



# More on first-order neural-symbolic integration and some further wrap-up

**Sebastian Bader**, ICCL, TU Dresden, Germany  
**Pascal Hitzler**, AIFB, University of Karlsruhe, Germany

ESLLI2008 – Friday Session



## The course so far

- Introduction and Motivation
- The Core Method for Propositional Logic
- Applications of the Propositional Core Method
- A New Approach to Pedagogical Extraction
- The Core Method for First-Order Logic
- More on First-Order & other Perspectives



## Contents (This part)

- **Why are we doing all this?**
- Some other first-order neural-symbolic approaches.
- Where are we going?



# Why are we doing all this?

Three motivations?

1. getting the best of both worlds for AI applications  
(Computer Science motivation)
2. understanding cognitive processes – AI in the original sense  
(Cognitive Science motivation)
3. modelling and understanding the human brain  
(Neuroscience motivation)



# Computer Science motivation

- combining robust (neural network) learning with declarative (logical) understanding and reasoning
- we talked about this already in the intro
- and we indeed made steps in this direction via realising the neural-symbolic cycle



# Why are we doing all this?

Three motivations?

1. getting the best of both worlds for AI applications  
(Computer Science motivation)
2. understanding cognitive processes – AI in the original sense  
**(Cognitive Science motivation)**
3. modelling and understanding the human brain  
(Neuroscience motivation)



# Cognitive Science motivation

- understanding and modelling of cognitive processes
- It is rather uncontroversial, that reasoning and learning are crucial for every cognitive system.
- Betty the Crow



# Cognitive Science motivation

- Reasoning and learning in cognitive systems face several challenges:
  - Reasoning and learning belong to different areas:
    - **Reasoning is classically considered as the problem to infer knowledge from given knowledge.**
    - **Learning is about establishing generalizations from facts, observations etc.**
  - Reasoning is based on logic theories, i.e. symbolic approaches
  - Learning is often based on subsymbolic / connectionist theories

## Symbolic vs. subsymbolic

There is an obvious tension between symbolic and subsymbolic representations.

	Symbolic Approaches	Subsymbolic Approaches
<b>Methods</b>	Mainly logical and / or algebraic	Mainly analytic
<b>Strengths</b>	Productivity, recursion, compositionality, declarativeness	Robustness, learning, parsimony, adaptivity
<b>Weaknesses</b>	lower cognitive abilities	Opaqueness, higher cognitive abilities
<b>Applications</b>	Reasoning, problem solving, planning etc.	Learning, motor control, vision etc.
<b>Relation to Neurobiology</b>	Not biologically inspired	Biologically inspired
<b>Other Features</b>	Crisp	Fuzzy



# Cognitive architectures

- Some cognitive architectures:
  - SOAR (Laird & Rosenbloom)
  - ACT-R (Anderson & Lebiere)
  - Clarion (Sun)
  - Micro-Psi (Dörner & Bach)
  - AMBR / DUAL (Kokinov)
- These architectures were developed for reasoning tasks and most of them implement also learning strategies.
- An interesting fact is that they contain (more or less) neural-symbolic representations.
  - Some contained neural-symbolic devices already in the original proposal (Clarion, AMBR, Micro-Psi).
  - Some were very recently partially extended into a subsymbolic direction (SOAR, ACT-R).



# Cognitively Motivated Constraints

- Gap between symbolic and subsymbolic representations.
  - From a cognitive perspective, there is a gap between subsymbolic signal input, signal / information processing, and the emergence of conceptual knowledge (and vice versa).
  - Currently there are no good ideas to explain this gap.
- Need for dynamic representations:
  - Some researchers argue for dynamic representations.
  - It is hard to explain contextually embedded dynamically generated representations with purely symbolic means.



# Cognitively Motivated Constraints

The role of models (situations):

- Symbolic representations tend to be not very well-suited for the fuzziness and context dependency of cognitive abilities.
- A number of experiments seem to support the claim that human reasoning is model- or situation-based.
- Symbolic frameworks are usually syntactic and not model-based.



# Cognitively Motivated Constraints

- There are neuropsychological findings that memory is constantly reorganized and rearranged in natural agents.
  - For symbolic systems constantly reorganizing memory systems are hard to model.
- Generalized intelligence (intelligence on a human scale) requires a variety of reasoning and learning paradigms.
  - In order to reduce the number of different frameworks for special problems, it may be reasonable to test neural-symbolic devices instead of pure symbol-based approaches.



# Cognitively Motivated Constraints

- Hypothesis:
  - **Neural-symbolic systems could be a first step for a theory of model-based reasoning.**



# Why are we doing all this?

Three motivations?

1. getting the best of both worlds for AI applications  
(Computer Science motivation)
2. understanding cognitive processes – AI in the original sense  
(Cognitive Science motivation)
3. modelling and understanding the human brain  
(**Neuroscience motivation**)



## Neuroscience motivation

- Striving to understand the (human) brain for various reasons (technological, medical, philosophical, ...)
- artificial neural networks are abstractions of biological neural systems
- Can neural-symbolic integration help to understand the (human) brain?



## Neuroscience motivation

- Can neural-symbolic integration help to understand the (human) brain?

I don't think so.

At least not yet.



# Neuroscience motivation

But!

Tianming Yang and Michael N. Shadlen: Probabilistic reasoning by neurons. *nature* 447, 28 June 2007, 1075-1080

- Experiment on rhesus monkeys.
- They were to learn a task which involved internalising a probabilistic stimulus.
- Neurons in the parietal cortex revealed the addition and subtraction of probabilistic quantities that underlie decision-making on this task.

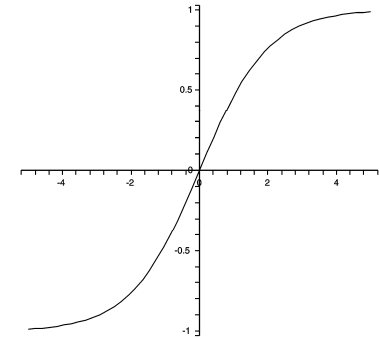


## Contents (This Part)

- Why are we doing all this?
- **Some other first-order neural-symbolic approaches.**
  - **Core Method revisited**
  - Propositional approximation using the Core Method
  - SHRUTI
  - Using Topos Theory
- Where are we going?

## Funahashi 1989 (simplified)

- $\sigma$  sigmoidal activation function
- $K \subseteq \mathbb{R}$  compact
- $f: K \rightarrow \mathbb{R}$  continuous
- $\varepsilon > 0$

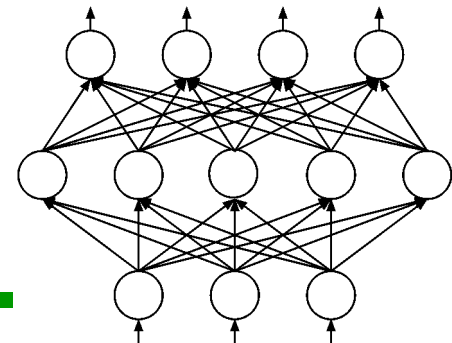


Then there exists a three-layer feedforward network with activation function  $\sigma$  and I/O-function  $F$ , so that

$$\max_{x \in K} \{d(f(x), F(x))\} < \varepsilon.$$

Here  $d$  is a metric which induces the natural topology on  $\mathbb{R}$ .

I.e. continuous functions can be *uniformly approximated* by such networks with arbitrary accuracy.



## Continuity of $T_P - I$

Hitzler, Hölldobler, Seda 2004

Let  $\mathcal{B}_A$  be the set of all body atoms in ground instantiated clauses of  $P$  with head  $A$ .

$T_P: I_P \rightarrow I_P$  is called *locally finite*, if  
 for all atoms  $A$  and all  $I \in I_P$   
 there exists a finite  $S \subseteq \mathcal{B}_A$ ,  
 such that  $T_P(J)(A) = T_P(I)(A)$   
 for all  $J \in I_P$  which coincide with  $I$  on  $S$ .

$$p(s(x)) \leftarrow p(x).$$

$$p(0)$$

$$p(x) \leftarrow p(s(x)).$$

$$\text{e.g. } \mathcal{B}_{p(s(0))} = \{p(0), p(s(s(0)))\}$$

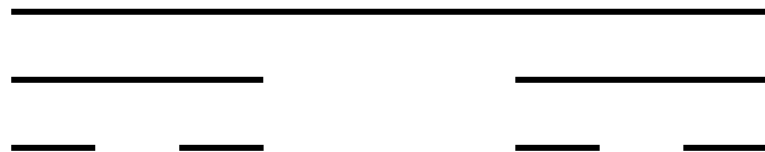
## Continuity of $T_P$ – II

$T_P: I_P \rightarrow I_P$  is locally finite

iff

$T_P$  is continuous in Cantor space.

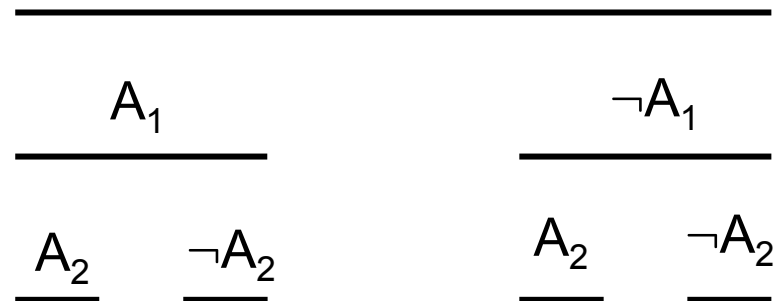
- Cantor-continuity is continuity wrt. the Cantor topology on the Cantor set.
- The Cantor topology is homeomorphic to the prefix-distance on (infinite) binary trees.
- The Cantor topology is homeomorphic to the subspace topology which is induced on a subset of  $\mathbb{R}$  which is compact, totally disconnected and dense in itself.



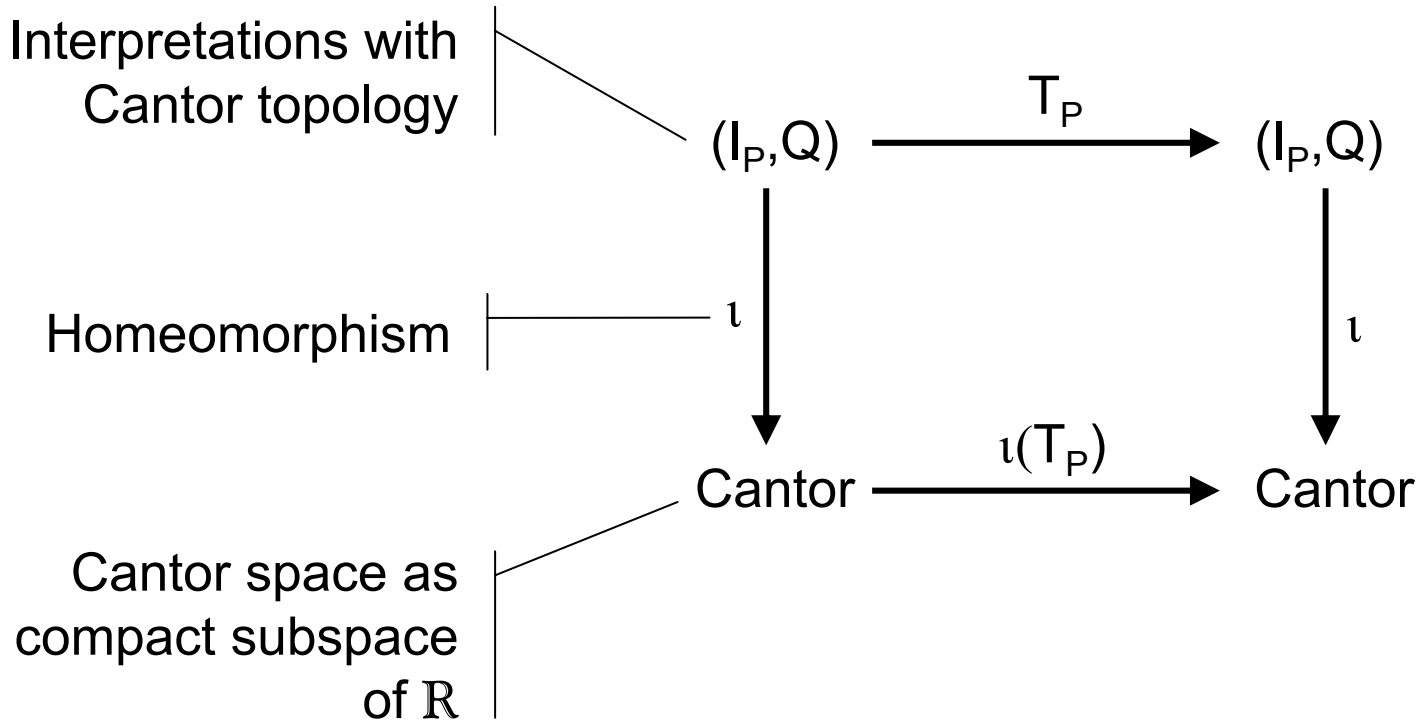
## Continuity of $T_P$ – III

- There are (uncountably) many homeomorphisms which map  $I_P$  with the Cantor topology into suitable subsets of  $\mathbb{R}$ .
- Locally finiteness is a logical (topology-free) characterisation of logic programs which can be represented in a connectionist way in the sense of Funahashi.
- Problem: this argumentation is not constructive!

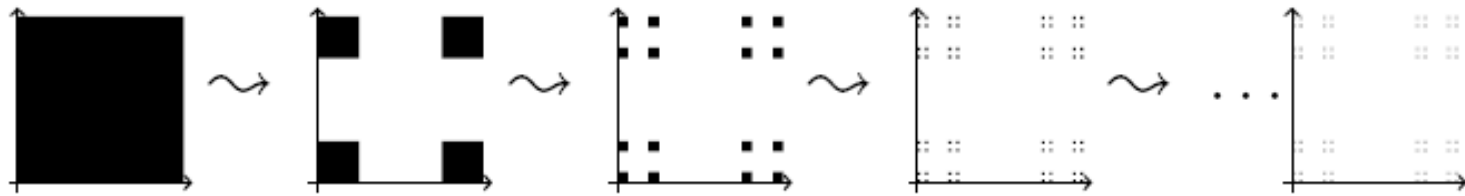
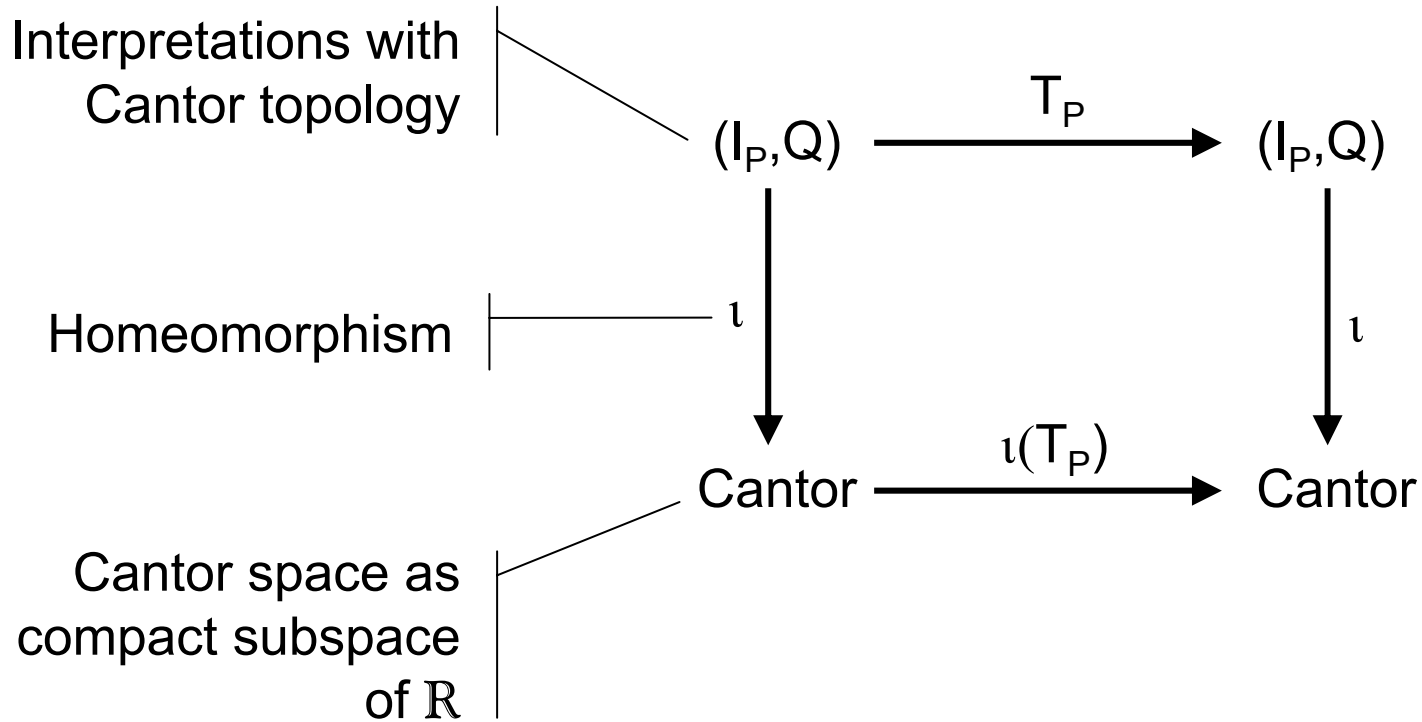
$A_1, A_2, \dots$  enumeration of  
Herbrand base  
Elements of Cantor Set  
identifiable with  
interpretations



# Relationship of $I_P$ to Cantor Space



# Relationship of $I_P$ to Cantor Space





# The Cantor topology as a paradigm bridge

- Connectionist side:
  - Cantor topology is a subtopology of the usual topology on the real numbers
- Logic Programming side:
  - Cantor topology captures useful notions of convergence of semantic operators, e.g.  
If  $T_P^n(K) \rightarrow I$  (for  $n \rightarrow \infty$  and some  $K$ ), then  $I$  is a model of  $P$ .



## Neuroscience motivation

- Can neural-symbolic integration help to understand the (human) brain?

I don't think so.

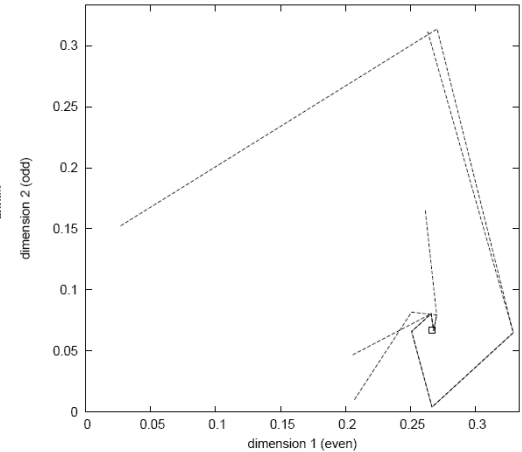
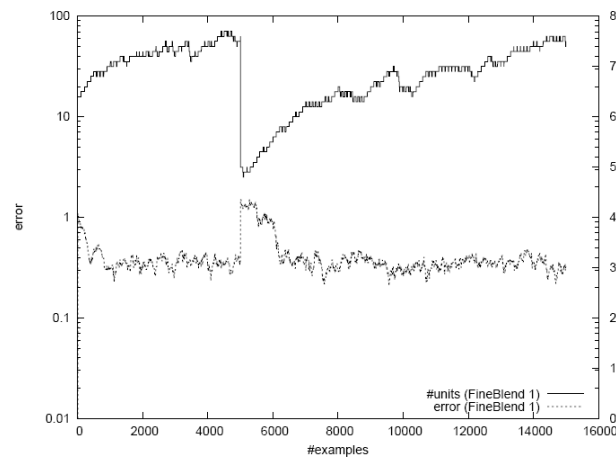
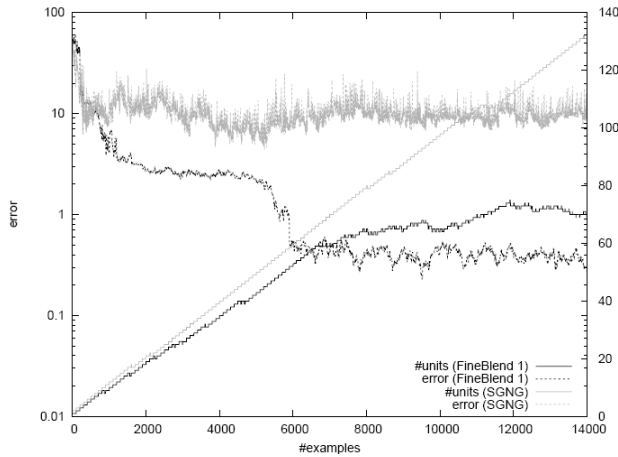
At least not yet.



## Bird's eyes perspective

- We use a structural representation based on a mathematical homeomorphism, which is not really feasible from a cognitive science perspective.
- We do not really represent the knowledge.
- We rather represent a model generator.
  
- I.e. in a sense, we represent a model!

# Reminder: That FOL Core achieves

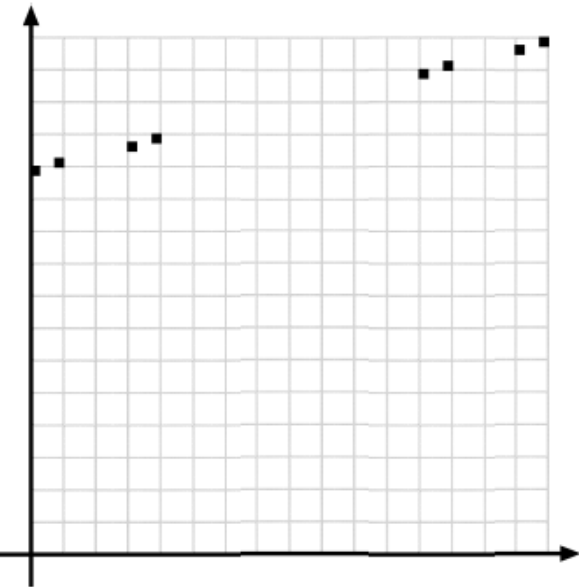


- learns first-order knowledge
- represents it robustly
- can generate the model (i.e. do reasoning)

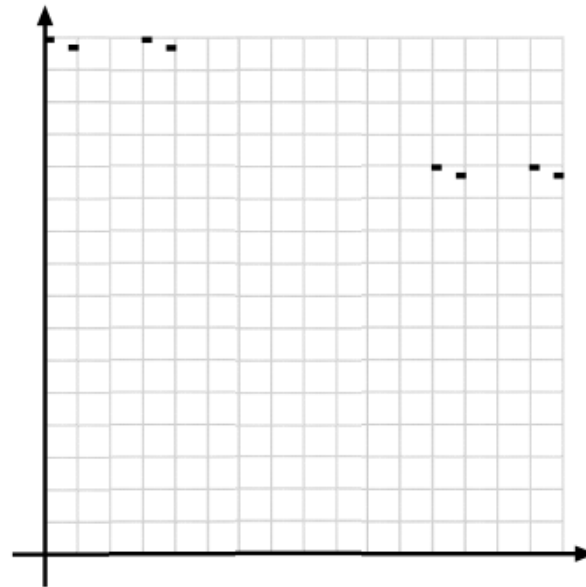
best of both worlds!

## A side remark on fractals, chaos, and dynamical systems

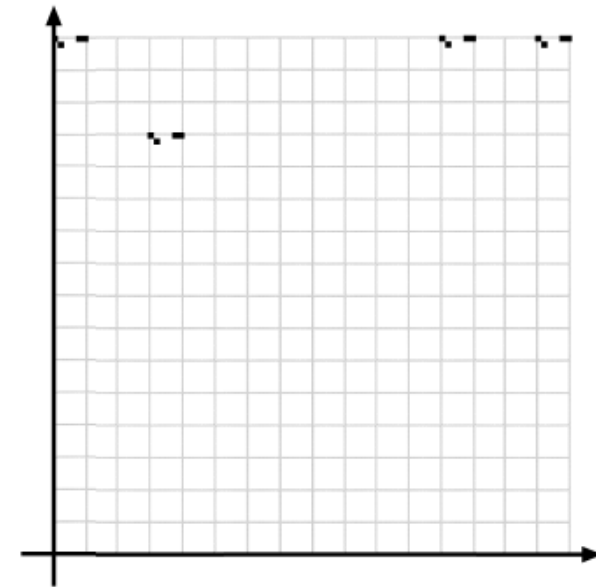
- The embedded graphs of logic programs appear to be self-similar.



```
n(0).
n(s(X)) ← n(X).
```



```
e(0).
e(s(X)) ← not e(X).
o(X) ← not e(X).
```



```
p(0).
p(s(X)) ← p(X).
p(X) ← not p(X).
```



## A side remark on fractals, chaos, and dynamical systems

- The embedded graphs of logic programs appear to be self-similar.
- I.e. they appear to be fractals.
- This can be formally verified (Bader & Hitzler 2004): They are attractors of so-called iterated function systems.



## A side remark on fractals, chaos, and dynamical systems

- Other findings support the fractal nature of logic programs (e.g. Blair et al) and of (recurrent) neural networks (e.g. Tino et al.)
- This leads us into the realm of dynamical systems and chaos theory.
- We currently do not have the mathematical means to really make sense of chaotic dynamical systems ...
  - perhaps such a breakthrough is necessary in order to advance substantially in cognitive science
  - ...



## Contents (This Part)

- Why are we doing all this?
- Some other first-order neural-symbolic approaches.
  - Core Method revisited
  - **Propositional approximation using the Core Method**
  - SHRUTI
  - Using Topos Theory
- Where are we going?



## Propositional First-Order approximation

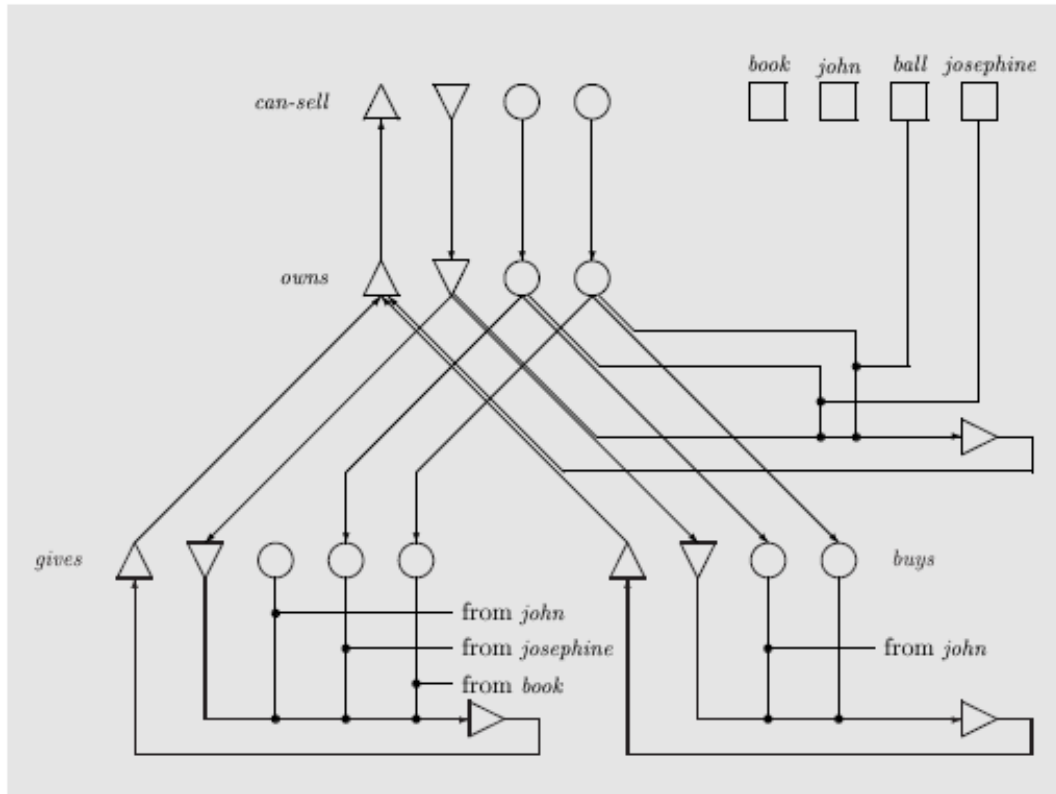
- Work by Seda et al.
- Idea:
  - use propositional core method.
  - given a first-order logic program, take its grounding (i.e. all possible variable substitutions by ground terms)
  - take a finite subprogram of this ground program, which is essentially propositional, as approximation
- It hasn't been developed (yet) into an experimental setting.



## Contents (This Part)

- Why are we doing all this?
- Some other first-order neural-symbolic approaches.
  - Core Method revisited
  - Propositional approximation using the Core Method
  - **SHRUTI**
  - Using Topos Theory
- Where are we going?

# SHRUTI



Shastri & Ajjanagadde 1993

Variable binding  
via time synchronization.

*Reflexive* (i.e. fast)  
*reasoning* possible.

Picture: Hölldobler,  
*Introduction to  
Computational Logic*, 2001

$\text{gives}(X, Y, Z) \rightarrow \text{owns}(Y, Z)$   
 $\text{buys}(X, Y) \rightarrow \text{owns}(X, Y)$   
 $\text{owns}(X, Y) \rightarrow \text{can-sell}(X, Y)$

$\text{gives}(\text{john}, \text{josephine}, \text{book})$   
 $(\exists X) \text{buys}(\text{john}, X)$   
 $\text{owns}(\text{josephine}, \text{ball})$



# SHRUTI

- Work by Shastri and Ajanagadde
- connectionist representation of deduction process, not of a model
- learning appears to be very difficult
- how to further develop this approach fruitfully
  - for either of the motivational perspectives?



## Contents (This Part)

- Why are we doing all this?
- Some other first-order neural-symbolic approaches.
  - Core Method revisited
  - Propositional approximation using the Core Method
  - SHRUTI
  - **Using Topos Theory**
- Where are we going?

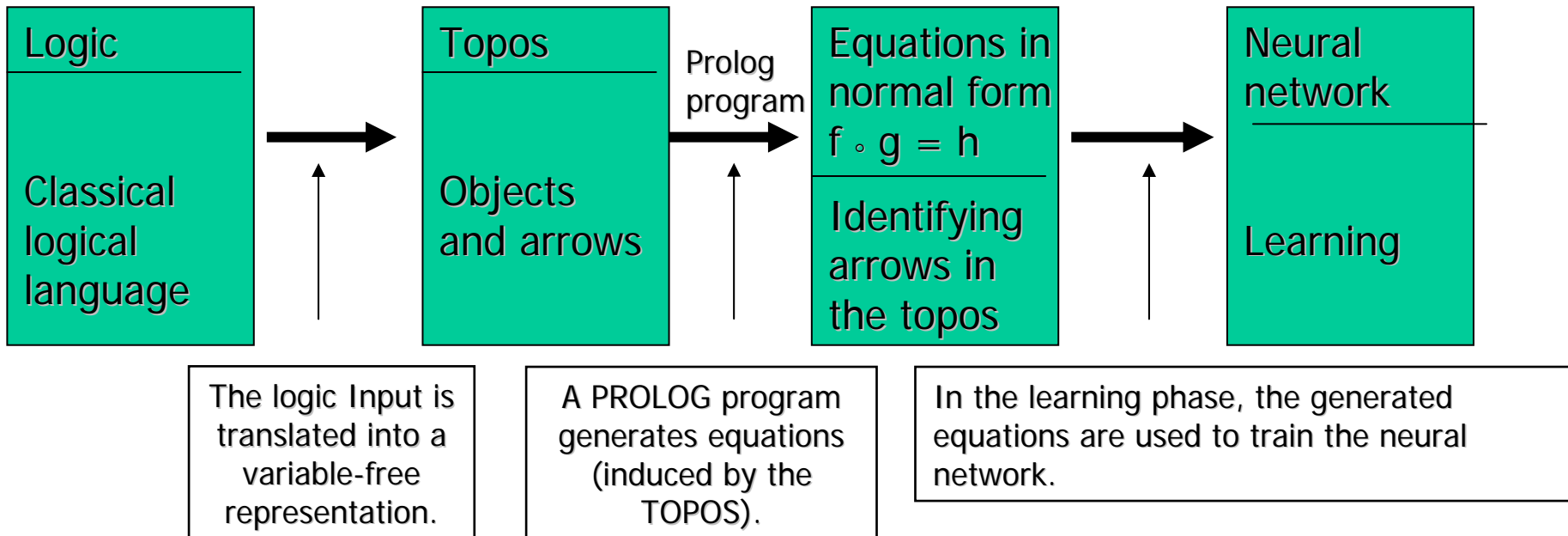


# Using Topos Theory

- Work by Gust & Kühnberger
- Using an abstract variable-free representation of predicate logical theories based on topoi from category theory.
- Learn connectionist model representation from this topos representation.

# The Training Architecture

The account has the following general architecture





# Inferences

- Logic
  - Models
    - **An interpretation  $I$  is a model of a set of axioms  $\Sigma$  iff  $I(A) = t$  for all  $A \in \Sigma$**
  - Semantic entailment
    - **$\Sigma \models A$  iff  $I(A) = t$  for all models  $I$  of  $\Sigma$**
  - Provability
    - **$\Sigma \vdash A$  iff  $A$  is provable from  $\Sigma$**
  - Soundness and completeness
    - **$\Sigma \vdash A$  iff  $\Sigma \models A$**
- Basic idea
  - Models are functions and functions can be learned by neural networks
  - Task: learn an interpretation function  $I$  such that  $I$  is a model of  $\Sigma$

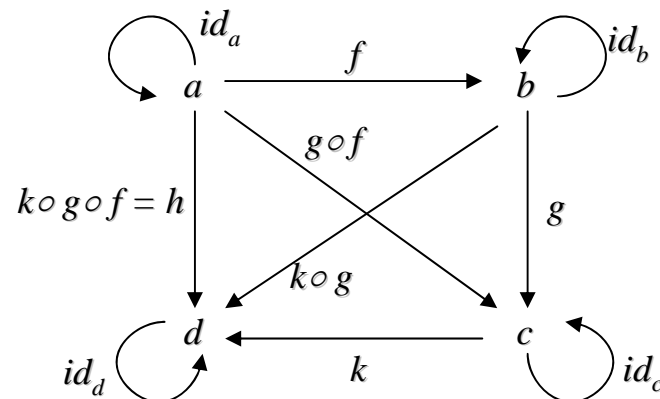


# Strategy for a Solution

- Neural networks can only learn functions of the following type
  - $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$
- Task
  - Try to find a structure  $X$  such that the interpretation function can be written as  $I: X \rightarrow X$
  - Reduce compositional complexity to one binary operation on  $X$  (composition in a category)
- Our proposal for a solution
  - Reification
  - Elimination of variables
  - Mapping logical expressions into  $X$
  - Representing the elements of  $X$  in a real-valued vector space

# The Definition of a Category

- A category  $\mathbf{C}$  is defined as a 6-tuple  $\mathbf{C} = \langle \text{Obj}_{\mathbf{C}}, \text{Ar}_{\mathbf{C}}, \circ_{\mathbf{C}}, \text{dom}, \text{codom}, \text{id} \rangle$ 
  - $\text{Obj}_{\mathbf{C}}$ : class of objects
  - $\text{Ar}_{\mathbf{C}}$ : class of arrows (between objects)
  - $\circ_{\mathbf{C}}$ : concatenation of arrows (associative)
  - $\text{dom}, \text{codom}$ : defined on arrows (domain and codomain)
  - $\text{id}$ : for each object  $b \in \text{Obj}_{\mathbf{C}}$ :  $\text{id}_b: b \rightarrow b$ , such that for each  $f: a \rightarrow b$  and  $g: b \rightarrow c$ , it holds:  $\text{id}_b \circ f = f$  and  $g \circ \text{id}_b = g$





# Topoi

- An example of a topos is the category **SET**
  - Objects are sets, arrows are set-theoretic functions
- A topos is a special type of category
  - It has a final object “!”
    - **For every  $c \in \text{Obj}_c$  there exists a unique  $!_c: c \rightarrow !$**
    - **For sets the one-element set  $\{*\}$  is final**
  - Products
    - **$a \leftarrow a \times b \rightarrow b$  with  $a \times b$  Cartesian product and the corresponding projections**
  - Pullbacks
    - **Is a very general kind of product construction**
  - Exponents
    - **For all  $a \rightarrow c^b$  there is a unique  $a \times b \rightarrow c$**
  - A subobject classifier
    - **true:  $! \rightarrow \Omega$  where  $\Omega$  is a truth value object (in SET a two-element set)**
    - **Intuition: the subobject classifier allows the characterization of subobjects by predicates (or characteristic functions)**
    - **Subsets in SET are representable by characteristic functions**



# Topoi

- Goldblatt (1984) showed that a topos can be used to represent first-order logic
- Features of a topos
  - Only one operation is necessary
    - **Everything is reduced to concatenation of arrows**
    - **Concatenation is based on compositionality**
    - **Therefore: no heterogeneous interpretation function needs to be defined**
  - Variables
    - **Variables are not explicitly represented**
    - **Variables do only occur implicitly**
  - Semantics
    - **Truth of formulas depend on constructions of arrows**



# Variable-Free Logic

- Logic expressions are recursively constructed using terms, predicates, quantifiers, and logical connectives.
  - There exist commuting diagrams for these constructions.
  - Examples:
    - **A constant  $c$  is an arrow  $[c]: ! \rightarrow u$ .**
    - **A predicate  $p$  is an arrow  $[p]: u \times \dots \times u \rightarrow \Omega$**
    - **A conjunction  $\wedge$  is an arrow  $[\wedge]: \Omega \times \Omega \rightarrow \Omega$  such that  $[\wedge] \circ (\text{true} \times \text{true}) = \text{true} \circ !$**
    - **Etc.**
- Notice: the resulting data structure is homogeneous, in the sense that only arrows code information and the only allowed operation is concatenation of arrows.

# Logic and Topos Theory

Logic	Topos
Constant $c$	$[c]: ! \rightarrow u$
Function $f$	$[f]: u \times \dots \times u \rightarrow u$
Term $t$	$[t]: ! \rightarrow u$
Predicate $p$	$[p]: u \times \dots \times u \rightarrow \Omega$
2-ary logical connectives	$\Omega \times \Omega \rightarrow \Omega$
1-ary connective	$\Omega \rightarrow \Omega$
Quantifiers	$\Omega^{X \times Y} \rightarrow \Omega^X$
Closed formula $A$	$[A]: ! \rightarrow \Omega$

Notice: Variables are not explicitly represented in a topos

Formulas containing free variables are represented as (complex) predicates

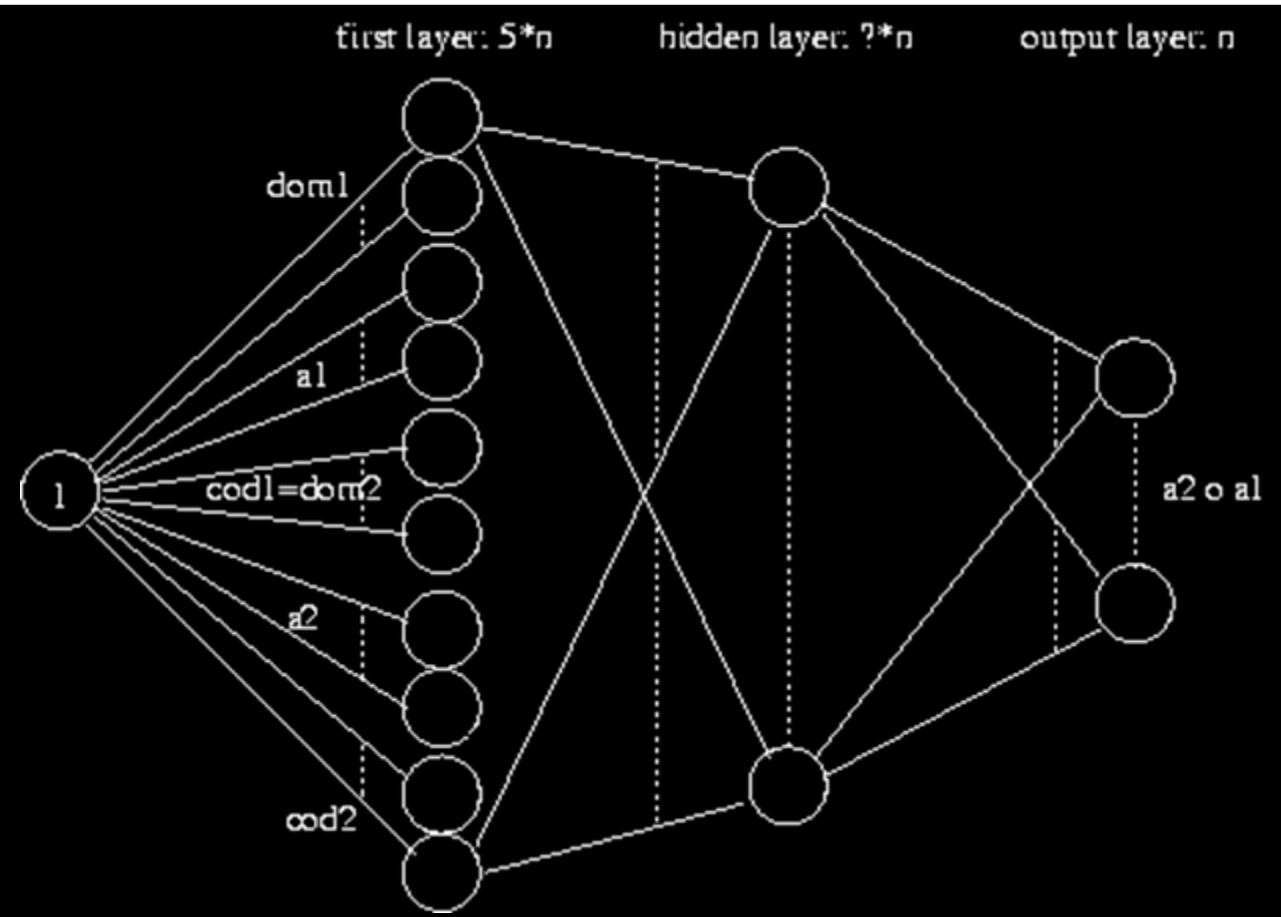
Quantification is therefore an operation mapping predicates to predicates



# Neural Learning

- In order to enable the neural network to learn, some objects and arrows have (fixed) representations:
  - *true*: (1.0,0.0,0.0,0.0,0.0)
  - *false*: (0.0,1.0,0.0,0.0,0.0)
    - **The truth values *true* and *false* are considered to be distinct.**
  - All other objects and arrows are initialized with random values.
- The input of the network represents two arrows:
  - Domain of the first arrow
  - Representation of the first arrow
  - Codomain of the first arrow
  - Representation of the second arrow
  - Codomain of the second arrow
- The network is a feedforward network learning by backpropagation.

# The Network



Objects of the topos are represented as points in an  $n$ -dimensional real-valued unit cube

Each arrow is also represented as a point (together with pointers to the respective domains and codomains)

The input is represented by weights from the initial node with activation 1



# Learning

- The network essentially learns the composition operator. Training examples are taken from the commutative diagrams for the topos.
- From another perspective, a model of the theory is learned. The trained network then correctly recognises true formulas in this model.
- Experiments on an implementation have shown that the approach works in principle, in an approximate fashion.

# Socrates Examples

## Rules:

*All human beings are mortal.*

*All mortal beings ascend to heaven.*

*All beings in heaven are angels.*

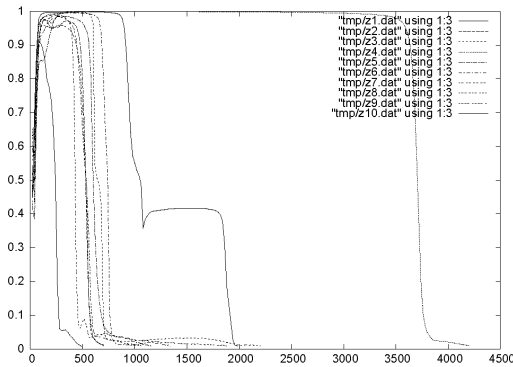
## Facts:

*Socrates is human.*

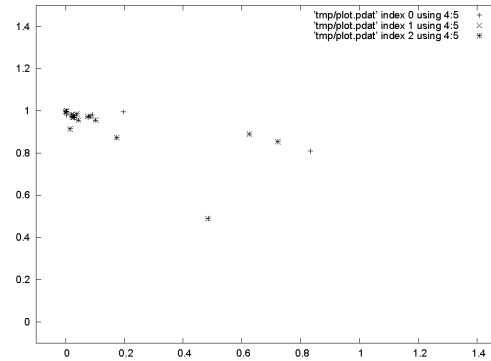
*Robot is not human.*



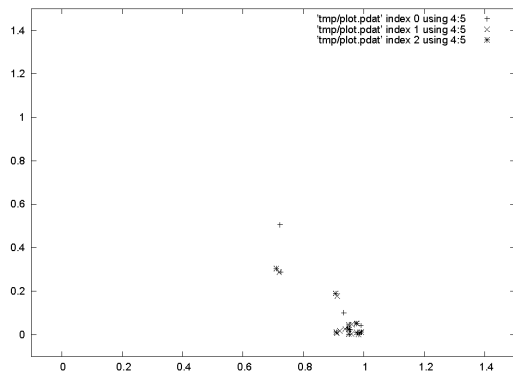
# Application Results



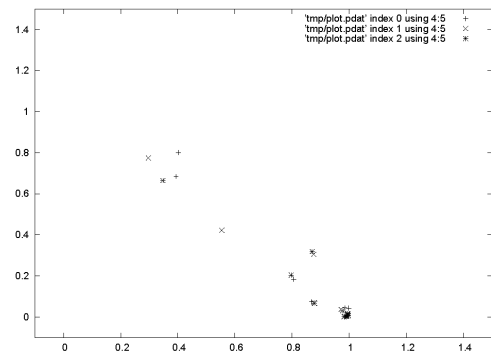
The maximal error of the system  
10 runs with maximal 4250 iterations



Distribution of *Robot* is an *angel*.  
10 runs with maximal 4250 iterations



Distribution of *Socrates* is an *angel*.  
10 runs with maximal 4250 iterations



Distribution of *Something* is an *angel*.  
10 runs with maximal 4250 iterations



## Contents (This Part)

- Why are we doing all this?
- Some other first-order neural-symbolic approaches.
- **Where are we going?**



## Next steps?

- Computer Science side:
  - further integrate learning and reasoning perspectives, e.g. by modifying the learning through background knowledge (e.g. Bader, Hölldobler, Marques 2008).
  - find more suitable logics ... e.g. description logics?
  - find suitable application scenarios, e.g. ontology learning?
- Generally, new creative ideas/approaches are needed to create critical mass



## Next steps?

- Cognitive Science side:
  - make neural-symbolic components part of general cognitive models (see e.g. Gust & Kühnberger, I-Cog)
  - identify test scenarios, i.e. solve cognitive tasks which such systems
- Generally, new creative ideas/approaches are needed to create critical mass



# Thanks

- Special Thanks to Kai-Uwe Kühnberger (University of Osnabrück) for some insights on the cognitive science perspective.
- I shamelessly copied a number of slides from his presentations ...

<http://www.neural-symbolic.org>