Genifer
– an artificial general intelligence

YKY (general.intelligence@Gmail.com)

with input from:
Abram Demski
Ben Goertzel
Martin Magnusson
William Taysom
Russell Wallace
Pei Wang

Preface, executive summary, to-do list

1. This book is a perpetual draft.
2. My personal reason for developing AGI is to achieve life extension.
3. The source code of Genifer is hosted on Google Code, including some very easy tutorial slides. Also feel free to contact me!

— YKY

Executive summary:

Inference: Genifer descended from classical logic-based A.I. Its 3 modes of inference are deduction, abduction (explaining), and induction (learning). This is common to NARS, OpenCog, Cyc.

Logic: Genifer is based on an **algebra of concept composition**, which replaces predicate logic as the internal structure of propositions.

KB: Genifer's KB stores logic formulas, similar to classical A.I. systems such as Cyc, and NARS. OpenCog is an exception in that it stores its knowledge as a hypergraph called AtomSpace.

Uncertainty: Genifer uses fuzzy-probabilistic logic, the probabilistic part is an exact algorithm for belief propagation in Bayesian networks. The fuzzy-probabilistic calculus is created by YKY based on the Beta distribution.

Bootstrapping: Genifer will be written in its own language, which is a **logical-functional** programming language based on Genifer's logic and an existing functional programming language such as Clojure or Haskell.

To-do:

Ch 1 (Introduction) Explain the new ideas that I learned about the relationship between propositional logic and topological logic.

Ch 2 (Architecture) Explain AIXI, algorithmic complexity, Solomonoff induction, etc. Explain distributive architecture. New idea that bootstrapping is possible.

Ch 3 (KR) — ok —

Ch 4 (Logic) New logic of concept composition. Ideas about equational unification and concepts. Explain background notions, eg paradoxes.

Ch 5 (Z) Add new idea on the "Java-girl paradox", which is in draft paper.

Ch 8 (Inference) Copy and paste Bayesian inference and factor graph stuff from the Lisp code to here.

Ch 9 (Pattern recognition) Matrix technique on similarity.

Ch 11 (Learning) A lot of new material is in the slides.

Ch 12 (NL) New idea of semantic parsing. New diagrams from GUI.

Ch 13 (Memory) Explain hierarchical clustering idea, ontology.

Ch 14 (Planning) May need re-think.

Ch 18 (Implementation) Bootstrap Genifer in its own language.

Appendix A Recommend more books for AGI sub-areas. Especially math books.

Contents

0 Introduction

0.1 Chicken-and-egg problem

Part I

Techniques

1 Machine learning basics

1.1 Inductive bias and "no free lunch" theorem

1.2 Structuralism, post-structuralism

2 Logic

"The only way to rectify our reasonings is to make them as tangible as those of the Mathematicians, so that we can find our error at a glance, and when there are disputes among persons, we can simply say: Let us calculate [calculemus], without further ado, to see who is right."

— Leibniz

2.1 The 3 main modes of human thinking

2.1.1 Deduction

2.1.2 Abduction

2.1.3 Induction

2.2 Architecture of logic-based AI system

2.3 Propositional logic

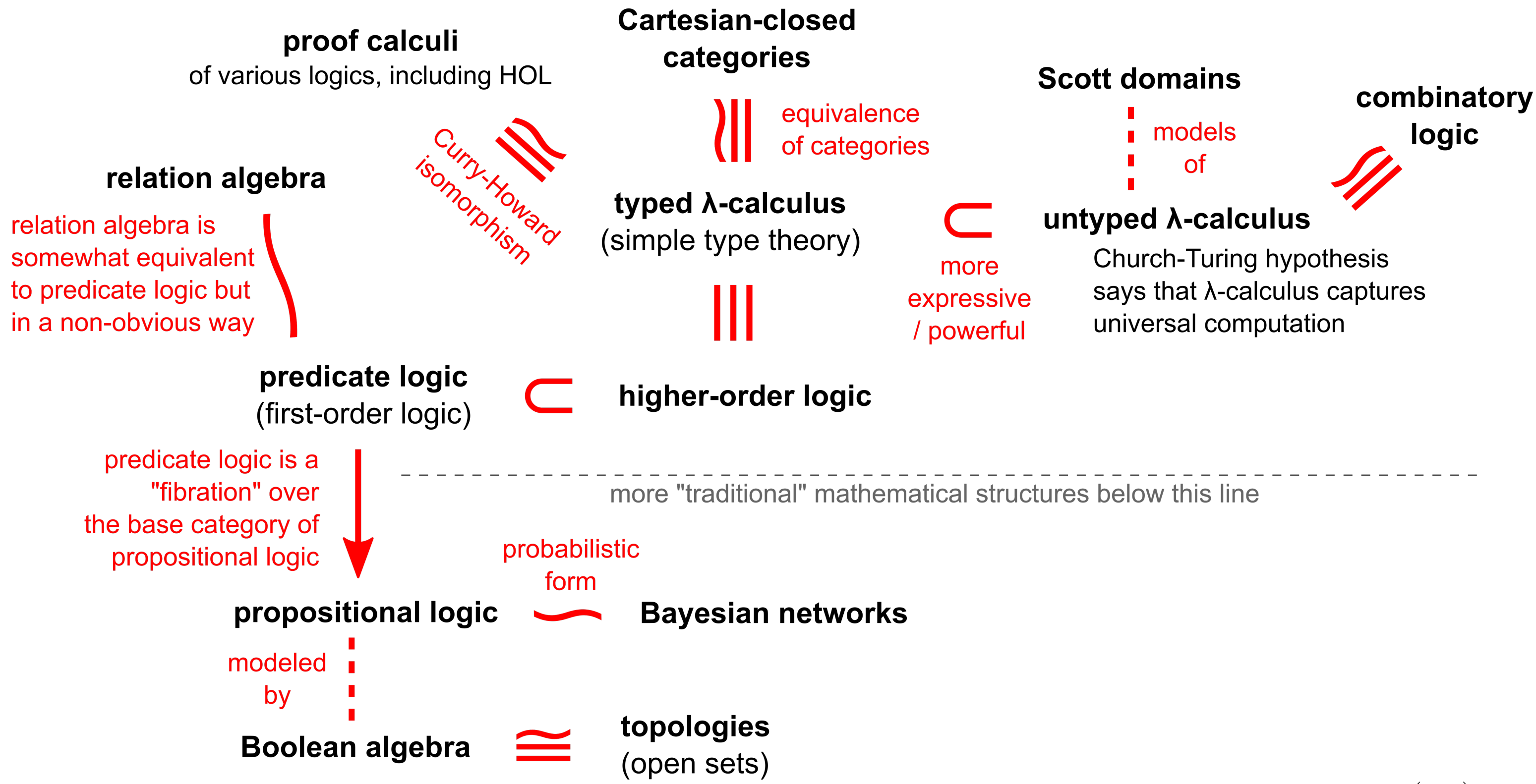
2.4 Predicate logic / first-order logic

2.5 Inference (classical)

2.5.1 Resolution algorithm

2.5.2 Unification algorithm

2.6 Relations between various logical structures



(2.1)

2.7 Second-order / higher-order logic

2.8 λ-calculus, combinatory logic

2.9 Curry-Howard isomorphism

2.10 Model theory

2.10.1 functorial semantics

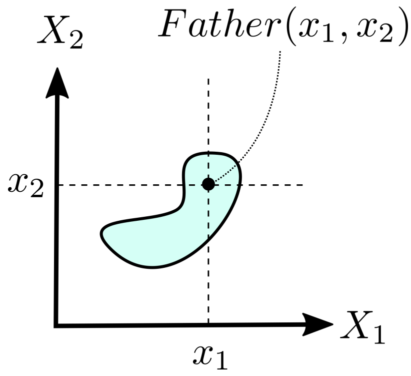
2.11 Algebraic logic, geometrization

(algebraization) Alfred Tarski **cylindric algebra** Paul Halmos **polyadic algebra** **relation algebra**

2.11.1 Cylindric algebra

Cylindric algebra Tarski first-order logic

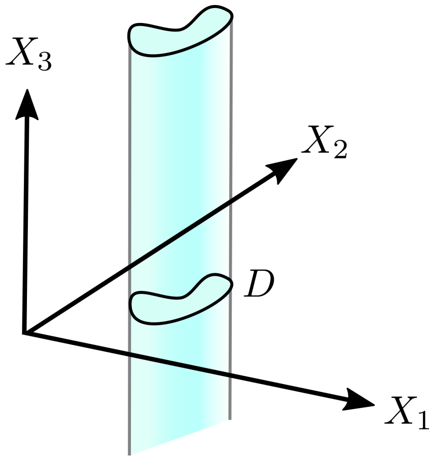
$x_1 R\, x_2$ $R(x_1, x_2)$ $X_1 \times X_2$ domain Cartesian



(2.2)

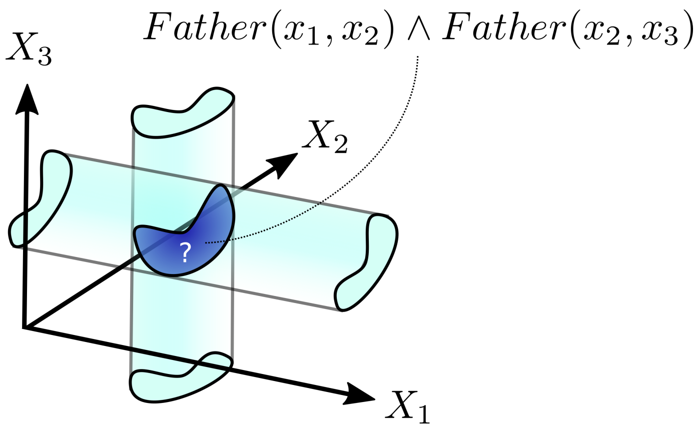
cylinders domains Cartesian

$Father(x_1, x_2)$ $X_1 \times X_2$ D x_3 “don’t care” domain X_3 $D \times X_3$



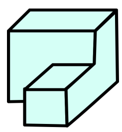
(2.3)

composition $R_1 \circ R_2$ cylinders intersection



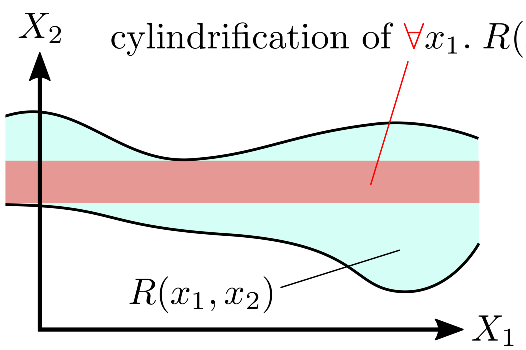
(2.4)

intersection L cylinders intersect



(2.5)

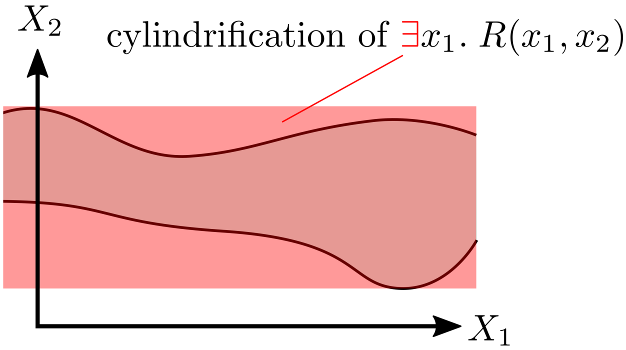
$\forall$ cylindrification



(2.6)

∀ cylindrification

$$ {}_{\forall} = \{(x_1, x_2)|\ \forall \hat{x}_1.\ R(\hat{x}_1, x_2)\} \tag{2.7} $$

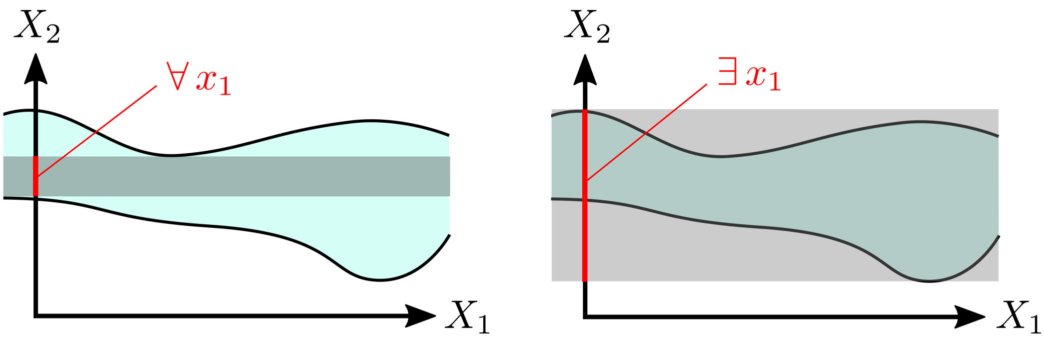


(2.8)

∃ cylindrification

$$ {}_{\exists} = \{(x_1, x_2)|\ \exists \hat{x}_1.\ R(\hat{x}_1, x_2)\} \tag{2.9} $$

∀ ∃ X_2



(2.10)

$$ \pi_1(x_1, x_2) = x_1, \qquad \pi_2(x_1, x_2) = x_2 \tag{2.11} $$

$$ \begin{aligned} {}_{\exists} &= \{x_2|\ \exists \vec{x}.\ [x_2 = \pi_2(\vec{x})\ \wedge\ \vec{x} \in R]\} \\ {}_{\forall} &= \{x_2|\ \forall \vec{x}.\ [x_2 = \pi_2(\vec{x})\ \rightarrow\ \vec{x} \in R]\} \end{aligned} \tag{2.12} $$

$\rightarrow$ $\wedge$

F William Lawvere π_i generalize $f : X \rightarrow Y$

$$ \begin{aligned} {}_{\exists} &= \{y|\ \exists x.\ [y = f(x)\ \wedge\ x \in R]\} \\ {}_{\forall} &= \{y|\ \forall x.\ [y = f(x)\ \rightarrow\ x \in R]\} \end{aligned} \tag{2.13} $$

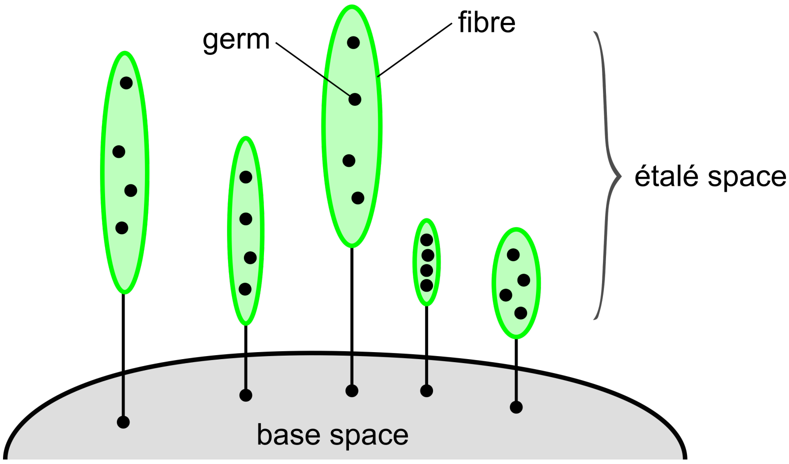
X Y $f : X \rightarrow Y$ f "substitution map"

∀ ∃ substitution map f co-variant **adjoints**

2.12 Category theory, categorical logic

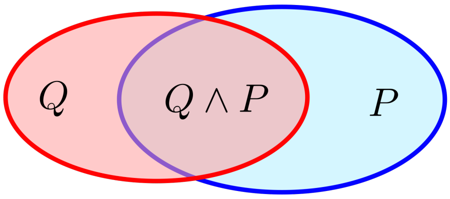
2.12.1 fibration

Fibration

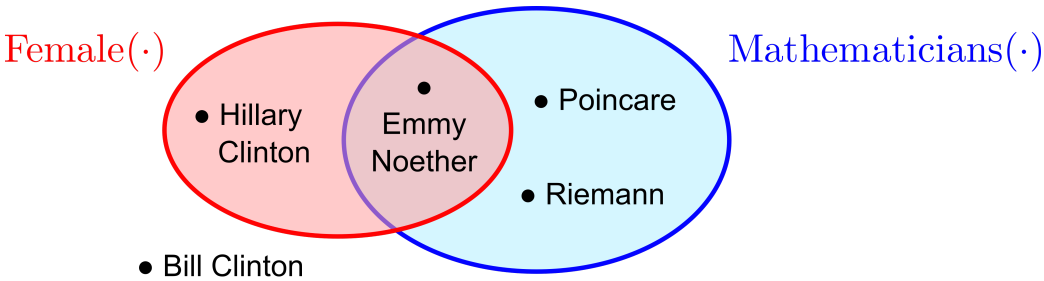


(2.14)

Q = "Mary is female" $\quad$ P = "Mary is a mathematician"

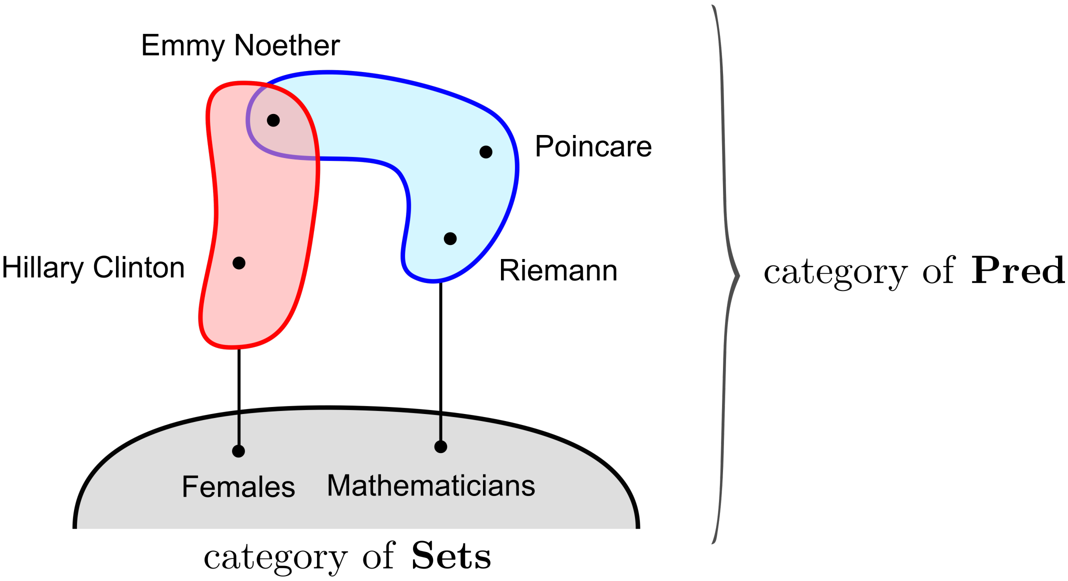


(2.15)



(2.16)

fibration



(2.17)

2.12.2 Lawvere quantification

[?], [?]

2.13 Quantum logic

2.14 Term rewriting systems

2.15 Graph rewriting systems, hypergraphs

Bibliography

Lawvere. *Functorial semantics of algebraic theories.* PhD thesis, Columbia university, 1963.

Lawvere and Rosebrugh. *Sets for mathematics.* Cambridge, 2003.

3 Uncertainty

3.1 Fuzziness

3.2 Probability

3.2.1 Bayesian networks

3.3 Confidence

3.4 Uncertain inference

3.4.1 MCMC (Markov chain Monte Carlo)

4 Neural networks

4.1 Neuroscience

4.1.1 Brain architecture

4.1.2 Neurons

4.1.3 Neuro-chemistry

4.2 Mathematics of neural networks

4.2.1 Non-linear analysis

4.2.2 Degree theory

4.2.3 Homology

4.2.4 Harmonic analysis

4.3 Deep learning

5 Evolution

5.1 History of natural evolution

5.2 Spectrum of the evolution operator

6 Reinforcement learning

6.1 Control theory / differential geometry

6.2 Optimization

Part II

Faculties

7 Pattern recognition

7.1 Vision

8 Belief revision / truth maintenance

9 Inductive learning

9.1 Logic-based inductive learning

10 Natural language

10.1 Syntax theory

10.2 Semantic theory

10.2.1 Abduction as interpretation

10.2.2 Montague grammar

10.2.3 Categorial grammar

11 Planning

11.1 Program synthesis

Part III

Architecture

12 Cognitive architectures

13 Memory systems

13.1 Working memory

13.2 Episodic memory

14 Implementation

14.1 Ethical issues

14.2 Business aspects

Symbols

$\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}$	classical number systems	
$\mathbb{Hyp}$	hypothesis space	§??
$\mathbb{Prop}$	(ground) proposition space	

General logic:

$\exists, \forall$	classical existential and universal quantifiers	
$\wedge, \vee, \neg$	classical binary logic AND, OR, NOT	
$\rightarrow$	(classical) implication	§??
$\vdash$	entailment, syntactic	
$\models$	entailment, semantic	
$=$	equality (logic predicate)	§??
$\approx$	similarity = fuzzy equality (logic predicate)	§??
$\subseteq$	inclusion ("is-a" relation)	§??
$\sim$	association (logic predicate)	
$a \circ b$	composition of concepts	§??
(a, b)	pairing or union	§??
$\lambda x.Mx$	lambda abstraction	
$M : \tau$	(type theory) expression M is of type τ	
$t \overset{R}{\Rightarrow} t'$	t rewrites to t' under rewriting system R	
$t \overset{R}{\leadsto} t'$	t narrows to t' under rewriting system R	§??
$A \bowtie B$	unify(A,B)	§??
[s_1]: formula	KB stores statement s_1	

Fuzzy and probabilistic logic:

$\#x.Q(x)$	probabilistic quantifier ("for some")	§??
$\rightarrowtriangle$	probabilistic implication (= Bayesian network link)	§??
$\overset{Z}{\wedge}, \overset{Z}{\vee}$	fuzzy AND and OR	§??
$\overset{P}{\wedge}, \overset{P}{\vee}$	probabilistic AND and OR	§??
$\odot$	a (fuzzy or probabilistic) operator that combines AND and OR	§??
$\Gamma(\cdot)$	fuzzy modifier	§??
ξ	point of neutrality (fuzzy logic)	§??
w	total number of support for a hypothesis	§??
w^+, w^-	positive and negative support for a hypothesis	§??

Categories of truth values:

$\mathcal{B}$	binary logic
$\mathcal{P}$	(binary) probabilistic logic
$\mathcal{Z}$	pure fuzzy logic
$\mathcal{P}(\mathcal{B})$	binary-probabilistic logic
$\mathcal{P}(\mathcal{Z})$	fuzzy-probabilistic logic

Miscellaneous:

"text"	texts in English / natural language
`source code`	source code
formula	logic formulas
To do: ...	things to do

Acknowledgements

In addition to the people listed on the title page, I'd like to thank the AGI mailing-list participants for years of discussions.