit}| \geq 2$, i.e. there exists $b \neq a, b \in x^{\diamondsuit}$, we can obtain $b \in y^{\diamondsuit}$ since $D(x, y) = \{a\}$. So, $x^{\diamondsuit(M-\{a\})} \subseteq y^{\diamondsuit(M-\{a\})}$, we have $x^{\diamondsuit\square} \neq x^{\diamondsuit(M-\{a\})\square(M-\{a\})}$. Otherwise, for any $b \neq a, b \in x^{\diamondsuit}$ such that $b \notin y^{\diamondsuit}$, we have $D(x, y) = \{a, b\}$. This contradicts the given condition. Therefore, a is a granular core attribute of $L_P(G, M, I)$.

(3)⇔(1). It can be proven naturally from definition of core.

Example 3. For Example 1, the granular discernibility attribute sets are described in Table 2. Where, the first column and the first row are x and y in Definition 4, respectively. According to Definition 3 and Theorem 2, it can be verified that the property-oriented concept lattice has two granular reducts: $B_1 = \{a, b, c, d\}$, $B_2 = \{a, b, c, e\}$, and the granular core is $\{a, b, c\}$.

Table 2. Granular discernibility attribute matrix of (G, M, I)

	1	2	3	4	5	6
1	$\emptyset$	de	acd	acd	cde	cd
2	$\emptyset$	$\emptyset$	ac	ac	c	c
3	b	be	$\emptyset$	$\emptyset$	be	$\emptyset$
4	b	be	$\emptyset$	$\emptyset$	be	$\emptyset$
5	$\emptyset$	$\emptyset$	a	a	$\emptyset$	$\emptyset$
6	b	be	a	a	be	$\emptyset$

4.3 Granular Discernibility Attribute Function

Definition 5. *Suppose (G, M, I) is a formal context. A **granular discernibility attribute function** M^* for the property-oriented concept lattice $L_P(G, M, I)$ is a Boolean function of m Boolean variables $\bar{a}_1$, $\bar{a}_2$, ..., $\bar{a}_m$ corresponding to the attributes $a_1, a_2, ..., a_m$ respectively, and it is defined by*

$$M^* := \wedge\{\vee\Lambda_{(x,y)} : \Lambda_{(x,y)} \in \Lambda, \Lambda_{(x,y)} \neq \emptyset\}.$$

Where $\vee\Lambda_{(x,y)}$ is the disjunction of all variables $\bar{a}$ such that $a \in \Lambda_{(x,y)}$.

Theorem 4. *Suppose (G, M, I) is a formal context. Then an attribute subset $B \subseteq G$ is a granular reduct of $L_P(G, M, I)$, if and only if $\bigwedge\limits_{a_j \in B} \bar{a}_j$ is a prime implicant of the discernibility attribute function.*

Proof. Necessity. Assume that $B \subseteq G$ is a granular reduct of $L_P(G, M, I)$. By Theorem 2, we have $B \cap D(x, y) \neq \emptyset$ for all $D(x, y) \in \Lambda$ with $D(x, y) \neq \emptyset$. We claim that for every $b \in B$, there must exist $D(x, y) \in \Lambda$ with $D(x, y) \neq \emptyset$ such that $B \cap D(x, y) = \{b\}$. If $|(B \cap D(x, y)| \geq 2$ for every $D(x, y) \in \Lambda$ with $b \in D(x, y)$, let $B' = B - \{b\}$, then by Theorem 2, we can see that B' is a granular consistent set of $L_P(G, M, I)$, which contradicts that B is a granular reduct. It follows that $\wedge B$ is a prime implicant of the discernibility attribute function M^*.

Sufficiency. If $\wedge B$ is a prime implicant of the discernibility attribute function M^*, then $B \cap D(x, y) \neq \emptyset$ for all $D(x, y) \in \Lambda$ with $D(x, y) \neq \emptyset$. Moreover, for every $b \in B$, there exists $D(x, y) \in \Lambda$ such that $B \cap D(x, y) = \{b\}$. Consequently, $B - \{b\}$ is not a granular consistent set of $L_P(G, M, I)$. Thus we conclude that B is a granular reduct of $L_P(G, M, I)$.

From this theorem, we know that if result of M^* are all the prime implicants of the discernibility attribute function, then we obtain all granular reducts of the property-oriented concept lattice $L_P(G, M, I)$. For simplicity, we use a_j instead of $\bar{a}_j$ in the following example.

Example 4. For Example 3, we can obtain the granular discernibility attribute function from Table 2:

$$\begin{aligned} M^* &= a \wedge b \wedge c \wedge (d \vee e) \wedge (a \vee c \vee d) \wedge (a \vee c) \wedge (c \vee d) \wedge (c \vee d \vee e) \wedge (b \vee e) \\ &= (a \wedge b \wedge c \wedge d) \vee (a \wedge b \wedge c \wedge e) \end{aligned}$$

4.4 Relation between a Granular Consistent Set and Lattice Consistent Set of a Property-Oriented Concept Lattice

Liu studied the lattice reduction of a property-oriented concept lattice [8], such reduction is to find a minimal subset of attribute set to preserve the lattice structure. This subsection discusses the relation between the lattice reduction and our granular reduction, and shows that a lattice consistent set must be a granular consistent set.

Definition 6. *[8] Suppose (G, M, I) is a formal context, $B \subseteq M$, $I_B = I \cap (G \times B)$. If the set of extensions of $L_P(G, B, I_B)$ and $L_P(G, M, I)$ are the same, that is, $L_P(G, B, I_B) =_U L_P(G, M, I)$, then B is a consistent set of $L_P(G, M, I)$. Moreover, $\forall d \in B$, if $L_P(G, B - \{b\}, I_{B-\{b\}}) \neq_U L_P(G, M, I)$, we say B is a reduct of $L_P(G, M, I)$.*

Here, to avoid confusion, we call these two notions lattice consistent set and lattice reduct of $L_P(G, M, I)$.

Theorem 5. *Suppose (G, M, I) is a formal context. $\forall B \subseteq M, B \neq \emptyset$, if B is a lattice consistent set of $L_P(G, M, I)$, then B is a granular consistent set of $L_P(G, M, I)$. Namely, $L_P(G, B, I_B) =_U L_P(G, M, I) \Rightarrow \forall x \in G$, $x^{\Diamond\Box} = x^{\Diamond B\Box B}$.*

Proof. Assume that $L_P(G, B, I_B) =_U L_P(G, M, I)$. For every $x^{\Diamond B\Box B} \in E_B$, we have $(x^{\Diamond B\Box B}, x^{\Diamond B}) \in L_P(G, B, I_B)$. There exists $(X', B') \in L_P(G, M, I)$ such that $X' = x^{\Diamond B\Box B}$ from $L_P(G, B, I_B) =_U L_P(G, M, I)$. By Lemma 2, we deduce that $x^{\Diamond B\Box B} = X' = \bigcup_{i=1}^{k} x_i^{\Diamond\Box}$ and $x_i^{\Diamond\Box} \in E_G, i = 1, 2, ...k$. Since $(x_i^{\Diamond\Box}, x_i^{\Diamond}) \in L_P(G, M, I), i = 1, 2, ...k$, and $L_P(G, B, I_B) =_U L_P(G, M, I)$, we deduce that there exists $(X_i, B_i) \in L_P(G, B, I_B)$ such that $X_i = x_i^{\Diamond\Box}, i = 1, 2, ...k$. By lemma 2, we obtain that $x_i^{\Diamond\Box} = X_i = \bigcup_{j=1}^{s} x_{ij}^{\Diamond B\Box B}, i = 1, 2, ...k; j = 1, 2, ...s$. Therefore, $x^{\Diamond B\Box B} = \bigcup_{i=1}^{k} x_i^{\Diamond\Box} = \bigcup_{i=1}^{k} (\bigcup_{j=1}^{s} x_{ij}^{\Diamond B\Box B})$. By Lemma 1, we have $k = s = 1$, and further obtain $x^{\Diamond B\Box B} = x_i^{\Diamond\Box} \in E_G$. So, $E_B \subseteq E_G$. Similar to the proposed approach, $E_G \subseteq E_B$ is proven. Therefore, B is a granular consistent set of $L_P(G, M, I)$.

Theorem 5 shows that if an attribute subset is not a granular consistent set of a property-oriented concept lattice, then it must be not a lattice consistent set of the property-oriented concept lattice. The following example shows their relations.

Example 5. For Example 1, Fig.1 shows the property-oriented concept lattice $L_P(G, M, I)$.

By the lattice consistence judgment theorems of $L_P(G, M, I)$ in [8], we obtain the only lattice reduct of $L_P(G, M, I)$: $B_1 = \{a, b, c, d\}$, the corresponding lattice is shown in Fig.2. While, we know from Example 3 and Example 4 that there are two granular reducts: $B_1 = \{a, b, c, d\}$ and $B_2 = \{a, b, c, e\}$. The corresponding

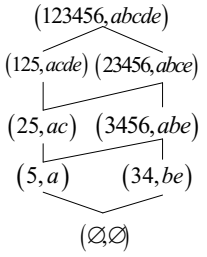


Fig. 1. $L_P(G, M, I)$

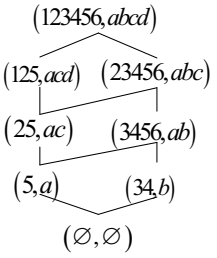


Fig. 2. $L_P(G, B_1, I_{B_1})$

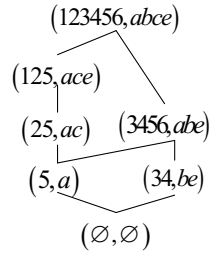


Fig. 3. $L_P(G, B_2, I_{B_2})$

lattice $L_P(G, B_1, I_{B_1})$ and $L_P(G, B_2, I_{B_2})$ are shown as Fig.2 and Fig.3 respectively. Both of them preserve all the granules of the property-oriented concept lattice $L_P(G, M, I)$, but the structure of $L_P(G, B_2, I_{B_2})$ is different from the original lattice, that is, B_2 is not a lattice consistent set.

5 Conclusion

Granular computing and knowledge reduction are two basic issues in knowledge representation and data mining. In this paper, we mainly discuss the granular reduction of property-oriented concept lattice. A granular reduction of a property-oriented concept lattice is a minimal attribute subset preserving the granules of the raw property-oriented concept lattice with whole attributes. The granular discernibility matrix and the granular discernibility attribute function have also been employed to calculate all the granular reducts.

Similar to this paper, we can study the granular of object-oriented concept lattices, which remove the objects that are not essential to preserve the granules. For further research, heuristic methods need to be developed to speed up the process to find granular reducts. We will also develop the proposed approach to deal with more complicated formal contexts, such as fuzzy formal contexts.

Acknowledgements

The authors gratefully acknowledge the reviewers for their constructive suggestions, and the support of the Natural Science Foundation of China (No.60703117), and NWU Graduate Innovation and Creativity Funds (No.09YZZ57).

References

1. Chen, Y.H., Yao, Y.Y.: A multiview approach for intelligent data analysis based on data operators. Information Sciences 178(1), 1–20 (2008)
2. Duntsch, I., Gediga, G.: Approximation operators in qualitative data analysis. In: de Swart, H., Orlowska, E., Schmidt, G., Roubens, M. (eds.) Theory and Application of Relation of Structures as Knowledge Instruments, pp. 216–233. Springer, Heidelberg (2003)

3. Ganter, B., Wille, R.: Formal Concept Analysis, Mathematical Foundations. Springer, Berlin (1999)
4. Gediga, G., Duntsch, I.: Modal-style operations in qualitative data analysis. In: Proceedings of the 2002 IEEE International Conference on Data Mining, pp. 155–162 (2002)
5. Gély, A., Medina, R., Nourine, L.: Representing lattices using many-valued relations. Information Sciences 179, 2729–2739 (2009)
6. Godin, R., Missaoui, R., Alaoui, H.: Incremental Concept Formation Algorithms Based on Galois (Concept) Lattices. Computational Intelligence 11(2), 246–267 (1995)
7. Kent, R.E., Bowman, C.M.: Digital libraries, conceptual knowledge systems and the nebula interface. Technical Report, University of Arkansas (1995)
8. Liu, M.Q., Wei, L., Zhao, W.: The reduction theory of object oriented concept lattices and property oriented concept lattices. In: Wen, P., et al. (eds.) RSKT 2009. LNCS, vol. 5589, pp. 587–593. Springer, Heidelberg (2009)
9. Sutton, A., Maletic, J.I.: Recovering UML class models from C++: A detailed explanation. Information and Software Technology 49(3), 212–229 (2007)
10. Wang, X., Ma, J.M.: A novel approach to attribute reduction in concept lattices. In: Wang, G.-Y., Peters, J.F., Skowron, A., Yao, Y. (eds.) RSKT 2006. LNCS (LNAI), vol. 4062, pp. 522–529. Springer, Heidelberg (2006)
11. Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: Rival, I. (ed.) Ordered Sets, pp. 445–470. Reidel, Dordrecht (1982)
12. Wu, W.Z., Leung, Y., Mi, J.S.: Granular computing and knowledge reduction in formal contexts. IEEE Transaction on Knowledge and Data Engineering 21(10), 1464–1474 (2009)
13. Yao, Y.Y.: Concept lattices in rough set theory. In: Dick, S., Kurgan, L., Pedrycz, W., Reformat, M. (eds.) Proceedings of 2004 Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS 2004), June 27-30, pp. 796–801 (2004), IEEE Catalog Number: 04TH8736
14. Yao, Y.Y.: A comparative study of formal concept analysis and rough set theory in data analysis. In: Tsumoto, S., Słowiński, R., Komorowski, J., Grzymała-Busse, J.W. (eds.) RSCTC 2004. LNCS (LNAI), vol. 3066, pp. 59–68. Springer, Heidelberg (2004)
15. Zhang, W.X., Wei, L., Qi, J.J.: Attribute reduction in concept lattice based on discernibility matrix. In: Ślęzak, D., Yao, J., Peters, J.F., Ziarko, W.P., Hu, X. (eds.) RSFDGrC 2005. LNCS (LNAI), vol. 3642, pp. 157–165. Springer, Heidelberg (2005)
16. Zhang, W.X., Wei, L., Qi, J.J.: Attribute reduction theory and approach to concept lattice. Science in China Series F: Information Sciences 48(6), 713–726 (2005)
17. Zhang, W.X., Wang, X.: Relations of attribute reduction between object and property oriented lattices. Knowledge-Based Systems 21, 398–403 (2008)
18. Zhu, W.: Relationship between generalized rough sets based on binary relation and covering. Information Sciences 179, 210–225 (2009)

Temporal Relational Semantic Systems

Karl Erich Wolff

Mathematics and Science Faculty
Darmstadt University of Applied Sciences
Holzhofallee 38, D-64295 Darmstadt, Germany
karl.erich.wolff@t-online.de
http://www.fbmn.fh-darmstadt.de/home/wolff

Abstract. In this paper we introduce the notion of a Temporal Relational Semantic System (TRSS) as a relational extension of a Temporal Conceptual Semantic System (TCSS) as defined by the author in 2007. TCSSs allow to represent states of distributed objects in concept lattices of specified views, the TRSSs enable the user to work in a conceptually simple way with arbitrary relational structures changing temporally. The flexible tools of conceptual scaling can now be applied also to relational structures. That yields for example a clear theoretical notion of the state of a temporal relation at some time granule. The main notions in TRSSs are visualized in some diagrams constructed from an example of a TRSS.

1 Introduction

In this paper Temporal Relational Semantic Systems are introduced with the purpose to develop practical, successful methods for the representation, evaluation and visualization of temporal relational structures in applications. These investigations are based on the Theory of Conceptual Graphs as developed by J. Sowa [So84, So00] and the mathematization of concepts in Formal Concept Analysis [Wi82, GW99a]. For the purpose of representing relational knowledge R. Wille [Wi97] has introduced power context families and their concept graphs. It has been shown by the author [Wo09a, Wo09b] that power context families can be described by Relational Semantic Systems which have the advantage that usual conceptual scaling can be applied to relational data. Based on former experiences with Temporal Conceptual Semantic Systems [Wo07a, Wo07b] the notion of a Temporal Relational Semantic System will be introduced in the following.

One of the central notions in this paper is the notion of a *relation* as a subset of a cartesian product of sets; we like to distinguish the notion of a relation clearly from related notions. To be short, we do that by some examples. To represent *relational statements* like 'In May 2008 Bob lived in London' and 'In spring 2009 Alice lived in Berlin' we may introduce the *relational formula* 'In the time period t of the year y person p lived in the location q' where t, y, p, q are variables for elements in domains D_1, D_2, D_3, D_4 . The *relation* of

M. Croitoru, S. Ferré, and D. Lukose (Eds.): ICCS 2010, LNAI 6208, pp. 165–180, 2010.

this relational formula with respect to the given set of relational statements is the set

$\{(May, 2008, Bob, London), (spring, 2009, Alice, Berlin)\} \subseteq \mathrm{D}_1 \times \mathrm{D}_2 \times \mathrm{D}_3 \times \mathrm{D}_4$.

In the following we shall represent any relational statement in a row of a data table in such a way that we protocol a short version of the relation formula, for example 'in...lived in.', and then we write down the corresponding tuple of the relational statement.

For an effective presentation of the main ideas around TRSSs we start with an example.

1.1 A Tabular Representation of a Temporal Relational Semantic System

In the rows of Table 1 we represent some temporal and non-temporal statements: 'from 2008 to 2009 Bob lived in England', 'in May 2008 Bob lived in London', 'in spring 2009 Alice lived in Berlin', 'in spring 2009 Bob met Alice in Paris', and 'Paris is the native town of Alice'.

Table 1. A data table for temporal relational information

statement	r^*	$TIME_1$	$TIME_2$	$PERSON_1$	$PERSON_2$	LOCATION
1	from.to..lived in.	2008	2009	BOB		ENGLAND
2	in...lived in.	May	2008	BOB		LONDON
3	in...lived in.	spring	2009	ALICE		BERLIN
4	in...met. in.	spring	2009	BOB	ALICE	PARIS
5	.is the native town of.			ALICE		PARIS

The main information for reading the statements in the intended sequence is represented in Table 2, called the *position table*. For example, the first position of *.is the native town of.* is the attribute LOCATION, its second position is the attribute $PERSON_1$.

Table 2. The position table

r^*	$TIME_1$	$TIME_2$	$PERSON_1$	$PERSON_2$	LOCATION
from.to..lived in.	1	2	3		4
in...lived in.	1	2	3		4
in...met. in.	1	2	3	4	5
.is the native town of.			2		1

Table 1 and Table 2 show a tabular representation of the main information in a temporal relational data system. This example is indeed a simple relational data system in the sense of the following definition.

2 Relational Data Systems and Relational Semantic Systems

We start with the definition of a simple relational data system.

Definition 1. *"simple Relational Data System"*
Let G, M, W be sets, $\lambda : G \times M \to W$. Then $\mathfrak{R} := (G, M, W, \lambda, r^*, \alpha, \pi)$ is called a simple Relational Data System (sRDS) if

- *$r^* \in M$, r^* is called the relational attribute,*
- *$W_{r^*} := \{\lambda(g, r^*) \mid g \in G\}$, W_{r^*} is called the set of relations of $\mathfrak{R}$,*
- *$\alpha : W_{r^*} \to \mathbb{N} := \{1, 2, \ldots\}$, α is called the arity,*
- *π is a mapping which maps each $c \in W_{r^*}$ to an injection $\pi_c : [1, \alpha(c)] \to M \setminus \{r^*\}$.*

To relate Definition 1 with the introductory example in section 1 we mention that there the set $\mathrm{G} = \{1, 2, 3, 4, 5\}$, the set of relational statements; M is the set $\{r^*$, TIME_1, TIME_2, PERSON_1, PERSON_2, LOCATION$\}$; W is the set of values in Table 1, as for example " in...lived in. ", 2009, BOB, and values indicating the missing values in the empty cells of Table 1; λ is the mapping which maps each row-column-pair in Table 1 to the value in its cell.; r^* is the special many-valued attribute from M, whose values are the used relations; the set of these relations is denoted by W_{r^*}; the mapping α assigns to each relation in W_{r^*} its arity. The position mapping π describes the positions for each relation from W_{r^*}; for example the relation $c :=$ .is the native town of. has arity 2, and its first position $\pi_c(1) = \mathrm{LOCATION}$ and its second position $\pi_c(2) = \mathrm{PERSON}_1$.

In a sRDS $\mathfrak{R}$ the elements of the set W_{r^*} are called the relations of $\mathfrak{R}$; they are not introduced as subsets of a cartesian product, but each element $c \in W_{r^*}$ determines in $\mathfrak{R}$ a unique relation, namely the set $\{\vec{c}(g) \mid g \in G\}$ where $\vec{c}(g) := (\lambda(g, \pi_c(i)))_{1 \le i \le \alpha(c)}$ is called the tuple of c at g.

For practical applications the notion of a simple relational data system will be sufficient for nearly all cases. Difficulties occur only when we try to represent an arbitrary power context family $\vec{\mathbb{K}} := (\mathbb{K}_0, \mathbb{K}_1, \mathbb{K}_2, \ldots)$ by a simple relational data system: if a context $\mathbb{K}_k$ has objects with empty intent or attributes with empty extent, then one needs a more complicated construction, namely that of a *Relational Data System* introduced by the author in [Wo09a].

Definition 2. *"Relational Data System"*
Let G, M, W be sets, $\lambda : G \times M \to W$. Then $\mathfrak{R} := (G, M, W, \lambda, r^*, A, \alpha, \beta, \pi)$ is called a Relational Data System (RDS) if

- *$r^* \in M$*
- *$A \subseteq W_{r^*} := \{\lambda(g, r^*) \mid g \in G\}$*
- *$\alpha : W_{r^*} \to \mathbb{N} := \{1, 2, \ldots\}$*
- *$\beta : W_{r^*} \to \mathfrak{P}(M \setminus \{r^*\}) := \{X \mid X \subseteq M \setminus \{r^*\}\}$*
- *π is a mapping which maps each $c \in W_{r^*}$ with $\beta(c) \neq \emptyset$ to a bijection $\pi_c : [1, \alpha(c)] \to \beta(c)$.*

For more details the reader is referred to [Wo09a, Wo09b].

The main purpose for the introduction of relational data systems is to combine the theory of relational structures with the possibilities of conceptual scaling. There are two different ways to do that. We could introduce conceptual scales on the complete many-valued context (G, M, W, λ) of a RDS. Instead, we wish to follow the traditional philosophical logic with its doctrines of concepts, judgments, and conclusions. Therefore, we consider the values of a data table as concepts and represent them as formal concepts of suitably chosen scales. That is done in the following definition of a relational semantic system.

Definition 3. *"Relational Semantic System"*
Let $\mathfrak{R} := (G, M, W, \lambda, r^*, A, \alpha, \beta, \pi)$ be a Relational Data System and for each $m \in M$ let $\mathbb{S}_m := (G_m, N_m, I_m)$ be a formal context and $\underline{\mathfrak{B}}(\mathbb{S}_m)$ its concept lattice. If $\lambda : G \times M \to W$ satisfies $\lambda(g, m) \in \mathfrak{B}(\mathbb{S}_m)$ for all $g \in G$ and all $m \in M$, then the pair $(\mathfrak{R}, (\mathbb{S}_m)_{m \in M})$ is called a Relational Semantic System (RSS).

It was shown by the author in [Wo09a, Wo09b] that RDS and their concept graphs can represent any power context family and any concept graph of a power context family [Wi97, Wi00, Wi04]. Relational semantic systems clearly have the same advantages and furthermore, they offer a simple way for conceptual scaling in relational structures. To extend relational semantic systems to temporal RSSs we first recall the basic ideas around temporal conceptual semantic systems.

3 Temporal Conceptual Semantic Systems

The purpose of introducing temporal relational semantic systems is to support all kinds of applications dealing with relational structures changing temporally. Our way to reach that goal is to combine the theory of temporal conceptual semantic systems as introduced by the author in [Wo06, Wo07a, Wo07b] with RSS and therefore with the relational theory around conceptual graphs of Sowa [So84, So00] and concept graphs of Wille [Wi97, Wi00, Wi04], Prediger [Pr98] and Prediger and Wille [PW99].

3.1 Conceptual Semantic Systems

The basic idea behind the following definition of a conceptual semantic system is to interprete the values of a data table as concepts and to represent them mathematically as formal concepts of given conceptual scales. First, we recall the definition of a conceptual semantic system and its semantically derived context [Wo05b, Wo06].

Definition 4. ***"Conceptual Semantic System"***
Let M be a set, and for each $m \in M$ let $\mathbb{S}_m := (G_m, N_m, I_m)$ be a formal context and $\underline{\mathfrak{B}}(\mathbb{S}_m)$ its concept lattice. Let G be a set and

$$\lambda : G \times M \to \bigcup_{m \in M} \mathfrak{B}(\mathbb{S}_m)$$

be a mapping such that $\lambda(g, m) \in \mathfrak{B}(\mathbb{S}_m)$.

Then the quadruple

$$\mathfrak{K} := (G,\ M,\ (\underline{\mathfrak{B}}(\mathbb{S}_m))_{m\in M}, \lambda)$$

is called a ***Conceptual Semantic System (CSS) with semantic scales*** $(\mathbb{S}_m)_{m\in M}$. *The elements of* M *are called* ***many-valued attributes****; the elements of* G *are called* ***instances****. We write* $m(g) := \lambda(g, m)$ *and* $m(G) := \{m(g) \mid g \in G\}$.

Remark 1: If $(\mathfrak{R}, (\mathbb{S}_m)_{m\in M})$ is a RSS, then $(G, M, (\underline{\mathfrak{B}}(\mathbb{S}_m))_{m\in M}, \lambda)$ is a CSS.

For any CSS we can introduce its semantically derived context which represents the many-valued context of its CSS without any loss of information.

Definition 5. ***"Semantically derived context"***
Let $\mathfrak{K} := (G,\ M,\ (\underline{\mathfrak{B}}(\mathbb{S}_m))_{m\in M}, \lambda)$ *be a CSS with semantic scales*

$$\mathbb{S}_m = (G_m, N_m, I_m)\ (m \in M),$$

*and let int(***c***) denote the intent of a concept* $\mathbf{c}$. *Then the formal context*

$$\mathbb{K} := (G, N, J)\ \textit{where}\ N := \{(m, n) \mid m \in M, n \in N_m\}\ \textit{and}$$
$$gJ(m, n) :\Longleftrightarrow n \in int(m(g))$$

is called the ***semantically derived context of*** $\mathfrak{K}$.

The definition of the incidence relation J of the semantically derived context $\mathbb{K} := (G, N, J)$ shows that any concept $m(g)$ where $m \in M$ and $g \in G$ can be reconstructed from the scale $\mathbb{S}_m$ via its intent which is stored in $\mathbb{K}$.

3.2 Views, Selections, and Traces

Now we recall the definitions of *views*, *selections*, and traces from [Wo07b].

Definition 6. *Let* $\mathbb{K} := (G, N, J)$ *be a formal context. Then any subset* $Q \subseteq N$ *is called a* ***view*** *of* $\mathbb{K}$. *The subcontext* $\mathbb{K}_Q := (G, Q, J \cap (G \times Q))$ *is called the* $Q-$**part** ***of*** $\mathbb{K}$.

In the following the concept lattice of the $Q - part$ of a view Q of the semantically derived context of a CSS will be used as a 'landscape' into which further information is embedded. To describe this embedding of information we use the following notion of a *selection*.

Definition 7. ***"Instance selection"***
Let $\mathfrak{K} := (G,\ M,\ (\underline{\mathfrak{B}}(\mathbb{S}_m))_{m\in M}, \lambda)$ *be a CSS,* $\mathcal{T}(M) := \{(\mathbf{c}_m)_{m\in M^*} \mid \mathbf{c}_m \in \mathfrak{B}(\mathbb{S}_m), \emptyset \neq M^* \subseteq M\}$, *called the set of tuples of semantic concepts over* M. *For a subset* $\mathcal{T} \subseteq \mathcal{T}(M)$ *any mapping*

$$\sigma : \mathcal{T} \to \mathfrak{P}(G) := \{X \mid X \subseteq G\}$$

is called ***an instance selection on*** $\mathcal{T}$. *For* $\mathbf{c} \in \mathcal{T}$ *the set* $\sigma(\mathbf{c}) \subseteq G$ *is called* ***the instance selection of*** $\mathbf{c}$ *or* ***the selection of*** $\mathbf{c}$.

The most important instance selection is the database selection

$$\sigma_{db}((\mathbf{c}_m)_{m\in M^*}) := \{g \in G \mid \forall_{m\in M^*}\ m(g) = \mathbf{c}_m\}.$$

In [Wo07a, Wo07b] the usage of other instance selections is shown.

Definition 8. ***"σ-Q-trace of a tuple"***
Let $\mathfrak{K} := (G, M, (\underline{\mathfrak{B}}(\mathbb{S}_m))_{m\in M}, \lambda)$ be a CSS with semantically derived context $\mathbb{K} := (G, N, J)$, σ an instance selection on $\mathcal{T} \subseteq \mathcal{T}(M)$, $\boldsymbol{c} \in \mathcal{T}$, and $Q \subseteq N$ a view with object concept mapping γ_Q. Then the set

$$\gamma_Q(\sigma(\boldsymbol{c}))$$

is called ***the σ-Q-trace of the tuple $\boldsymbol{c}$.***

As shown in [Wo07b] the σ-Q-traces of tuples can be used very effectively to visualize important subsets in the concept lattice of the Q-part of the derived context. That is conceptually the same technique as to represent towns, mountains, and rivers in a geographical map.

3.3 Temporal Conceptual Semantic Systems

From [Wo07a, Wo07b] we recall the definition of a Temporal Conceptual Semantic System (TCSS). There are four main ideas which are mathematically described in the definition of a TCSS:

1. A TCSS should be a CSS having a specified many-valued *time attribute* T whose scale contains all temporal concepts which are needed for the given purpose.
2. It also should have a specified set $\mathcal{O}$ of *temporal objects* which are used to describe moving objects (like cars) as opposed to static objects (like houses), clearly with respect to the given purpose.
3. Each temporal object $\mathfrak{o} \in \mathcal{O}$ is associated with a binary relation $\mathcal{R}_\mathfrak{o}$ of *base transitions* where each base transition is a pair of formal concepts from the time scale $\mathbb{S}_T$. Each base transition of the temporal object $\mathfrak{o}$ describes one step of $\mathfrak{o}$ in time; other temporal objects may do other steps in time.
4. Each temporal object $\mathfrak{o}$ of a TCSS is represented as a tuple of semantic concepts over M:

$\mathfrak{o} := (\mathbf{c}_m)_{m\in M^*} \in \mathcal{T}(\mathrm{M})$.

Definition 9. ***"Temporal Conceptual Semantic System"***
Let $\mathfrak{K} := (G, M, (\underline{\mathfrak{B}}(\mathbb{S}_m))_{m\in M}, \lambda)$ be a CSS, $T \in M$, $\mathcal{O} \subseteq \mathcal{T}(M)$, and for each $\mathfrak{o} \in \mathcal{O}$ let $\mathcal{R}_\mathfrak{o}$ be a binary relation on $\mathfrak{B}(\mathbb{S}_T)$. Then the quadruple

$$(\mathfrak{K}, T, \mathcal{O}, (\mathcal{R}_\mathfrak{o})_{\mathfrak{o}\in\mathcal{O}})$$

is called ***a Temporal Conceptual Semantic System (TCSS) with time attribute*** *T, the set $\mathcal{O}$ of* ***temporal objects****, and for each temporal object $\mathfrak{o} \in \mathcal{O}$ its* ***time relation*** *$\mathcal{R}_\mathfrak{o}$. The elements of $\mathcal{R}_\mathfrak{o}$ are called* ***base transitions of*** *$\mathfrak{o}$. The elements of $\mathfrak{B}(\mathbb{S}_T)$ are called* ***time granules****.*

Clearly, we could also specify a a *set of time attributes* as a subset of the set M of many-valued attributes, but that can be easily reduced to a single time attribute by taking the tuples of the set of time attributes as values of a single time attribute. In practice, such a set of time attributes is useful, for example for using different columns in the data table for days and months. Hence, for temporal relational semantic systems we will take a set of time attributes instead of just a single time attribute. That will have some consequences, mainly to replace the time granules (as formal concepts of the time scale $\mathbb{S}_T$) by *tuples* of formal concepts from the time scales, hence by elements of $\mathcal{T}(\mathrm{T}) := \{(\mathbf{c}_m)_{m\in M^*} \mid \mathbf{c}_m \in \mathfrak{B}(\mathbb{S}_m), \emptyset \neq M^* \subseteq T\}$.

4 Temporal Relational Semantic Systems

In the following we introduce Temporal Relational Semantic Systems (TRSS) based on the definition of a Temporal Conceptual Semantic System (TCSS). There are two main changes to be made: first, we have to replace the CSS by a RSS; second, we replace the single time attribute by a set T of time attributes.

The first change can be done easily since any RSS yields a CSS as mentioned in Remark 1 in section 3.1; therefore, it is possible to extend the theory of TCSSs to the theory of TRSSs. The second change enables us to represent statements with many time concepts; for example, dates are often given in the form 'ddmmyy' for days, months, and years; they will be represented as tuples of formal concepts from the semantic scales of the set T of time attributes.

Definition 10. ***"Temporal Relational Semantic System"***
Let $(\mathfrak{R}, (\mathbb{S}_m)_{m\in M})$ be a Relational Semantic System and r^* its relational attribute. Let $\emptyset \neq T \subseteq M \setminus \{r^*\}$, $\mathcal{O} \subseteq \mathcal{T}(M \setminus T)$, and for each $\mathfrak{o} \in \mathcal{O}$ let $\mathcal{R}_\mathfrak{o}$ be a binary relation on the set $\mathcal{T}(T)$; the elements of $\mathcal{T}(T)$ are called time granules, the elements of $\mathcal{R}_\mathfrak{o}$ are called base transitions of $\mathfrak{o}$. Then the quintuple

$$(\mathfrak{R}, (\mathbb{S}_m)_{m\in M}, T, \mathcal{O}, (\mathcal{R}_\mathfrak{o})_{\mathfrak{o}\in\mathcal{O}})$$

is called a ***Temporal Relational Semantic System*** *(TRSS) with the set T of* ***time attributes****, the set $\mathcal{O}$ of* ***temporal objects****, and for each temporal object $\mathfrak{o} \in \mathcal{O}$ its* ***time relation*** *$\mathcal{R}_\mathfrak{o}$.*

The time granules in $\mathcal{T}(\mathrm{T})$ are in the case that $|T| = 1$ (the 1-tuples of) the concepts of the time scale.

In the following we introduce the notion of a state of a temporal object $\mathfrak{o} \in \mathcal{O}$ at a time granule $\mathfrak{t} \in \mathcal{T}(\mathrm{T})$.

4.1 State of a Temporal Object at a Time Granule

Definition 11. ***"State of a temporal object at a time granule"***
Let $(\mathfrak{R}, (\mathbb{S}_m)_{m\in M}, T, \mathcal{O}, (\mathcal{R}_\mathfrak{o})_{\mathfrak{o}\in\mathcal{O}})$ be a TRSS and $\mathfrak{K} = (G, M, (\underline{\mathfrak{B}}(\mathbb{S}_m))_{m\in M}, \lambda)$ its CSS. Let Q be a view of the semantically derived context $\mathbb{K} = (G, N, J)$

of $\mathfrak{K}$. *For each temporal object* $\mathfrak{o} = (\boldsymbol{c}_m)_{m\in M_\mathfrak{o}} \in \mathcal{O}$ *and for each time granule* $\boldsymbol{t} = (\boldsymbol{c}_n)_{n\in M_\mathfrak{t}} \in \mathcal{T}(T)$ *the tuple* $(\mathfrak{o}, \boldsymbol{t})$ *is called an* ***actual object****. Let* σ *be an instance selection on a set* $\mathcal{T}$ *such that* $(\mathfrak{o}, \boldsymbol{t}) \in \mathcal{T}$.

The σ***-**Q***-state of the temporal object*** $\mathfrak{o}$ ***at time granule*** $\boldsymbol{t}$ *is defined as*

$$\gamma_Q(\sigma(\mathfrak{o}, \boldsymbol{t}))$$

which is the σ*-*Q*-trace of the actual object* $(\mathfrak{o}, \boldsymbol{t})$.

The actual object $(\mathfrak{o}, \mathbf{t})$ is understood here as the tuple $(\mathbf{c}_m)_{m\in M_\mathfrak{o}\cup M_\mathfrak{t}}$ which is a tuple over M since $M_\mathfrak{o} \subseteq M \setminus T$ and $M_\mathfrak{t} \subseteq T$.

A typical example of an actual object is the pair $(High, Monday)$ which represents a high-pressure-zone at Monday; here 'High' denotes a formal context of a pressure scale and 'Monday' a formal concept of a time scale. For details see [Wo07b].

4.2 Life Space and Life Track of a Temporal Object

Definition 12. ***"Life space and life track of a temporal object"***
Let $(\mathfrak{R}, (\mathbb{S}_m)_{m\in M}, T, \mathcal{O}, (\mathcal{R}_\mathfrak{o})_{\mathfrak{o}\in\mathcal{O}})$ *be a TRSS and* $\mathfrak{K} = (G, M, (\underline{\mathfrak{B}}(\mathbb{S}_m))_{m\in M}, \lambda)$ *its CSS. Let* Q *be a view of the semantically derived context* $\mathbb{K} = (G, N, J)$ *of* $\mathfrak{K}$. *We call the set* $T(\mathfrak{K}) := \{(\boldsymbol{c}_m)_{m\in T} \mid \exists_{g\in G}\forall_{m\in T}\lambda(g, m) = \boldsymbol{c}_m\}$ *the set of* T*-tuples of* $\mathfrak{K}$. *For each temporal object* $\mathfrak{o} = (\boldsymbol{c}_m)_{m\in M_\mathfrak{o}} \in \mathcal{O}$ *let* σ *be an instance selection on a set* $\mathcal{T}$ *containing* $\{(\mathfrak{o}, \boldsymbol{t}) \mid \boldsymbol{t} \in T(\mathfrak{K})\}$. *Then we call*

$$\mathcal{S}_{\sigma Q}(\mathfrak{o}) := \bigcup_{\boldsymbol{t}\in T(\mathfrak{K})} \gamma_Q(\sigma(\mathfrak{o}, \boldsymbol{t}))$$

the σ*-*Q***-life space of the temporal object*** $\mathfrak{o}$. *The set*

$$\mathcal{L}_{\sigma Q}(\mathfrak{o}) := \{((\mathfrak{o}, \boldsymbol{t}), \gamma_Q(\sigma(\mathfrak{o}, \boldsymbol{t}))) \mid \boldsymbol{t} \in T(\mathfrak{K})\}$$

is called ***the labelled*** σ***-**Q***-life space of*** $\mathfrak{o}$.

If $|\gamma_Q(\sigma(\mathfrak{o}, \boldsymbol{t}))| \leq 1$ *for each* $\boldsymbol{t} \in T(\mathfrak{K})$, *then we call* $\mathcal{L}_{\sigma Q}(\mathfrak{o})$ ***the*** σ***-**Q***-life track of*** $\mathfrak{o}$.

4.3 Transition of a Temporal Object

Definition 13. ***"*σ*-*Q*-transition of a temporal object"***
Let $(\mathfrak{R}, (\mathbb{S}_m)_{m\in M}, T, \mathcal{O}, (\mathcal{R}_\mathfrak{o})_{\mathfrak{o}\in\mathcal{O}})$ *be a TRSS and* $\mathfrak{K} = (G, M, (\underline{\mathfrak{B}}(\mathbb{S}_m))_{m\in M}, \lambda)$ *its CSS. Let* Q *be a view of the semantically derived context* $\mathbb{K} = (G, N, J)$ *of* $\mathfrak{K}$. *For each temporal object* $\mathfrak{o} = (\boldsymbol{c}_m)_{m\in M_\mathfrak{o}} \in \mathcal{O}$ *and for each base transition* $(\boldsymbol{s}, \boldsymbol{t}) \in \mathcal{R}_\mathfrak{o}$ *let* σ *be an instance selection on a set* $\mathcal{T}$ *containing the set* $\{(\mathfrak{o}, \boldsymbol{s}), (\mathfrak{o}, \boldsymbol{t})\}$. *Then we call the pair*

$$(((\mathfrak{o}, \boldsymbol{s}), (\mathfrak{o}, \boldsymbol{t})), (\gamma_Q(\sigma(\mathfrak{o}, \boldsymbol{s})), \gamma_Q(\sigma(\mathfrak{o}, \boldsymbol{t}))))$$

the σ***-**Q***-transition of*** $\mathfrak{o}$ *induced by the base transition* $(\boldsymbol{s}, \boldsymbol{t})$ *leading from the* ***initial place*** $((\mathfrak{o}, \boldsymbol{s}), \gamma_Q(\sigma(\mathfrak{o}, \boldsymbol{s})))$ *to the* ***final place*** $((\mathfrak{o}, \boldsymbol{t}), \gamma_Q(\sigma(\mathfrak{o}, \boldsymbol{t})))$.

4.4 The Life Space Digraph

On the labelled life space of an object $\mathfrak{o}$ we can easily introduce a digraph by "transporting" the time relation $\mathcal{R}_\mathfrak{o}$ into this space. That generalizes the life track digraph for CTSOTs [Wo05a] and for TCSSs [Wo07b] .

Definition 14. ***"Life space digraph of a temporal object"*** *Let $(\mathfrak{R}, (\mathbb{S}_m)_{m\in M}, T, \mathcal{O}, (\mathcal{R}_\mathfrak{o})_{\mathfrak{o}\in\mathcal{O}})$ be a TRSS and $\mathfrak{K} = (G, M, (\underline{\mathfrak{B}}(\mathbb{S}_m))_{m\in M}, \lambda)$ its CSS. Let Q be a view of the semantically derived context $\mathbb{K} = (G, N, J)$ of $\mathfrak{K}$. For each temporal object $\mathfrak{o} = (\boldsymbol{c}_m)_{m\in M_\mathfrak{o}} \in \mathcal{O}$ let σ be an instance selection on a set $\mathcal{T}$ covering the sets $\{(\mathfrak{o}, \boldsymbol{s}), (\mathfrak{o}, \boldsymbol{t})\}$ where $\boldsymbol{s}\,\mathcal{R}_\mathfrak{o}\,\boldsymbol{t}$. Then* ***the life space digraph of*** *$\mathfrak{o}$ is defined as the directed graph $(\mathcal{L}_{\sigma Q}(\mathfrak{o}), \hat{\mathcal{R}}_\mathfrak{o})$ where*

$((\mathfrak{o}, \boldsymbol{s}), \gamma_Q(\sigma(\mathfrak{o}, \boldsymbol{s})))\ \hat{\mathcal{R}}_\mathfrak{o}\ ((\mathfrak{o}, \boldsymbol{t}), \gamma_Q(\sigma(\mathfrak{o}, \boldsymbol{t}))) :\Leftrightarrow \boldsymbol{s}\ \mathcal{R}_\mathfrak{o}\ \boldsymbol{t}.$

5 Conceptual Investigation of the Introductory Example

5.1 Complete Description of the Introductory Example

Now, we give a complete description of the introductory example as a TRSS, roughly described by Table 1 and Table 2.

We introduce the TRSS $\mathfrak{R}_1 := (\mathfrak{R}, (\mathbb{S}_m)_{m\in M}, T, \mathcal{O}, (\mathcal{R}_\mathfrak{o})_{\mathfrak{o}\in\mathcal{O}})$ with $\mathfrak{K} = (\mathrm{G}, \mathrm{M}, (\underline{\mathfrak{B}}(\mathbb{S}_m))_{m\in M}, \lambda)$ as its CSS. As introduced in section 2 let $\mathrm{G} := \{1, 2, 3, 4, 5\}$; $\mathrm{M} := \{r^*, \mathrm{TIME}_1, \mathrm{TIME}_2, \mathrm{PERSON}_1, \mathrm{PERSON}_2, \mathrm{LOCATION}\}$. For each $m \in M$ we now introduce the scale $\mathbb{S}_m$. For the relational attribute r^* we define the scale $\mathbb{S}_{r^*}$ by the formal context in Table 3:

Table 3. The scale $\mathbb{S}_{r^*}$

r^*	r1	r2	r3	r4	r0
from.to..lived in.	×				×
in...lived in.		×			×
in...met. in.			×		
.is the native town of.				×	

The attribute concept of $r0$ can be used for a 'simultaneous search for the relations $r1$:= from.to..lived. and $r2$:= in...lived in.'. That will be shown later.

We define the scale $\mathbb{S}_{TIME_1}$ by the formal context in Table 4:

Table 4. The scale $\mathbb{S}_{TIME_1}$

	2008	spring	May
2008	×		
spring		×	
May		×	×
/			

In this scale we express that 'May' has the attribute 'spring'.

We define the scale $\mathbb{S}_{TIME_2}$ by the formal context in Table 5:

Table 5. The scale $\mathbb{S}_{TIME_2}$

	2008	2009	2008-2009
2008	×		×
2009		×	×
/			

We define the scale $\mathbb{S}_{PERSON_1}$ by the formal context in Table 6:

Table 6. The scale $\mathbb{S}_{PERSON_1}$

	ALICE	BOB
ALICE	×	
BOB		×

We define the scale $\mathbb{S}_{PERSON_2}$ by the formal context in Table 7:

Table 7. The scale $\mathbb{S}_{PERSON_2}$

	ALICE
ALICE	×
/	

We define the scale $\mathbb{S}_{LOCATION}$ by the formal context in Table 8:

Table 8. The scale $\mathbb{S}_{LOCATION}$

	England	London	Berlin	Paris	continental
ENGLAND	×				
LONDON	×	×			
BERLIN			×		×
PARIS				×	×

After having introduced all scales we introduce the mapping λ using Table 1 where for each column m each value of m is interpreted as the corresponding object concept in the scale $\mathbb{S}_m$. For example, the value '2008' is interpreted as the object concept of the formal object 2008 in the scale $\mathbb{S}_{TIME_1}$. We mention, that the same value '2008' in the $TIME_2$-column is interpreted as the object concept of '2008' in the scale $\mathbb{S}_{TIME_2}$. One can avoid that by introducing a common scale for TIME_1 and TIME_2, but we are free to choose what we like to do. Our choice here leads to a smaller number of attributes in the derived context.

The simple RDS of the TRSS $\mathfrak{R}_1$ has been explained in section 2. The set T of time attributes of $\mathfrak{R}_1$ is defined as $T := \{TIME_1, TIME_2\}$.

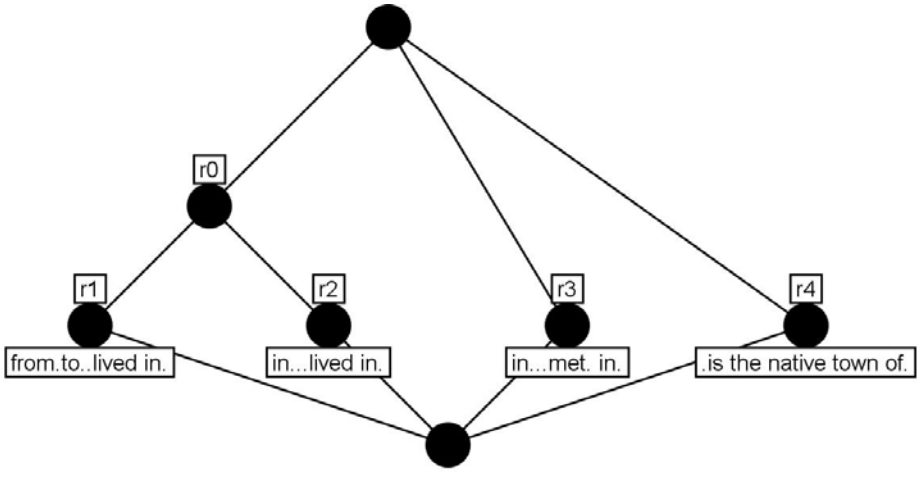


Fig. 1. The concept lattice of the scale $\mathbb{S}_{r*}$

We choose as our set $\mathcal{O}$ of temporal objects of $\mathfrak{R}_1$ the set $\mathcal{O} := \{\mathcal{BOB}\}$ and BOB's set of base transitions as $\mathcal{R}_{\mathcal{BOB}} := \{(\text{May, 2008}), (\text{spring, 2009})\}$. Now, we have finished the complete description of the TRSS $\mathfrak{R}_1$. In this paper we shall not discuss applications of life tracks and transitions, since that has been done by the author in previous papers [Wo07a, Wo07b].

5.2 Visualizing the Derived Context of the TRSS $\mathfrak{R}_1$

The following Proposition 1 yields the main reading rule for line diagrams of the derived context of a CSS.

Proposition 1. *Let* $\mathfrak{K} := (G,\ M,\ (\underline{\mathfrak{B}}(\mathbb{S}_m))_{m\in M}, \lambda)$ *be a CSS with semantic scales*
$\mathbb{S}_m = (G_m, N_m, I_m)$ $(m \in M)$, $\mathbb{K} := (G, N, J)$ *its derived context, and* γ *the object concept mapping of* $\mathbb{K}$.

Then for any $g \in G$ *and* $m \in M$

$$int(m(g)) = \{n \in N_m \mid (m, n) \in g^{\mathbb{K}}\}.$$

Proof. By definition of the derived context of a CSS
n ∈ int$(m(g))$ ⇔ g J (m,n) ⇔ (m,n) ∈ $g^{\mathbb{K}}$.

Proposition 1 yields the following

Reading rule for line diagrams of the derived context of a CSS:
For any instance g and any many-valued attribute m the elements n of the intent of the formal concept m(g) of the scale $\mathbb{S}_m$ can be seen in a line diagram of the concept lattice of the derived context $\mathbb{K}$ by collecting all attributes $n \in N_m$ such that the attribute concept of (m,n) is a superconcept of the object concept of g.

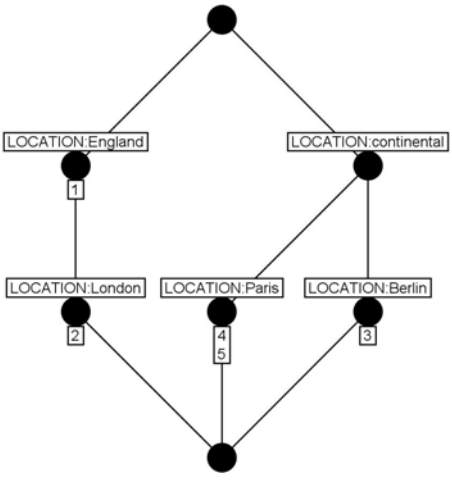


Fig. 2. The concept lattice of the LOCATION-part of the derived context

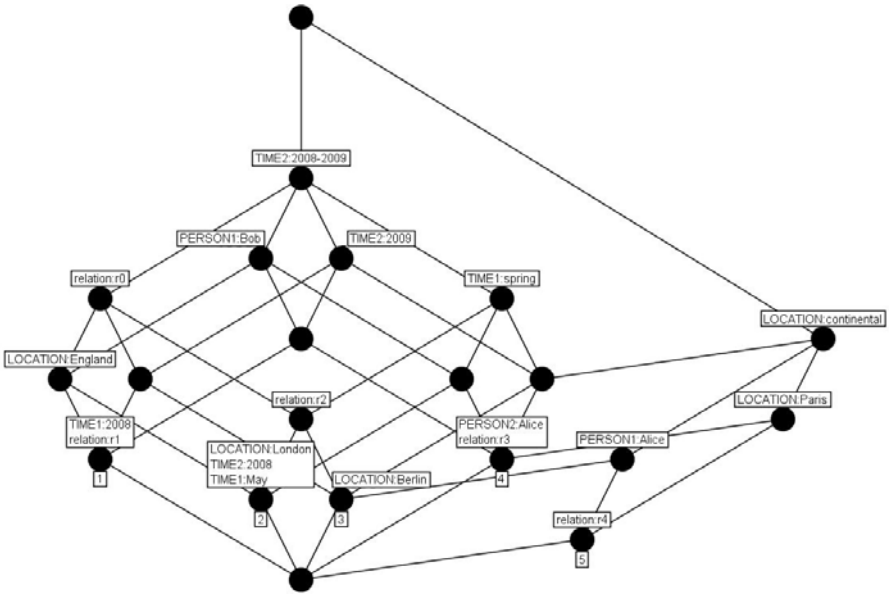


Fig. 3. The concept lattice of the derived context of the TRSS $\mathfrak{R}_1$

This reading rule holds also for $m-$parts (m $\in$ M) of the derived context which are defined as $Q-$parts where $Q := \{(\text{m,n}) \mid n \in N_m\}$.

The concept lattice of the LOCATION-part of the derived context $\mathbb{K}$ of the TRSS $\mathfrak{R}_1$ is shown in Fig. 2. It tells that statement 1 has in the LOCATION column a concept which has as intent just the set {England}, hence $\lambda(1, LOCATION) = ENGLAND$.

Fig. 3 shows the concept lattice of the derived context of the TRSS $\mathfrak{R}_1$. Here, the attributes of the r^*-part are denoted by (relation: n) instead of (r^*, n).

By the reading rule, Fig. 3 shows all the information of Table 1: when we use the position table we can read from Fig. 3 each relational statement of Table 1. For example: We can read the relational statement 1 from Fig. 3 by starting with the object concept of statement 1 (in the left of Fig. 3) and searching for those attributes of the derived context of $\mathfrak{R}_1$ which belong to the intent of the object concept of 1. Hence we have to look upwards from the circle of the object concept of statement 1 in Fig. 3 to find the appropriate attributes. In this case, we find 'relation: r1' and 'relation: r0'; since $\{r_0,r_1\}$ is the intent of the formal object 'from.to..lived in.' in the scale $\mathbb{S}_{r^*}$ we conclude from Proposition 1 that $\lambda(1,r^*)$ = 'from.to..lived in.'. Using the positions of this relation, we continue looking for the TIME_1 scale attribute, which is '2008'; for TIME_2 we find two scale attributes: '2009'and '2008-2009' which form the intent of the formal object '2009' in the scale for TIME_2; then we find PERSON_1: Bob, and LOCATION: England. That yields statement 1.

It is obvious that the concept lattice of the derived context of a TRSS is, even in small examples, not a good structure for successful visualizations. In the next section we use suitable views of the derived context of the TRSS and embed traces of tuples of concepts into the concept lattices of these views.

5.3 Visualizing Traces of Tuples in Suitable Views

The main technique for the construction of a good visualization, for example a map of a country, is the choice of a suitable view, into which further information is embedded, for example some information about lakes, forests, and mountains. This technique can be used also for the more general knowledge representation in TRSSs. First of all we choose a view Q of the derived context of the TRSS, then we choose a tuple $\mathbf{c}$ of concepts and a selection σ and construct the trace $\gamma_Q(\sigma(\mathbf{c}))$. Often we wish to visualize in the concept lattice of the $Q-$part many traces. That yields *relational trace diagrams*, introduced by the author in [Wo09b].

5.4 Visualizing States of Relations at a Time Granule

A very important special case of traces are the states. According to the definition of states we can introduce the state of a relation, for example the relation 'in...lived in.', at a time granule, for example (μ(spring), μ(2008-2009)) where μ(spring) denotes the attribute concept of 'spring' in the scale of TIME_1, and μ(2008-2009) denotes the attribute concept of '2008-2009' in the scale of TIME_2. To visualize that state we use the program TOSCANAJ [BH05] in the following way.

First of all we construct for the given many-valued context in Table 1 the scales for all the many-valued attributes (using the program ELBA or CERNATO), then we choose in TOSCANAJ the following sequence of scales for r^*, TIME_1, TIME_2, LOCATION, PERSON_1. Then we choose in the scale for r^* the attribute concept of r2, in the scale for TIME_1 the attribute concept of 'spring', in the scale

for TIME_2 the attribute concept of '2008-2009'. Then we draw the nested line diagram for the scales of LOCATION and of PERSON_1. The resulting diagram is shown in Fig. 4. The set of objects concepts in this diagram consists of the object concepts of the statements 2 and 3. The set of these two object concepts is exactly the state of the relation 'in...lived in.' at the time granule (μ(spring), μ(2008-2009)) in the view of the attributes of LOCATION and PERSON_1. This state of the relation tells the answer to the question: Which persons lived in the spring of the years 2008-2009 in which locations? The answer can be seen from Fig. 4: In spring 2008-2009 person Bob lived in London and person Alice lived in Berlin, and that is stated in the relational statements 2 and 3.

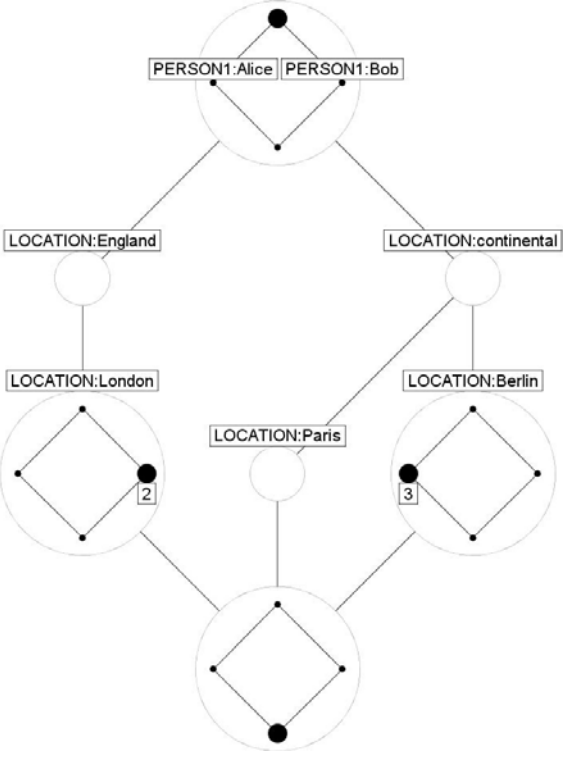


Fig. 4. The state of the relation 'in...lived in.' at the time granule '(spring, 2008-2009)' in the view of the attributes of LOCATION and PERSON_1 in the corresponding nested line diagram

6 Conclusion and Future Work

The introduction of TRSSs gives a theoretical and practical framework for temporal relational structures. It combines the theory of conceptual graphs by John Sowa, the theory of concept graphs by Rudolf Wille and the theory of temporal

conceptual structures as developed by the author. The theory of TRSSs is a starting point for many developments. At the moment there is a small temporal part in the program SIENA which can be used to draw life tracks of objects. For the investigation of distributed temporal objects like a High on a weather map it would be very useful for practical applications to extend the program TOSCANAJ by a strong temporal relational tool.

References

[BH05] Becker, P., Correia, J.H.: The ToscanaJ Suite for Implementing Conceptual Information Systems. In: Ganter, B., Stumme, G., Wille, R. (eds.) Formal Concept Analysis. LNCS (LNAI), vol. 3626, pp. 324–348. Springer, Heidelberg (2005)

[De91] Devlin, K.: Logic and Information. Cambridge University Press, Cambridge (1991)

[EGSW00] Eklund, P., Groh, B., Stumme, G., Wille, R.: A contextual-logic extension of TOSCANA. In: Ganter, B., Mineau, G.W. (eds.) ICCS 2000. LNCS (LNAI), vol. 1867, pp. 453–467. Springer, Heidelberg (2000)

[GW99a] Ganter, B., Wille, R.: Formal Concept Analysis: mathematical foundations. Springer, Heidelberg (1999); German version: Springer, Heidelberg (1996)

[He02] Hereth, J.: Relational Scaling and Databases. In: Priss, U., Corbett, D.R., Angelova, G. (eds.) ICCS 2002. LNCS (LNAI), vol. 2393, pp. 62–76. Springer, Heidelberg (2002)

[Pr98] Prediger, S.: Kontextuelle Urteilslogik mit Begriffsgraphen. In: Ein Beitrag zur Restrukturierung der mathematischen Logik. Dissertation, TU Darmstadt 1998. Shaker, Aachen (1998)

[PW99] Prediger, S., Wille, R.: The lattice of concept graphs of a relationally scaled context. In: Tepfenhart, W.M., Cyre, W. (eds.) ICCS 1999. LNCS (LNAI), vol. 1640, pp. 401–414. Springer, Heidelberg (1999)

[So84] Sowa, J.F.: Conceptual structures: information processing in mind and machine. Addison-Wesley, Reading (1984)

[So00] Sowa, J.F.: Knowledge representation: logical, philosophical, and computational foundations. Brooks Cole Publ. Comp., Pacific Grove (2000)

[Wi82] Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: Rival, I. (ed.) Ordered Sets, pp. 445–470. Reidel, Dordrecht (1982)

[Wi97] Wille, R.: Conceptual Graphs and Formal Concept Analysis. In: Delugach, H.S., Keeler, M.A., Searle, L., Lukose, D., Sowa, J.F. (eds.) ICCS 1997. LNCS (LNAI), vol. 1257, pp. 290–303. Springer, Heidelberg (1997)

[Wi00] Wille, R.: Contextual Logic summary. In: Stumme, G. (ed.) Working with conceptual structures: Contributions to ICCS 2000, pp. 265–276. Shaker-Verlag, Aachen (2000)

[Wi04] Wille, R.: Implicational Concept Graphs. In: Wolff, K.E., Pfeiffer, H.D., Delugach, H.S. (eds.) ICCS 2004. LNCS (LNAI), vol. 3127, pp. 52–61. Springer, Heidelberg (2004)

[Wo01] Wolff, K.E.: Temporal Concept Analysis. In: Mephu Nguifo, E., et al. (eds.) ICCS 2001 International Workshop on Concept Lattices-Based Theory, Methods and Tools for Knowledge Discovery in Databases, pp. 91–107. Stanford University, Palo Alto (2001)

[Wo02a] Wolff, K.E.: Transitions in Conceptual Time Systems. In: Dubois, D.M. (ed.) International Journal of Computing Anticipatory Systems, CHAOS 2002, vol. 11, pp. 398–412 (2002)

[Wo04] Wolff, K.E.: 'Particles' and 'Waves' as Understood by Temporal Concept Analysis. In: Wolff, K.E., Pfeiffer, H.D., Delugach, H.S. (eds.) ICCS 2004. LNCS (LNAI), vol. 3127, pp. 126–141. Springer, Heidelberg (2004)

[Wo05a] Wolff, K.E.: States, Transitions, and Life Tracks in Temporal Concept Analysis. In: Ganter, B., Stumme, G., Wille, R. (eds.) Formal Concept Analysis. LNCS (LNAI), vol. 3626, pp. 127–148. Springer, Heidelberg (2005)

[Wo05b] Wolff, K.E.: States of Distributed Objects in Conceptual Semantic Systems. In: Dau, F., Mugnier, M.-L., Stumme, G. (eds.) ICCS 2005. LNCS (LNAI), vol. 3596, pp. 250–266. Springer, Heidelberg (2005)

[Wo06] Wolff, K.E.: Conceptual Semantic Systems - Theory and Applications. In: Goncharov, S.S., Downey, R., Ono, H. (eds.) Proceedings of the 9th Asian Logic Conference on Mathematical Logic in Asia, pp. 288–301. World Scientific, New Jersey (2006)

[Wo07a] Wolff, K.E.: Basic Notions in Temporal Conceptual Semantic Systems. In: Gély, A., Kuznetsov, S.O., Nourine, L., Schmidt, S.E. (eds.) Contributions to ICFCA 2007, 5th International Conference on Formal Concept Analysis, Clermont-Ferrand, France, pp. 97–120 (2007)

[Wo07b] Wolff, K.E.: Applications of Temporal Conceptual Semantic Systems. In: Zagoruiko, N.G., Palchunov, D.E. (eds.) Knowledge - Ontology - Theory, Russian Academy of Sciences, vol. 2, pp. 3–16. Sobolev Institute for Mathematics, Novosibirsk (2007)

[Wo09a] Wolff, K.E.: Relational Semantic Systems, Power Context Families, and Concept Graphs. In: Wolff, K.E., Rudolph, S., Ferré, S. (eds.) Contributions to ICFCA 2009. International Conference on Formal Concept Analysis 2009, pp. 63–78. Verlag Allgemeine Wissenschaft, Darmstadt (2009)

[Wo09b] Wolff, K.E.: Relational Scaling in Relational Semantic Systems. In: Rudolph, S., Dau, F., Kuznetsov, S.O. (eds.) Conceptual Structures: Leveraging Semantic Technologies. LNCS (LNAI), vol. 5662, pp. 307–320. Springer, Heidelberg (2009)

FcaBedrock, a Formal Context Creator

Simon Andrews and Constantinos Orphanides

Conceptual Structures Research Group, Communication and Computing Research Centre, Sheffield Hallam University, Sheffield, UK
s.andrews@shu.ac.uk, corphani@my.shu.ac.uk

Abstract. FcaBedrock employs user-guided automation to convert c.s.v. data sets into Burmeister .cxt and FIMI .dat context files for FCA.

1 Introduction

Data often exists in the form of flat-files of comma separated values. For FCA to be carried out, these data must be converted into Formal Contexts. Many tools exist to carry out analysis of Formal Contexts but few exist that carry out this preparatory task. Elba performs this task for data-base tables to supply ToscanaJ with Formal Contexts [3], but FcaBedrock deals with flat-file data, producing Formal Context files that can be used by a number of tools and programs. Moreover, FcaBedrock has been developed to convert large data sets into Formal Contexts. It has now been made available at *Sourceforge*[1]. FcaBedrock discretizes and Booleanizes data; taking each many-valued attribute and converting it into as many Boolean attributes as it has values and converting continuous values using ranges [4]. Data can be interpreted in many ways leading to inconsistent analysis and problems in measuring the performance of FCA algorithms [1,5]. FcaBedrock solves these problems by documenting data conversions in re-usable, editable, meta-data files called Bedrock files (Figure 1).

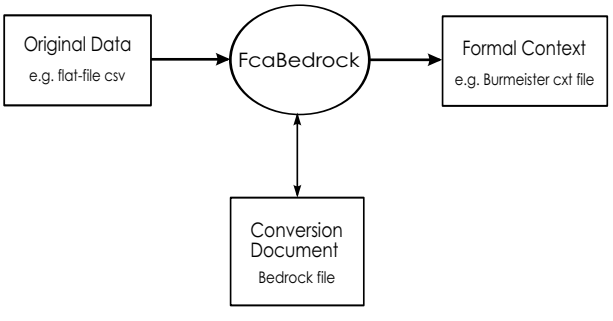


Fig. 1. FcaBedrock Process

[1] http://sourceforge.net/projects/fcabedrock/

M. Croitoru, S. Ferré, and D. Lukose (Eds.): ICCS 2010, LNAI 6208, pp. 181–184, 2010.

2 Operation

Figure 2 shows FcaBedrock. There are fields for the names and types of the original data attributes, whether they are to be converted, their categories and the corresponding category values found in the data file. These can be entered by the user, input via a Bedrock file or auto-detected from a data-file. The names of the categories and the category file values are not always the same, so FcaBedrock uses both; the category values are required for converting data and the category names appear in the Context file. The example shown is the *Mushroom* data set from the UCI Machine Learning Repository [2].

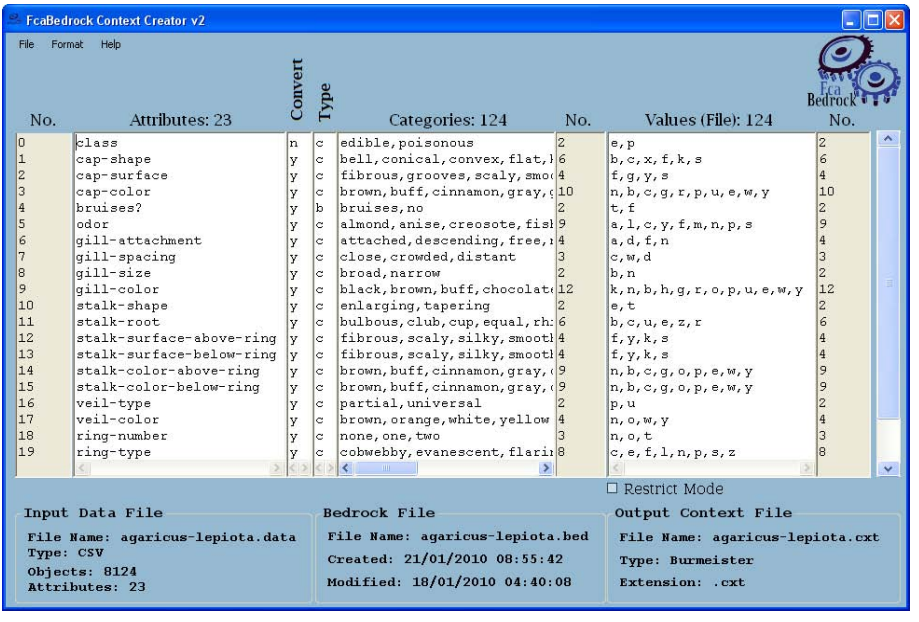


Fig. 2. FcaBedrock

3 Capabilities

The following is a list of some of the capabilities of FcaBedrock (for a more detailed description, see the software documentation at *Sourceforge*). Figure 3 illustrates some of features using the *Adult* data set from UCI [2].

- Data types converted: categorical, continuous and Boolean.
- Input file formats: Many column csv, three-column csv (triples).
- Output file formats: Burmeister (.cxt), FIMI format (.dat)[2].
- Auto-detection of meta-data from data-file.

[2] `http://fimi.cs.helsinki.fi/`

– Interpretation of data:
  - Restrict conversion to user-defined values.
  - Exclude from conversion user-defined values.
  - Freedom over treatment of missing values.

No.	Attributes: 15	Convert	Type	Categories: 103	No.	Restrict Values (File): 5	No.
0	age	y	o	<,20,40,60,>	4		0
1	workclass	y	c	Private,Self-emp-not-inc,S	8		0
2	fnlwgt	n	c		0		0
3	education	y	c	Bachelors,Some-college,11t	16	Bachelors, Masters, Doc	3
4	education-num	n	c		0		0
5	marital-status	y	c	Married-civ-spouse,Divorc	7		0
6	occupation	y	c	Tech-support,Craft-repair,	14		0
7	relationship	y	c	Wife,Own-child,Husband,Not	6		0
8	race	y	c	White,Asian-Pac-Islander,A	5		0
9	sex	y	c	Female,Male	2	Female	1
10	capital-gain	n	c		0		0
11	capital-loss	n	c		0		0
12	hours-per-week	n	c		0		0
13	native-country	y	c	United-States,Cambodia,Eng	41		0
14	class	n	c	>50K,<=50K	2	>50K	1

Fig. 3. Creating an *Adult* sub-context using restrict-to values

The following is an example adapted from the UCI *Adult* data set, using a data file called `mini-adult.data` with eight instances and five attributes (*age*, *education*, *employment*, *sex* and *US-citizen*) plus a *salary* class. File 2 shows a corresponding output from FcaBedrock in the cxt format.

```
39, Bachelors, Clerical, Male, Yes, <=50K
50, Bachelors, Managerial, Female, Yes, <=50K
38, HS-grad, Unskilled, Male, Yes, <=50K
53, 11th, Unskilled, Male, Yes, <=50K
28, Bachelors, Professional, Female, Yes, >50K
37, Masters, Managerial, Female, No, <=50K
49, ?, Clerical, Female, No, <=50K
52, HS-grad, Managerial, Male, Yes, >50K
```

File 1. `mini-adult.data`

4 Evaluation

An initial version FcaBedrock was evaluated by a class of final-year Computing undergraduates at Sheffield Hallam University (SHU). Results of this evaluation fed into the development of the version now at *Sourceforge*. This version was then evaluated by successfully converting a number of data sets into Formal Contexts, including the *Mushroom*, *Adult*, *Internet Advertisements*, *Flags* and *Tic-tac-toe* data sets from UCI and several internal student information and supermarket data sets at SHU. The Context files produced were successfully and consistently processed by two Formal Concept generators.

```
B       7                       employment-Unskilled
        age-<30                 sex-Male
8       age-30to<40             sex-Female
15      age-40to<50             US-citizen
        age->=50                .X..X...X...X.X
0       education-Bachelors     ...XX....X...XX
1       education-Masters       .X.....X...XX.X
2       education-11th          ...X..X....XX.X
3       education-HS-grad       X...X.....X..XX
4       employment-Clerical     .X...X...X...X.
5       employment-Managerial   ..X.....X....X.
6       employment-Professional ...X...X.X..X.X
```

File 2. `mini-adult.cxt`, *Burmeister* context file.

There are several other file formats used in FCA, obtainable from a cxt file produced by FcaBedrock using the conversion tool, *FcaStone* [7]. A version of FcaBedrock is being developed that takes RDF-S and OWL as input [6]. This work will form a core part of CUBIST ("Combining and Uniting Business Intelligence with Semantic Technologies"), awarded under the European Union's 7th Framework Programme, 5th ICT call, topic 4.3: Intelligent Information Management; STREP Project No.: FP7 257403.

References

1. Andrews, S.: Data Conversion and Interoperability for FCA. In: CS-TIW 2009, pp. 42–49 (2009), `http://www.kde.cs.uni-kassel.de/ws/cs-tiw2009/proceedings_final_15July.pdf`
2. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine, CA (2007), `http://www.ics.uci.edu/~mlearn/MLRepository.html`
3. Becker, P., Correia, J.H.: The ToscanaJ Suite for Implementing Conceptual Information Systems. In: Ganter, B., Stumme, G., Wille, R. (eds.) Formal Concept Analysis. LNCS (LNAI), vol. 3626, pp. 324–348. Springer, Heidelberg (2005)
4. Ganter, B., Wille, R.: Conceptual Scaling. In: Roberts, F. (ed.) Applications of Combinatorics and Graph Theory to the Biological and Social Sciences. IMA, vol. 17, pp. 139–168. Springer, Heidelberg (1989)
5. Kuznetsov, S.O., Obiedkov, S.A.: Comparing Performance of Algorithms for Generating Concept Lattices. Journal of Experimental and Theoretical Artificial Intelligence 14, 189–216 (2002)
6. Passin, T.B.: Explorer's Guide to the Semantic Web, Manning, Greenwich, CT (2004)
7. Priss, U.: FcaStone - FCA File Format and Interoperability Software. In: Croitoru, M., Jaschkë, R., Rudolph, S. (eds.) CS-TIW 2008, pp. 33–43 (2008)

From Generalization of Syntactic Parse Trees to Conceptual Graphs

Boris A. Galitsky[1], Gábor Dobrocsi[2], Josep Lluis de la Rosa[1] and Sergey O. Kuznetsov[3]

[1] Univ. Girona Spain
[2] Univ Miskolc Miskolc Hungary
[3] Higher School of Economics, Moscow Russia

Abstract. We define sentence generalization and generalization diagrams as a special sort of conceptual graphs which can be constructed automatically from syntactic parse trees and support semantic classification task. Similarity measure between syntactic parse trees is developed as a generalization operation on the lists of sub-trees of these trees. The diagrams are representation of mapping between the syntactic generalization level and semantic generalization level (anti-unification of logic forms). Generalization diagrams are intended to be more accurate semantic representation than conventional conceptual graphs for individual sentences because only syntactic commonalities are represented at semantic level.

1 Introduction

Proceeding from parsing to semantic level is an important task toward natural language understanding, and has immediate applications in tasks such information extraction and question answering [1,4,7]. In the last ten years there has been a dramatic shift in computational linguistics from manually constructing grammars and knowledge bases to partially or totally automating this process by using statistical learning methods trained on large annotated or non-annotated natural language corpora.

In this study we attempt to approach conceptual tree using pure syntactic information such as syntactic parse trees. We explore the possibility of high-level *semantic* classification of natural language sentences based on *full syntactic parse trees*. We address semantic classes which appear in information extraction and knowledge integration problems usually requiring deep natural language understanding [2,3,5]. One of such problems is search relevancy, measuring semantic similarity between questions and answers by matching respective parse trees.

The main question of this study is what kind of semantic patterns can be inferred from complete parse tree structure. We believe that applying graph-based machine learning technique to such structure as syntactic trees, which have rather weak links to high-level semantic properties, can deliver satisfactory semantic classification results.

Learning syntactic parse trees allows performing semantic inference in a domain-independent manner without using ontologies. We apply parse tree generalization techniques to solving the following the problem of classifying search results in respect to being relevant and irrelevant to search query.

M. Croitoru, S. Ferré, and D. Lukose (Eds.): ICCS 2010, LNAI 6208, pp. 185–190, 2010.

2 Generalizing Natural Language Sentences

To measure similarity of abstract entities expressed by logic formulas, a least-general generalization was proposed for a number of machine learning approaches, including explanation based learning and inductive logic programming. It is the opposite of most general unification [6] therefore it is also called anti-unification. To measure similarity between natural language (NL) expressions, we extend the notion of generalization from logic formulas to syntactic parse trees of these expressions. If it were possible to define similarity between natural language expressions at pure semantic level, least general generalization would be sufficient. However, in horizontal search domains where construction of full ontologies for complete translation from NL to logic language is not plausible, therefore extension of the abstract operation of generalization to syntactic level is required. Rather then extracting common keywords, generalization operation produces a syntactic expression that can be semantically interpreted as a common meaning shared by two sentences.

1) Obtain parsing tree for each sentence. For each word (tree node) we have lemma, part of speech and form of word information. This information is contained in the node label. We also have an arc to the other node.
2) Split sentences into sub-trees which are phrases for each type: verb, noun, prepositional and others; these sub-trees are overlapping. The sub-trees are coded so that information about occurrence in the full tree is retained.
3) All sub-trees are grouped by phrase types.
4) Extending the list of phrases by adding equivalence transformations. Generalize each pair of sub-trees for both sentences for each phrase type.
5) For each pair of sub-trees yield an alignment, and then generalize each node for this alignment. For the obtained set of trees (generalization results), calculate the score.
6) For each pair of sub-trees for phrases, select the set of generalizations with highest score (least general).
7) Form the sets of generalizations for each phrase types whose elements are sets of generalizations for this type.
8) Filtering the list of generalization results: for the list of generalization for each phrase type, exclude more general elements from lists of generalization for given pair of phrases.

For a pair of phrases, generalization includes all *maximum* ordered sets of generalization nodes for words in phrases so that the order of words is retained. In the following example

To buy digital camera today, on Monday

Digital camera was a good buy today, first Monday of the month

Generalization contains {*digital - camera , today – Monday*} , where part of speech information is not shown. *buy* is excluded from both generalizations because it occurs in a different order in the above phrases. *Buy - digital - camera* is not a generalization because *buy* occurs in different sequence with the other generalization nodes.

Result of generalization can be further generalized with other parse trees or generalizations. For a set of sentences, the totality of generalizations forms a lattice: order on generalizations is set by the subsumption relation and generalization score. Generalization of parse trees obeys the associativity by means of computation: it has to be verified and resultant list extended each time new sentence is added.

3 From Generalization to Logical Form Representation

We now demonstrate how the generalization framework can be combined with the semantic representation such as logic forms to perform learning of text meaning. We have demonstrated how semantic features can be deduced from syntactic parse trees when appropriate similarity operation is found. However in a number of applications certain semantic knowledge is available, so it does not have to be learned. In this section we show how to combine pre-set semantic information with the learned one to build most accurate semantic representation.

We use notes from a number of customers of a bank. The dataset of three paragraphs is introduced:

1p. A friend transferred funds from a checking to a savings account. He then used the saving funds to pay for his mortgage.
2p. Premier account customers decided to transfer their funds from premier to regular savings account. The couple then used their premier account for automated mortgage payment.
3p. A mortgagee customer transferred the mortgage account from fixed to adjustable. She then decided to use the remaining funds as a last payment of mortgage for her second home.

To demonstrate a deep level understanding of meanings of these paragraphs, let us introduce two classes of "individual bank users" and "corporate bank users" and demonstrate how these classes can be formed from our data and classification performed. Notice that there is no explicit indication of belonging to one of this classes, it has to be inferred from text. There could be other classes where semantic information has to be inferred such as 'obtained funds are used for something' and 'no such statement is made', 'account type transfer' and 'refinancing', and many more.

We intend to express commonalities between the elements of training set to 'explain' belonging to a class, following the classical methodology of induction (Mill 1843). We hypothesize that common linguistic features of a training set *cause* the target feature (the class). In this section we form these features on both syntactic level by means of generalization and on semantic level by means of logical anti-unification. To do that, we will first proceed on syntactic level, and then show how it can be done on semantic level of logic forms. Then we finally show how the syntactic level can be mapped into semantic one. Single lines depict generalizations for the first sentence of each paragraph (a), double line – for the second sentence (b). There are multiple sentences appearing in different order in a general case. The lattice depicts the relation of "being more general' between generalization results.

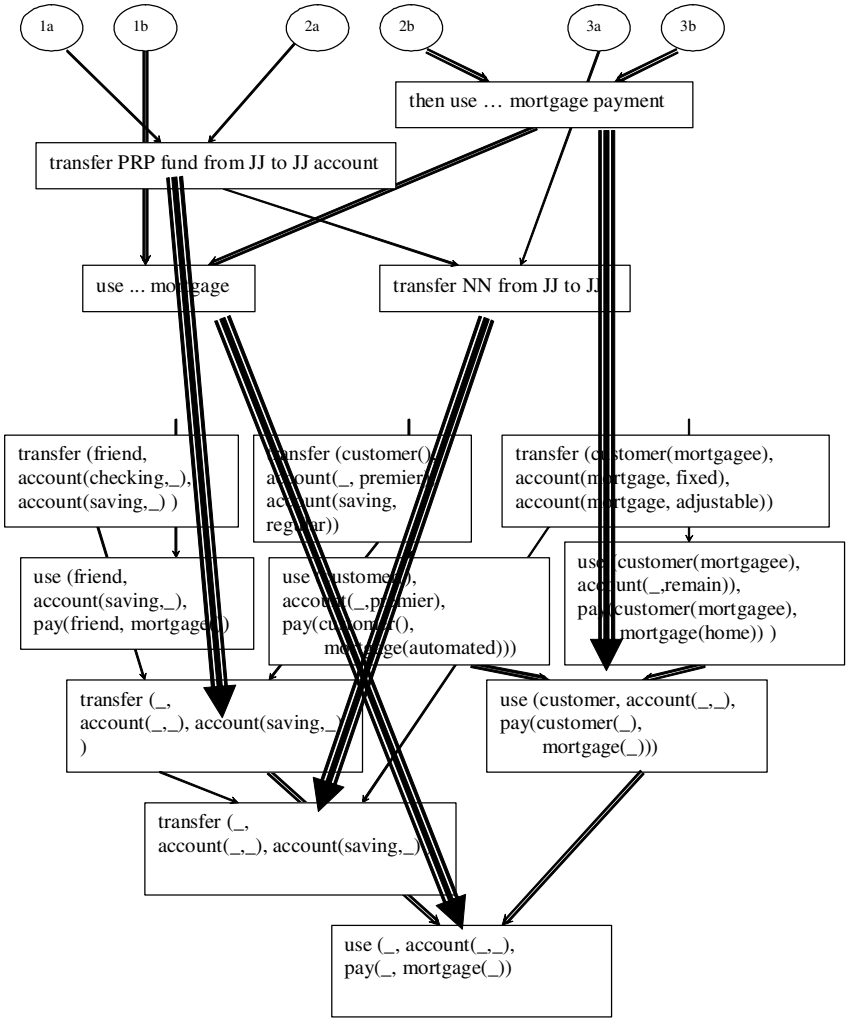


Fig. 1. Conceptual graph for tree paragraphs

To define mapping into logic forms, we need to form logical predicates and specify semantic types of their arguments. For a pair sentences, we can first generalize them and then translate result into a logic form. Alternatively, we can translate each sentence into logic form first and then anti-unify these logic forms. Fig. 1 shows multiple paths to the results of operations of generalization and anti-unification. There is a criterion for optimal path: the resultant score of expression. For a logic form, the score is a number of terms in the expression; this fits well the score of generalization. We define an *optimal path* to the logic form of a set of samples as the one leading to the resultant logic form with the highest score.

In Fig. 1 we visualize our version of the conceptual graph for three sentences above. There is a lattice of generalizations for three paragraphs from positive set (on the top), and a lattice of anti-unifications for three paragraphs (on the bottom). There is a mapping between syntactic and semantic levels. Results of generalization of sentences are mapped into anti-unification of respective logic forms.

4 Evaluation and Conclusion

Evaluation of search included an assessment of classification accuracy for search results as relevant and irrelevant. Since we used the generalization score between the query and each hit snapshot, we drew a threshold of five highest score results as relevant class and the rest of search results as irrelevant. We used the Yahoo search API and applied the generalization score to find the highest score hits from first fifty Yahoo search results. We then consider the first five hits with the highest generalization score (not Yahoo score) to belong to the class of relevant answers. Third and second rows from the bottom contain classification results for the queries of 3-4 keywords which is slightly more complex than an average one (3 keywords); and significantly more complex queries of 5-7 keywords respectively.

The total average accuracy (F-measure) for all above problems is 79.2%. Since the syntactic generalization was the only source of classification, we believe the accuracy is satisfactory. A practical application would usually use a hybrid approach with rules and keyword statistic which would deliver higher overall accuracy, but such application is beyond the scope current paper. Since the generalization algorithm is deterministic, higher accuracy can be also achieved by extending training set.

Table 1. evaluation of classification accuracy

Type of search query	Relevancy of Yahoo search, %, averaging over 10	Relevancy of re-sorting by generalization, %, averaging over 10	Relevancy comp to baseline, %
3-4 word phrases	77	77	100.0%
5-7 word phrases	79	78	98.7%
8-10 word single sentences	77	80	103.9%
2 sentences, >8 words total	77	83	107.8%
3sentences,>12 words total	75	82	109.3%

In this study we demonstrated that such high-level sentences semantic features as *being informative* can be learned from the low level linguistic data of complete parse tree. Unlike the traditional approaches to *multilevel* derivation of semantics from syntax, we explored the possibility of linking low level but detailed syntactic level with high-level pragmatic and semantic levels *directly*.

For a few decades, most approaches to NL semantics relied on mapping to First Order Logic representations with a general prover and without using acquired rich knowledge sources. Significant development in NLP, specifically the ability to acquire knowledge and induce some level of abstract representation is expected to

support more sophisticated and robust approaches. A number of recent approaches are based on shallow representations of the text that capture lexico-syntactic relations based on dependency structures and are mostly built from grammatical functions extending keyword matching (Durme et al 2003). Similarly to the above studies, we address the semantic inference in a domain-independent manner. Syntactic match allows solving problems of semantic relevancy without use of ontologies, therefore finding a number of commercial applications including relevancy engine at citizens' journalism portal AllVoices.com.

References

1. Allen, J.F.: Natural Language Understanding. Benjamin Cummings (1987)
2. Banko, M., Cafarella, J., Soderland, S., Broadhead, M., Etzioni, O.: Open information extraction from the web. In: Proceedings of the Twentieth International Joint Conference on Artificial Intelligence, Hyderabad, India, pp. 2670–2676. AAAI Press, Menlo Park (2007)
3. Dzikovska, M., Swift, M., Allen, J., de Beaumont, W.: Generic parsing for multi-domain semantic interpretation. In: International Workshop on Parsing Technologies (Iwpt 2005), Vancouver BC (2005)
4. Cardie, C., Mooney, R.J.: Machine Learning and Natural Language. Machine Learning 1(5) (1999)
5. Galitsky, B.: Natural Language Question Answering System: Technique of Semantic Headers. Advanced Knowledge International, Australia (2003)
6. Robinson, J.A.: A machine-oriented logic based on the resolution principle. Journal of the Association for Computing Machinery 12, 23–41 (1965)
7. Ravichandran, D., Hovy, E.: Learning surface text patterns for a Question Answering system. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002), Philadelphia, PA (2002)
8. Durme, B.V., Huang, Y., Kupsc, A., Nyberg, E.: Towards light semantic processing for question answering. In: HLT Workshop on Text Meaning (2003)

Conceptual Structures for Reasoning Enterprise Agents

Richard Hill

School of Computing
University of Derby,
Derby, DE22 1GB, UK
r.hill@derby.ac.uk

Abstract. If agents are to facilitate interoperability at the knowledge level, they must be able to implement plans in response to what they perceive about their own environment, together with any future intentions. As potential business risk-takers, enterprise agents must be able to take advantage of new scenarios by understanding the ontologies of other enterprises. Conceptual Structures such as Conceptual Graphs (CG) are a more natural way of expressing business transactions. Coupled with the Belief, Desire and Intention (BDI) model, the Transaction Model (TM), expressesd as CG, is used as part of an agent's model for reasoning on a particular course of action. A brief exemplar illustrates how a reasoning Multi-Agent System application, governed by the Transaction Model, can be implemented using Transaction Agent Modelling (TrAM) and tools such as Amine, Jason and JADE.

1 Introduction

One of the challenges of managing enterprise knowledge is the inclusion of existing disparate, heterogenous systems that are already in use. This is compounded by a service approach to business where the expansion of an enterprise is realised by the acquisition or interoperation with external services, who themselves present disparate knowledge architectures. This is a compelling case for agent based architectures. For the agents to be able to interoperate at the knowledge level they need to be able to reason between what they know about themselves and their environment, and what they intend to do. One approach is to utilise ontologies as a means of describing knowledge artefacts within a given domain. The ontological representation is then updated and appended as required by the relevant parties which consume it during the course of their interactions. This paper proposes an approach to inform a Belief, Desire and Intention[2] agent's operations within an enteprise knowledge architecture, by way of a transaction-based conceptual catalogue[3].

2 Transaction Agent Modelling

The Transaction Model (TM) has illustrated how daily, operational, qualitative decisions can be represented in a way that is better understood by an

M. Croitoru, S. Ferré, and D. Lukose (Eds.): ICCS 2010, LNAI 6208, pp. 191–194, 2010.

organisation. Prior work[3] briefly presents the REA[7] accounting model, in such a way that models derived from the resulting ontology are therefore based upon sound accounting principles. Transaction Agent Modelling (TrAM)[4] uses REA and the TM together, supporting the generation of models that have a robust transaction base, with a common understanding of the core concepts and their meaning. This understanding is assisted by conceptual structures since they provide a more natural representation of the knowledge within an enterprise.

3 The Process

Table 1 outlines the key tasks associated with this approach. In brief, strategic enterprise goals are modelled as high-level transactions. These strategic goals are then broken down into sub-goals by delegation to individual business functions, and by further analysis using TrAM[8]. From the analysis, agent roles and associated sub-goals are derived. Communicative acts are then defined for each agent role using the TM Conceptual Catalogue[3]. Finally environmental details are added to ensure that the agents respond accordingly to known and unknown events.

Table 1. Key tasks for building enterprise agents together with associated tools

Task	Description	Tool
Early requirements	Identify strategic goals of enterprise	Amine[1]
Requirements analysis	Represent goals and transactions as CG	Amine
Design specification 1	Identify internal and external agent roles.	Jason[6]
Design specification 2	Map communicative acts to agent goals.	Jason
Implementation	Specify environment details.	JADE[5]

4 An Exemplar Case Study

This work refers to the running case study in the community healthcare domain, described in[4,8]. For convenience the generic TM graph is described below using Amine syntax[1]:

```
[Act:super]-
 -part->[economicEvent:a]-
  -event_subject->[economicResource:x]-
     -source->[insideAgent:i],
      -destination->[outsideAgent:o];;
 -part->[Economic_Event:b]-
  -event_subject->[economicResource:y]-
     -source->[outsideAgent:o],
     -destination->[insideAgent:i]
```

After various specialisations there is a scenario whereby an `Older Person` undertakes a transaction with a `Care Provider` (For further case study detail see:[3,4,8]). The `Older Person` is unable to administer insulin injections and therefore relies on a visiting `Care Provider`. In this situation there is a need to maintain the correct dosage and frequency of the drug for the `Older Person`. After making a request, the `Older Person` agent delegates a goal to the `Care Provider` agent; specifying the outcome but not dictating how it should be achieved. There is also another dimension to the situation. The `Older Person` has limited mobility and therefore rarely leaves their home. Social contact is valued greatly so there are periodic prompts from the `Older Person` agent to the `Care Provider` agent for some conversation. Of course an even more pro-active system would have the `Care Provider` agent manage 'social contact' as a goal as well. Since it is possible to easily add goals as they are modelled using the TrAM based approach, it is now straightforward to add these to the implementation.

For brevity only the sample Jason[6] code for the `Older Person` agent is shown:

```
// Agent olderPerson in project CommCare.mas2j
/* Initial goals */
!get(insulin). //initial goal
// verify whether I am getting sufficient social contact
!check_lonely.
/* Plans to trigger in response to [economicEvent] */
+!get(insulin) : true
// delegate achieve goal to careProvider Agent
    <- .send(careProvider, achieve, has(olderPerson,insulin)).
+!has(olderPerson,insulin) : true
    <- !eat(food).
+!has(olderPerson,insulin) : true
    <- !get(insulin).
+!eat(food) : has(olderPerson,insulin)
    <- nibble(food);
    !eat(food).
+!eat(food) : not has(olderPerson,insulin)
    <- true.
+!check_lonely : true // Sometimes I get lonely
    <- .random(X); .wait(X*5000+2000);
       // I would like some social contact from time to time
      .send(careProvider, askOne, time(_), R);
      .print(R);
      !!check_lonely.
```

5 Conclusions

This paper argues by way of a brief exemplar that conceptual structures are a means of representing a more natural form of transaction model in enterprise-wide

knowledge architectures. Enterprise agents need to cross boundaries and engage in business transactions with other enterprises and other knowledge bases. As such, there is a need for a common representation of enterprise knowledge that is rich and expressive, yet can be interpreted and reasoned against. In particular, the use of a vocabulary that is goal-oriented has served to simplify the translation from requirements model to design specification, and now, through to implementation onto de facto Multi-Agent System (MAS) platforms such as JADE. Using a BDI model of agency, an exemplar enterprise function is considered and represented as CG, before a transaction-based conceptual catalogue is applied that enables a design specification for individual enterprise agents to be produced. This work builds upon the use of CG to support the capture of enterprise semantics at the early requirements stage[4] as well as providing the rigour for any subsequent analysis, by providing a tangible route forward for building MAS applications.

References

1. Amine Platform, `http://amine-platform.sourceforge.net/` (last accessed 29th January 2010)
2. Bratman, M.E., Israel, D.J., Pollack, M.E.: Plans and resource-bounded practical reasoning. Computational Intelligence 4, 349–355 (1988)
3. Hill, R., Polovina, S.: An Automated Conceptual Catalogue for the Enterprise. In: Eklund, P., Haemmerlè, O. (eds.) Supplementary Proceedings of the 16th International Conference on Conceptual Structures (ICCS 2008), Toulouse, France, July 7-11, vol. 354, pp. 99–106. Published by CEUR-WS (2008), ISSN: 1613-0073
4. Hill, R.: Capturing and Specifying Multi-Agent Systems for the Management of Community Healthcare. In: Yoshida, H., Jain, A., Ichalkaranje, A., Jain, L.C., Ichalkaranje, N. (eds.) Advanced Computational Intelligence Paradigms in Healthcare - 1, ch. 6. SCI, vol. 48, pp. 127–164. Springer, Berlin (2006), ISBN: 978-3-540-47523-1
5. Java Agent DEvelopment Framework (JADE), `http://jade.tilab.com/` (last accessed 29th January 2010)
6. Jason, a Java-based interpreter for an extended version of AgentSpeak, `http://jason.sourceforge.net/JasonWebSite/Jason%20Home.php` (last accessed January 29, 2010)
7. McCarthy, W.E.: The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment. The Accounting Review, 554–578 (1982)
8. Polovina, S., Hill, R.: Enhancing the Initial Requirements Capture of Multi-Agent Systems through Conceptual Graphs. In: Dau, F., Mugnier, M.-L., Stumme, G. (eds.) ICCS 2005. LNCS (LNAI), vol. 3596, pp. 439–452. Springer, Heidelberg (2005)
9. Sowa, J.F.: Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley, Reading (1984)

Conceptual Graphs for Semantic Email Addressing

Dat T. Huynh and Tru H. Cao

Faculty of Computer Science and Engineering
Ho Chi Minh City University of Technology, Vietnam
{htdat,tru}@cse.hcmut.edu.vn

Abstract. Nowadays, email has become pervasive and played an important role in an information society. However, in order to send an email to a person or a group of people, senders have to specify exactly the email addresses of recipients. Meanwhile, semantic email addressing is a technique that allows senders to describe a specific group of recipients semantically. In this paper, we propose a novel method of using conceptual graphs to represent the information of recipients for semantic email addressing.

Keywords: Email receiver description, ontology, RDF query language, conceptual graph mapping.

1 Introduction

Accompanying with the exponential growth of information on the World Wide Web, the requirements of transferring information and communication via email become more and more complicated and professional. The task of processing a large number of emails everyday costs a lot of time and effort for humans. Moreover, in order to send an email to a person and a group of people, senders must know in advance and specify exactly addresses of receivers. However, the email addresses are changed from time to time. As a result, the traditional email system cannot work if senders do not remember and update receivers' addresses.

Semantic email is a new research trend of email processing in which meta-tags are embedded into emails to describe their semantic information for machine consumption ([1], [4], [5], [8], [9]). In general, the previous research works focused on the improvement of email processing rather than the specification of recipients. Meanwhile, Semantic Email Addressing, or SEA, is a novel term in the research field of semantic email recently proposed by [7] to describe the email addresses of recipients.

In this paper, we propose a novel method in which we exploit conceptual graphs (CGs) to help senders express information of receivers for SEA. A sender can compose CGs to describe the information of receivers at the interface layer of email clients. Then, those CGs are processed by the semantic email server to retrieve receivers' email addresses for sending messages. For graph matching problem between CGs and RDF graphs in the ontology of discourse, we exploit

M. Croitoru, S. Ferré, and D. Lukose (Eds.): ICCS 2010, LNAI 6208, pp. 195–198, 2010.

the subsumption mechanism of SeRQL (a query language) in Sesame ([2]), a robust knowledge management system. Firstly, CGs are converted into SeRQL queries by a semantic email server. Next, the SeRQL queries are executed by the Sesame query engine to retrieve actual addresses of receivers that satisfy requirements of the sender. Eventually, those email addresses will be used to distribute the message of the sender to all corresponding receivers.

The rest sections of this paper are organized as follows. Section 2 presents our proposed method of using SEA CGs to describe receivers in SEA. Subsequently, Section 3 discusses the mapping of CGs to SeRQL queries and retrieval of knowledge of recipients from an ontology. Finally, Section 4 concludes the paper and suggests some future works.

2 Conceptual Graphs for Receiver Description

For semantic email addressing, [7] proposed an idea of using patterns to define a group of recipients for SEA. Although the usage of fixed patterns for representing information of recipients is simple and helpful in some cases, they are not sufficient and flexible enough for more expressive representation, which allows to represent the attributes of receivers, their relations with other entities, and constraints on their property values. As a consequence, users cannot describe on the fly arbitrary target recipients unless the system provides all possible predefined patterns.

Meanwhile, CGs have a smooth mapping to and from natural language. In addition, due to the graph-based representation, they furnish not only non-expert users with friendly graphic interfaces but also the robust and flexible knowledge representation. Therefore, we propose the usage of CGs with the queried referents at interface level for semantic email addressing.

A SEA CG is defined as a non-nested CG whose concept referents can be either an individual referent, the generic referent, denoted by '*', or the queried referent, denoted by '?'. The generic referent means that it does not care about a matched individual referent. Meanwhile, the queried referent represents the receivers that users want to send emails to.

For examples, suppose that users want to send an email to all lecturers who are teaching the Compilers course, it can be composed as the SEA CG F in Fig. 1. Similarly, the SEA CG G in the figure expresses that projected email receivers are students and those lecturers who supervise them. Meanwhile, the SEA CG H in the figure depicts a case in which users want to send emails to all lecturers who are teaching the same course with a lecturer whose full name is Huynh Tan Dat.

SEA CGs composed by senders must be checked for a well-formed CG with respect to the ontology before they are sent to the semantic email server. A SEA CG is well-formed if the concept type of each concept node connected to a relation node is the same as, or a sub-type of, the corresponding neighbor concept type of the relation type of that relation node as defined in the ontology. Moreover, for semantic email addressing, the concept types emerging with the

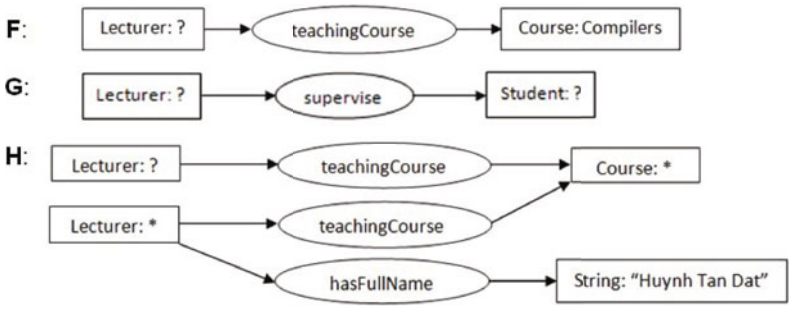


Fig. 1. Examples of SEA CGs

queried referents in the concept nodes of SEA CG must be sub-types of Person concept type in the ontology.

3 CG-SeRQL Mapping for Retrieving Recipients

Ontology plays an indispensable role in semantic email addressing because it not only provides meta-data for composing SEA CGs but also contains the knowledge of recipients in the system. In our work, an ontology in academic domain has been built by utilizing VN-KIM KBM ([6]) and based on Resource Description Framework (RDF), a standard model for representing Web information, whose statements can be viewed as graphs.

Given a SEA CG, retrieval of recipients' email addresses that satisfies the SEA CG leads to the problem of graph matching between SEA CGs and RDF graphs in the ontology. Although it was shown that there is a close mapping between CGs and RDF graphs ([10]), the realization of a system to retrieve RDF graphs is a formidable task. Meanwhile, many new technologies for management of knowledge and information, in particular on the Web, have already emerged. Sesame is considered as a good platform for storage and management of knowledge bases in RDF/RDFS format and has been widely used for semantic web systems. Sesame offers SeRQL, a powerful query language for RDF graphs with a good feature of the subsumption mechanism, where a concept or relation type in a query can match with its subtypes in a knowledge base.

Nevertheless, SeRQL is not suitable for end-users, who are not familiar with the language's syntax and RDF structure. With the graphical knowledge representation, CGs are more readable. Therefore, in order to exploit Sesame infrastructure, we employ SEA CGs introduced above at the interface level and map them to SeRQL for querying projected recipients from the ontology of discourse. We have adapted the idea from [3] to translate a SEA CG into its equivalent SeRQL clauses for retrieving recipients' email addresses. Furthermore, in our algorithm the path expression $\{x_i\}$ <*http://www.cse.hcmut.edu.vn/univ#hasOrg EmailAddress*> $\{e_i\}$ for each queried concept must be generated and augmented to FROM clause of the SeRQL query.

4 Conclusion

In this paper, we have proposed an idea of using CGs as a method to describe receivers in a semantic email system. SEA CGs provide users with a friendly interface and easily readable description of information of receivers. They are then mapped to SeRQL queries for retrieving receivers and their email addresses to send messages to.

Currently, SEA CGs defined in this paper are only simple ones. Meanwhile, users may want to send emails to the professors who are currently supervising the most PhD students in the faculty, for instance. To support that, SEA CGs are to be extended with aggregate functions and logical connectives. In addition, the translation from receivers' descriptions in natural language into CGs could be incorporated to develop more intelligent semantic email systems. Those are some research topics that we are currently investigating.

References

1. Balasubramanyan, R., Carvalho, V.R., Cohen, W.: CutOnce - Recipient Recommendation and Leak Detection in Action. In: Proceedings of EMAIL 2008: the AAAI Workshop on Enhanced Messaging (2008)
2. Broekstra, J., Kampman, A., Harmelen, F.V.: Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. In: Horrocks, I., Hendler, J. (eds.) ISWC 2002. LNCS, vol. 2342, pp. 54–68. Springer, Heidelberg (2002)
3. Cao, T.H., Huynh, D.T.: Subsumption Degrees between Entity Types and Names for Approximate Knowledge Retrieval. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 15(1), 21–42 (2007)
4. Celino, I., Corcoglioniti, F., Valle, E.D.: Towards a Semantic Contact Management. In: Proceedings of the 2nd International ISWC+ASWC Workshop on Finding Experts on the Web with Semantics (FEWS 2007), Busan, Korea, pp. 64–77 (2007)
5. Etzioni, O., et al.: Semantic Email: Adding Lightweight Data Manipulation Capabilities to the Email Habitat. In: Proceedings of 6th International Workshop on the Web and Databases, pp. 12–13. ACM Press, New York (2003)
6. Huynh, D.T., Cao, T.H., Ta, H.Q., Nguyen, L.H.: VN-KIM KBM: A Distributed and Collective Tool for Managing Semantic Web Knowledge Bases. In: Proceedings of the 1st Workshop on Human Factors and the Semantic Web (SWAHA 2008, in conjunction with ASWC 2008), pp. 153–158 (2008)
7. Kassoff, M., Petrie, C., Zen, L., Genesereth, M.: Semantic Email Addressing: The Semantic Web Killer App? In: IEEE Internet Computing, pp. 48–55 (January/February 2009)
8. Mcdowell, L., Etzioni, O., Halevy, A.: Semantic Email: Theory and Applications. Journal of Web Semantics: Science, Services and Agents on the World Wide Web 2(2), 153–183 (2004)
9. Scerri, S., Davis, B., Handschuh, S., Hauswirth, M.: Semanta - Semantic Email Made Easy. In: Aroyo, L., Traverso, P., Ciravegna, F., Cimiano, P., Heath, T., Hyvönen, E., Mizoguchi, R., Oren, E., Sabou, M., Simperl, E. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 36–50. Springer, Heidelberg (2009)
10. Yao, H., Etzkorn, L.: Conversion from the Conceptual Graph (CG) model to the Resource Description Framework (RDF) model. In: Proceedings of 12th International Conference on Conceptual Structures, pp. 98–114 (2004)

Introducing Rigor in Concept Maps

Meena Kharatmal and G. Nagarjuna

Homi Bhabha Centre for Science Education (TIFR),
V. N. Purav Marg, Mankhurd, Mumbai 400 088, India
meena@hbcse.tifr.res.in, nagarjun@gnowledge.org

Abstract. Although concept maps have been found to be effective in science education research, these are critiqued for being informal due to informal usage of relation and attribute names thereby resulting in ambiguity. Refined concept mapping, a development over the regular concept mapping is an approach towards introducing rigor and parsimony in representing knowledge. The method proposed suggests to substitute the ambiguous relation names with well-defined relation names to concepts consistently while mapping a domain. We suggest the use of this method for introducing rigor in concept mapping and position it among the other models of knowledge representation in an inverse semantic spectrum.

Keywords: concept maps, conceptual structures, disambiguation, education, knowledge representation, rigor.

1 Introduction

Concept map, a two-dimensional representation of knowledge, is a simple graphical form of knowledge representation method comprising of nodes (concepts) and arcs (linking phrases). Although concept maps have shown to have significant impact in education[1], the knowledge representation community critiqued it for being informal [2,3]. In an analysis of four different types of knowledge representation models [2], concept maps, being one among them, are claimed to be informal. Although concept maps are easy to construct, the maps drawn by different persons of the same domain often do not match. This is due to the choice of different linking phrases, though the concept names chosen are often the same. While this may serve the purpose of eliciting the knowledge of the learner, often due to lack of discipline (rules) the propositions cannot express the intended meaning since the linking phrases are chosen from natural language. Examples given in section 2. This obviously prevents them from being fit for a formal representation [4], but also the free usage of linking phrases does not lead to rigor in representation of scientific knowledge [5]. We shall propose a simple methodology to refine concept mapping so as to make the representation more clear and rigorous. We suggest how this method can help students become rigorous by re-representing concept maps as refined concept maps. We also relate the program with existing attempts to represent knowledge for machines, and how it can play the role of a bridge method linking the informal models and formal models of conceptual structures.

M. Croitoru, S. Ferré, and D. Lukose (Eds.): ICCS 2010, LNAI 6208, pp. 199–202, 2010.

2 Refined Concept Maps—A Methodology for Introducing Rigor

Refined Concept Mapping (RCM) uses a finite set of well-defined relation names consistently to represent a body of knowledge. The Novakian or Traditional Concept Mapping (TCM) uses linking words such as—*is a, can be, have, may be*, etc. Since these linking words do not portray the exact meaning, we propose to replace them with semantically well-defined relation names such as—*part of, includes, surrounded by, located in, has function*, etc. Explicit use of conceptual relations (predicates) exemplified with constraints in conceptual graphs have also been one of the insights that we have drawn from for focusing on well-defined relation names in representing knowledge [6]. Since our domain is of biology, we draw from the Open Biological and Biomedical Ontologies (OBO) [7] foundry which is collaboratively developing and publishing well-defined relations—the OBO Relation Ontology (RO) [8]. For e.g. the OBO defines *part_of as: For continuants: C part_of C′ if and only if: given any c′ that instantiates C at a time t, there is some c such that c′ instantiates C′ at time t, and c part_of c′ at t*. Similarly, *located_in is defined as: C located_in C′ if and only if: given any c that instantiates C at a time t, there is some c′ such that: c′ instantiates C′ at time t and c located_in c′*. In OBO, the relation names are categorized as foundational, temporal, spatial and participation [9]. The relation names are chosen based on the classification scheme—inclusion (class, meronymy, (component-object, member-collection, portion-mass, stuff-object, phase-activity, place-area, feature-event), spatial), possession, attachment, attribution, antonym, synonym and case [10].

Now, we shall illustrate the methodology to transform informal (natural language) propositions into refined propositions by replacing merely the relation and attribute names with well-defined ones. This is how we propose rigor can be introduced. The following are the propositions from the traditional maps:

(1) [sharks]→(can be)→ [great white shark, tiger shark]
(2) [shark teeth]→(can be)→ [big, small]
(3) [nucleus] → (is a)→ [double layered membrane structure]
(4) [nucleus]→ (is one of the)→ [organelles in a cell]
(5) [nucleus]→(is present in)→ [each living cell]
(6) [nucleus] → (is small)→ [in animal cell]

The corresponding RCM propositions following the same order are:

(1′) [sharks] → (includes) →[great white shark, tiger shark]
(2′) It is possible that, [shark teeth]→ (has size)→ [big, small]
(3′) [nucleus] → (enveloped by)→ [double layered membrane structure]
(4′) [nucleus]→ (kind of)→ [organelles in a cell]
(5′) [nucleus]→(part of)→ [each living cell]
(6′) [nucleus]→ (part of)→ [animal cell]; [nucleus] →(has size)→[small]

We eliminated ambiguous relation name *can be* by resolving it into *includes* in (1′), *has size* in (2′), and appropriate possibility modality expressed by *can*

be is inserted. In the propositions (3-6), one single relation name *is a* is being ambiguously used for four different meanings. This ambiguity is eliminated when substituted by appropriate relation names—*enveloped by, includes, part of, has size,* respectively shown in (3′-6′). The ambiguity of *is a* link is already being pointed by experts in the field of semantic network [11]. Unlike in TCM, in RCM we propose the possibility of inserting tags for modalities and quantifiers. Thus, along with rigor, the substitution also helped in enhancing expressivity.

We proposed this methodology for science education suggesting the transformation of conceptual structures of novices into experts and hypothesized that roots of rigorous representations lie in predicate terms [5,12,13]. The methodology adds weight to and is coherent with semantic holism [14], where the meaning of a node (term) arises by virtue of its position in the neighbourhood of the node, rather than from the node itself.

3 Discussion

Cognitive development studies, in the context of teaching-learning, compared the conceptual structures of novices and experts in terms of coherence, abstractness, parsimony, integration, explicitness etc. [15,16,17]. Following [16], we attempted to make the implicit meaning explicit by re-representing the relation names. Therefore, the suggested transformation method of RCM can be used in teaching-learning context. Although, the set of relation names in RCM are part of a natural language, the rigor is introduced by following well-defined terms consistently.

Knowledge Representation (KR) studies by both cognitive and computer scientists also suggest gradual transitions from less formal to more formal representations, implicit to explicit, more ambiguous to least ambiguous giving rise to an inverse spectrum as shown in Fig. 1 (partially adapted from [18]).

The current semantic web project is also making knowledge more and more explicit to enable partial interpretation by computers. Our suggested methodology of RCM may also contribute to meet the objective of semantic web by focusing on the predicate terms instead of object terms.

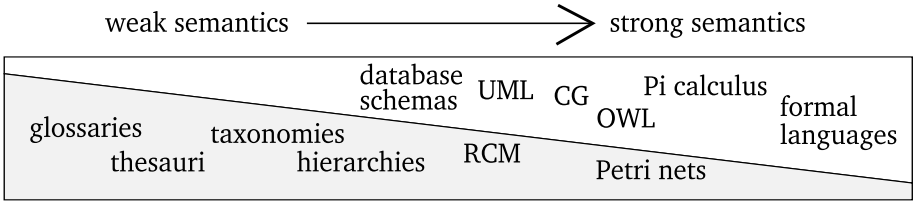


Fig. 1. Semantic spectrum presented indicating the inverse relation between ambiguity and rigor. The KR models on the left are more ambiguous and less rigorous whereas on the right are less ambiguous and more rigorous. The position, scale and the models presented are merely an indicative of the idea and not comprehensive.

References

1. Mintzes, J.J., Wandersee, J., Novak, J.D. (eds.): Teaching Science for Understanding– A Human Constructivist View. Academic Press, USA (1998)
2. Sowa, J.: Concept mapping. Talk Presented at the AERA Conference, San Francisco (2006), http://www.jfsowa.com/talks/cmapping.pdf
3. Kremer, R.: A Concept Mapping Tool to Handle Multiple Formalisms. In: Proceedings of AAAI Spring Symposium on Artificial Intelligence in Knowledge Management, pp. 86–93 (1997), http://www.aaai.org/Papers/Symposia/Spring/1997/SS-97-01/SS97-01-016.pdf
4. Canas, A.J., Carvalho, M.: Concept Maps and AI: An Unlikely Marriage? In: Proceedings of SBIE: Simposio Brasileiro de Informatica Educativa, Manaus, Brasil (2004)
5. Kharatmal, M., Nagarjuna, G.: A Proposal to Refine Concept Mapping for Effective Science Learning. In: Canas, A.J., Novak, J.D. (eds.) Concept Maps: Theory, Methodology, Technology. Proceedings of the Second International Conference on Concept Mapping, San Jose, Costa Rica (2006)
6. Sowa, J.: Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley Publishing Company, USA (1984)
7. The Open Biological and Biomedical Ontologies, http://www.obofoundry.org
8. The OBO Relation Ontology, http://www.obofoundry.org/ro/
9. Smith, B., Ceusters, W., Klagges, B., Kohler, J., Kumar, A., Lomax, J., Mungall, C., Neuhaus, F., Rector, A., Rosse, C.: Relations in Biomedical Ontologies. Genome Biology 6(5) (2005), http://genomebiology.com/2005/6/5/R46
10. Winston, M., Chaffin, R., Herrman, D.: A Taxonomy of Part-Whole Relations. Cognitive Science 11, 417–444 (1987)
11. Brachman, R.: What IS-A Is and Isn't: An Analysis of Taxonomic Links in Semantic Networks. IEEE Computer 16(10), 30–36 (1983)
12. Kharatmal, M., Nagarjuna, G.: Exploring Roots of Rigor: A Proposal of a Methodology for Analyzing the Conceptual Change from a Novice to an Expert. In: Canas, A.J., Reiska, P., Ahlberg, M., Novak, J.D. (eds.) Concept Mapping: Connecting Educators. Proceedings of the Third International Conference on Concept Mapping, Tallinn, Estonia & Helsinki, Finland (2008)
13. Kharatmal, M., Nagarjuna, G.: Refined Concept Maps for Science Education–A Feasibility Study. In: epiSTEME 3 Third International Conference on Review of Science, Technology and Mathematics Education, Mumbai, India (2009)
14. Quine, W.: From a Logical Point of View. In: Nine Logico-Philosophical Essays. Harvard University Press, USA (1953)
15. Brewer, W., Samarapungavan, A.: Children's Theories vs. Scientific Theories: Differences in Reasoning or Differences in Knowledge? In: Hoffman, Palermo (eds.) Cognition and the Symbolic Processes: Applied and Ecological Perspectives, pp. 209–232. Erlbaum, New Jersey (1991)
16. Karmiloff-Smith, A.: Beyond Modularity: A Developmental Perspective on Cognitive Science. MIT Press, USA (1995)
17. Nagarjuna, G.: Layers in the Fabric of Mind: A Critical Review of Cognitive Ontogeny. In: Ramadas, J., Chunawala, S. (eds.) Research Trends in Science, Technology and Mathematics Education. Homi Bhabha Centre for Science Education, Mumbai (2006)
18. McGuinness, D.: Ontologies Come of Age. In: Fensel, D., Hendler, J., Lieberman, J., Wahlster, W. (eds.) Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential. MIT Press, USA (2003)

Conceptual Knowledge Acquisition Using Automatically Generated Large-Scale Semantic Networks

Pia-Ramona Wojtinnek, Brian Harrington, Sebastian Rudolph, and Stephen Pulman

Oxford University Computing Laboratory, Oxford, UK
o wojt b to t o b o
Institute AIFB, Karlsruhe Institute of Technology, DE
o t

Abstract. We present a method for automatically creating large-scale semantic networks from natural language text, based on deep semantic analysis. We provide a robust and scalable implementation, and sketch various ways in which the representation may be deployed for conceptual knowledge acquisition. A translation to RDF establishes interoperability with a wide range of standardised tools, and bridges the gap to the field of semantic technologies.

1 Introduction

Graph-based models for representing conceptualizations have a long-standing history, ranging from expressive logical frameworks (as laid out in Peirce's work and further developed into conceptual graphs [1]) to widely applied graph-based Semantic Web formalisms like the Resource Description Framework (RDF) [2]. Graph-based representations of knowledge have been shown to provide both intuitive and formally rigorous access to the represented information.

In this work, we produce a graph-based conceptual model which provides a semantic middleground between statistical and symbolic formalisms: while it exhibits structural dependencies way beyond mere co-occurrence, it still features a fault tolerant way of representing the conceptual semantics of the original textual resource rather than providing a crisp logical description.

We further develop the ASKnet system [3] for conceptual knowledge acquisition and representation. ASKNet uses NLP tools to extract semantic information from text, and then, through a novel use of spreading activation theory, combines that information into an integrated large-scale semantic network. By mapping together concepts and objects that relate to the same real-world entities, ASKNet is able to produce a single unified entity relationship style semantic network. Combining information from multiple sources results in a representation which can reveal information that could not have been obtained from analyzing the original sources separately.

We modify the ASKnet framework to represent the conceptual backbone of a given text corpus in an aggregated yet structurally informative way and present its use for Word Sense Induction and as a representation of word context. Furthermore, in Section 3, we provide a translation of the described graph model into RDF and sketch the plethora of benefits that arise from the interoperability achieved by the alignment with this wide-spread, standardised, graph-based Semantic Web KR formalism. An extended version of this publication is available as technical report [4].

M. Croitoru, S. Ferré, and D. Lukose (Eds.): ICCS 2010, LNAI 6208, pp. 203–206, 2010.

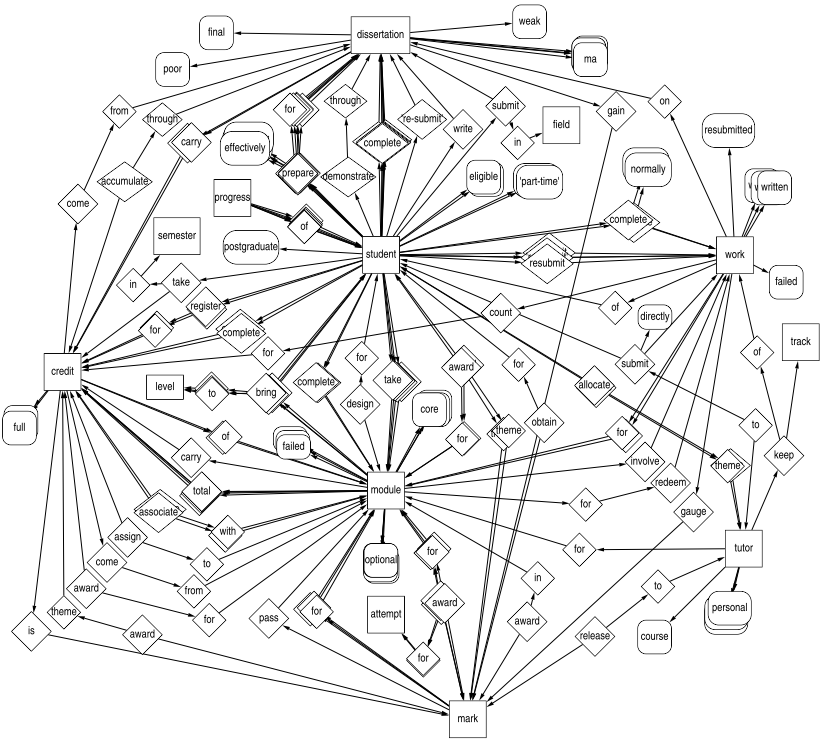
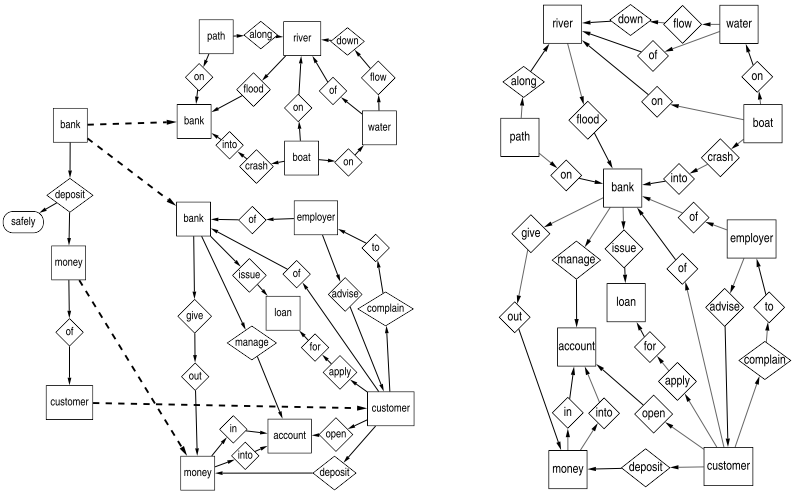


Fig. 1. Subgraph displaying a few concepts and relations from sample network

2 Developments for Conceptual Knowledge Acquisition

We build semantic networks for conceptual knowledge acquisition using the AskNet approach. By integrating the information on concepts, instead of Named Entities, we construct a network representing concepts and relations between them. The network is entirely based on the deep semantic parse of the given text provided by Boxer [5]. Figure 1 shows a subgraph of a sample network, which was built from a few paragraphs taken from Graduate Studies Handbooks. Multiple occurrences of the same relations have been pictured as overlapping. Frequency provides ground for weighting of relations, while the details may vary depending on the application at hand. Only a subset of the occurring relation types are depicted (those from verbs, prepositional modifiers and attributes). Complex structures such as from propositions (*Students are required to complete a dissertation*) have been left out for visibility purposes, but are represented in the full network via reification. The benefit of the network structure lies in its dense and interconnective representation of the syntactically based relations in a cross-sentence and cross-document way. In the sample text, *dissertation* and *module* rarely co-occurred in a sentence, but the network shows their strong connection over *student* and explicitly specifies the relations. We sketch ways in which both the building process and the resulting network can be used for conceptual knowledge acquisition.



(a) update algorithm (b) cluster identification

Fig. 2. Two WSI strategies

Our first aim is to automatically learn from the network which nodes correspond to the same concept, tackling the gap between words and concepts. One subproblem is distinguishing word senses: for a polysemous word, we aim to have one node per sense in the resulting network, merging all occurrences to the correct node. This corresponds to unsupervised Word Sense Induction and Discrimination [6]. We experiment with two strategies. For the first strategy, we establish the senses incrementally while building the network (cf. Figure 2(a)), using the information already in the network to decide on the mapping or separate addition of a new instance. This is in line with our previous update algorithm for the integration of Named Entities [3]. The approach can be made semi-supervised by starting off with a network based on a sense annotated corpus such as SemCor [7]. For the second strategy, we first do not disambiguate while building, simply merging every node with the same label. We then use the resulting context subclusters around a polysemous word to induce its senses (cf. Figure 2(b)). We can then do a second building run using the established clusters to discriminate occurrences before adding them to the network. Related work on co-occurrence graphs built from target-word-specific paragraph collections has been successful [8].

Our second aim is to demonstrate the usability of the resulting network for hierarchical and non-hierarchical clustering of terms (overview eg. [9]). We build a vector representation for a target word from its context in the network and evaluate our result against vector representations derived from other, well-known types of context such as co-occurrence in a paragraph or syntactic slots. The length of the vector is potentially equal to the total number of object nodes, with appropriate pruning. We use spreading activation to retrieve the values, corresponding to the amount of activation each node receives when the target node is fired. This measure reflects the semantic relatedness of each node to the target node, leveraging the rich network structure and thus creating a more robust vector representation.

3 RDF Serialization

In order to exploit semantic technologies, we implemented an RDF serialization of our graph model. Thereby, RDF triple stores can be used for storage of large-scale networks; querying via SPARQL allows for retrieval of graph patterns as well as on-the-fly creation of new networks; RDF-compatible graph-drawing tools greatly facilitate to visualize and explore the networks. Beyond that, using RDF also enables interoperability on the resource level: As data from various domains becomes publicly available as Linked Data in the RDF format, external resources (e.g. lexical, encyclopedic, or ontological) can be easily accessed and integrated with graph models created by our approach, enabling intense usage of background knowledge and countering potential problems with graph sparseness.

4 Conclusion

We have presented an approach for building large-scale semantic networks automatically from text, employing deep semantic processing. Our graph model provides a well-balanced middle ground between purely symbolic and numerical approaches to graph-based knowledge representation. We have identified several ways in which our semantic network models can be used for conceptual knowledge acquisition. Our implementation of the building algorithm is highly competitive in terms of coverage and performance. Future work includes a rigorous evaluation in order to investigate the added value of our approach compared to other graph representations generated from text such as co-occurrence graphs, resources such as ConceptNet and WordNet as well as task specific non graph-based methods.

References

1. Sowa, J.: Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley, Reading (1984)
2. Manola, F., Milner, E.: RDF Primer. W3C Recommendation (February 10, 2004), tt www w o t
3. Harrington, B., Clark, S.: Asknet: automated semantic knowledge network. In: Proc. 22nd National Conf. on Artificial intelligence (AAAI 2007), pp. 889–894. AAAI Press, Menlo Park (2007)
4. Wojtinnek, P.R., Harrington, B., Rudolph, S., Pulman, S.: Conceptual knowledge acquisition using automatically generated large-scale semantic networks. Technical report, Oxford University Computing Laboratory (April 2010)
5. Curran, J., Clark, S., Bos, J.: Linguistically motivated large-scale NLP with C&C and Boxer. In: Proc. 45th Annual Meeting of the ACL, Demo and Poster Sessions, pp. 33–36. ACL (June 2007)
6. Navigli, R.: Word sense disambiguation: A survey. ACM Comput. Surv. 41(2), 1–69 (2009)
7. Miller, G.A., Leacock, C., Tengi, R., Bunker, R.T.: A semantic concordance. In: HLT 1993, pp. 303–308. ACL (1993)
8. Agirre, E., Soroa, A.: UBC-AS: a graph based unsupervised system for induction and classification. In: SemEval 2007, pp. 346–349. ACL (2007)
9. Biemann, C.: Ontology learning from text: A survey of methods. LDV Forum 20(2), 75–93 (2005)

Author Index