Representing first-order knowledge by artificial neural networks

Pascal Hitzler

Artificial Intelligence Institute, Dresden University, Germany

Motivation

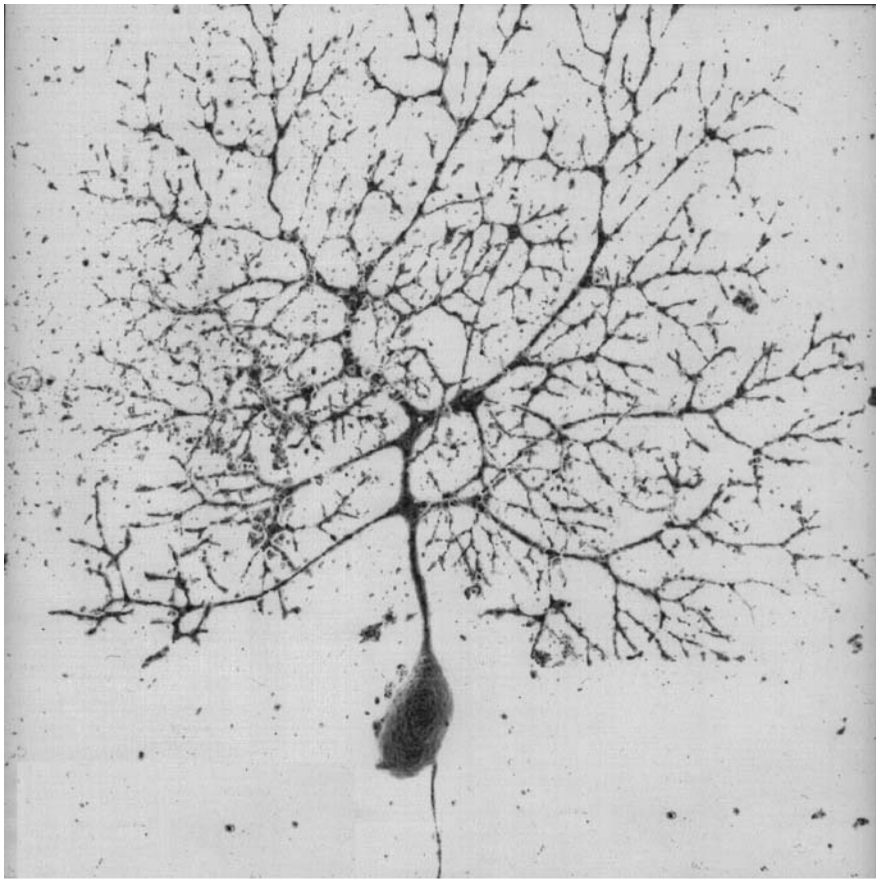
- *Biological* neural networks can easily do logical reasoning.
- Why is it so difficult with *artificial* ones?

Contents

- Brief history on logic and connectionism.
- The first-order language problem.
- Continuity of semantic operators in logic programming.
- Representing logic programs by neural networks.

New results were obtained in partial collaboration with Sebastian **Bader**, Steffen **Hölldobler** (Dresden, Germany), or Anthony K. **Seda** (Cork, Ireland).

Biological neural nets

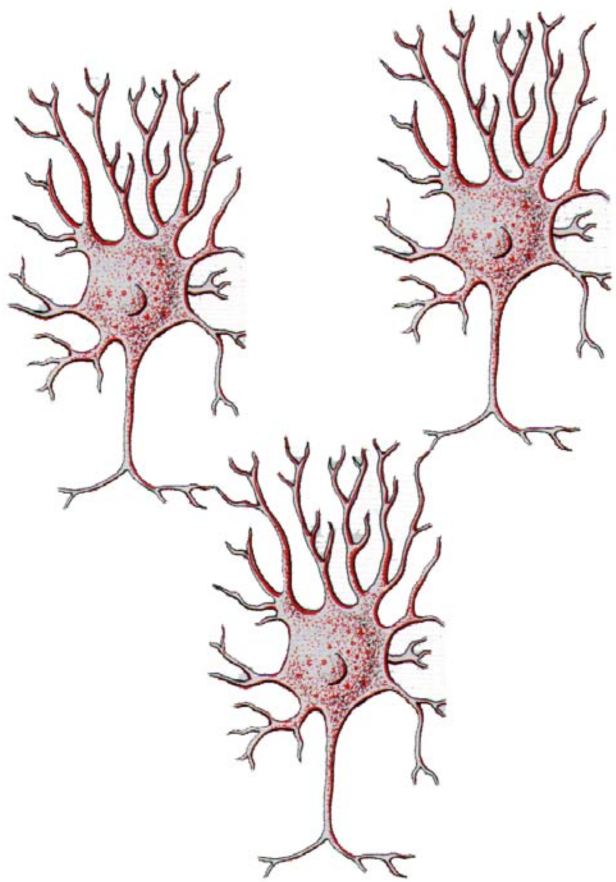


Neuron,
with dendrites, soma, and axon.

(Purkinje cell from cerebellum)

Picture:
Spektrum der Wissenschaft 10,
October 2001

Biological neural nets



Potentials are being propagated from the dendrites to the soma.

If the accumulated potential is above a certain threshold, the neuron fires.

The resulting potential is being propagated to other neurons via the axon.

Pictures: Birbaumer & Schmidt, Biologische Psychologie, ^{2}1991

Artificial neural nets

(Finite) set of *units* (nodes, neurons) with *connections*.

▶ graph

Possible abstractions:

- Potentials are real numbers.
- Propagation doesn't need time.
- Potentials accumulate as weighted sums.
  (Weights stand for synaptic activity and can be *learned*.)
- Units become active in discrete time steps.
- Activation function is (basically) the same everywhere in the network.

There exist many competing architectures.

Artificial neural nets

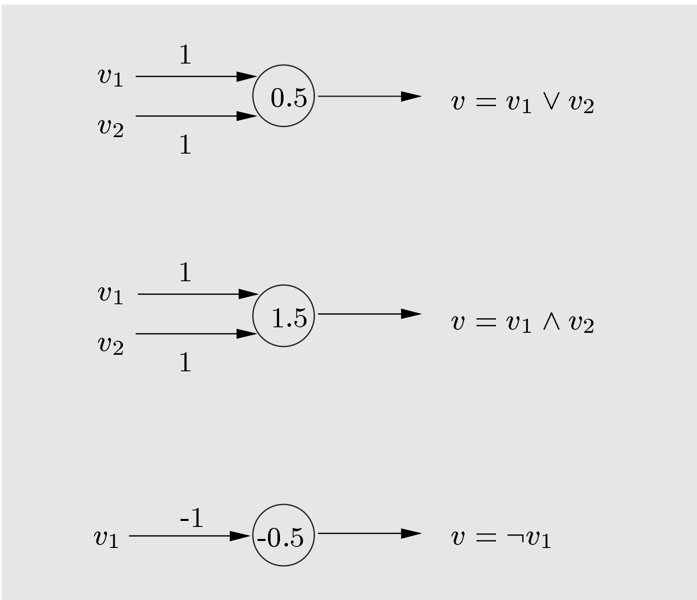
In particular:

- Every unit computes a *simple* input-output function.
- The units are *blind* concerning the sources of their input and the targets of their output.

Information (knowledge)
is being represented by the
(*weighted*) *connections*
in the network!

▶ *Connectionist systems.*

McCulloch-Pitts networks



McCulloch & Pitts 1943

Neurons with binary activation functions for $\vee$, $\wedge$, $\neg$.

Updates are being computed for all units at the same time.

McCulloch-Pitts networks are exactly the finite automata.

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

McCulloch-Pitts networks: extensions

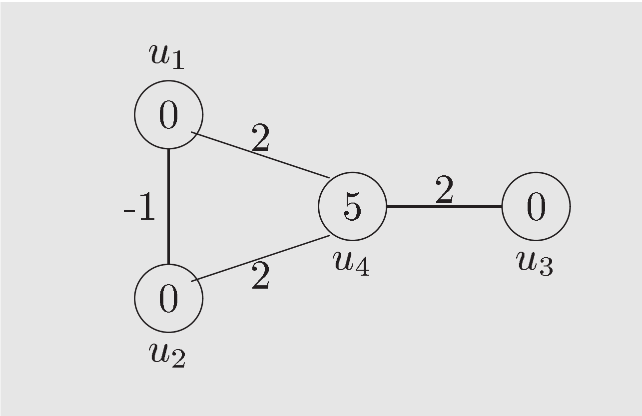
Hölldobler & Kalinke 1994: Representation of propositional logic programs by 3-layer feedforward networks.

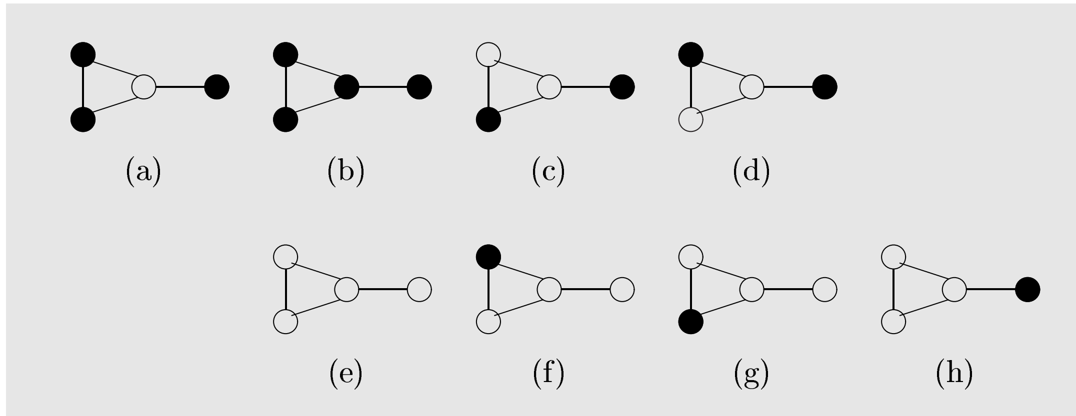
D'Avila Garcez, Broda, Gabbay, Zaverucha 1999/2001:
Extension to sigmoidal (differentiable) activation functions.
Learning possible via backpropagation (gradient descent).

Idea:

- Knowledge representation via network.
  - Learning via backpropagation.
    - Extraction of learned knowledge.

Symmetric nets and propositional logic





Pinkas 199x: Hopfield networks with symmetric connections.

Update by probabilistic choice of unit.

There exsists relation between stable states in network
and models of propositional formulae (via energy minimization).

▶ Treatment of some non-monotonic propositional logic.

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

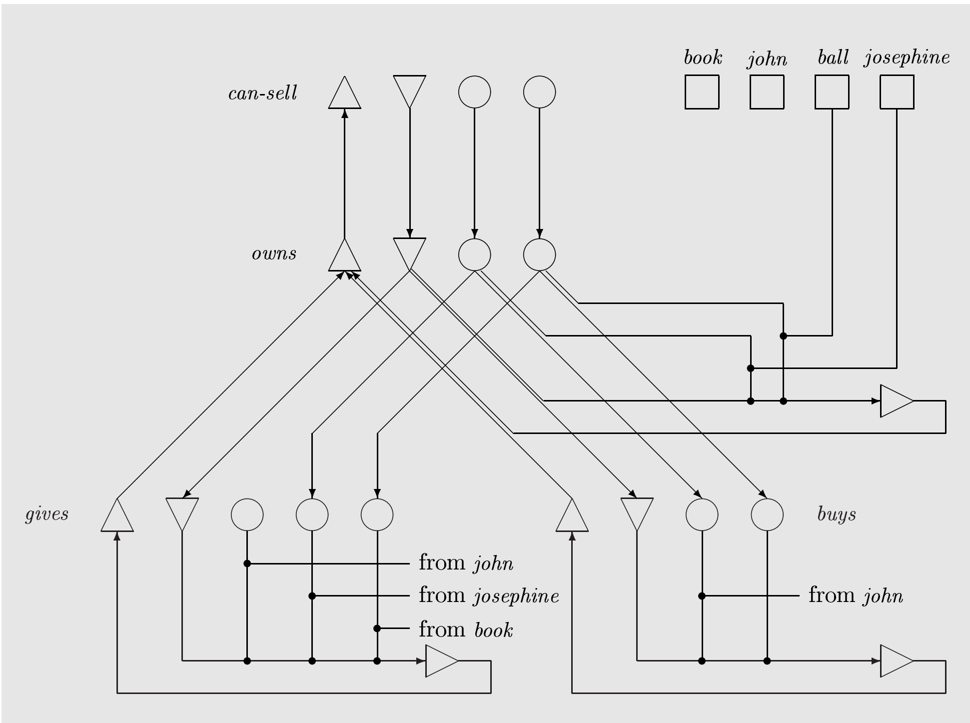
Beyond propositional logic

Variable bindings?

Term representation?

Infinite ground instantiations?

SHRUTI



Shastri & Ajjanagadde 1993

Variable binding
via time synchronization.

Reflexive (i.e. fast)
reasoning possible.

Picture: Hölldobler,
Introduction to Computational Logic, 2001

gives(X,Y,Z) → owns(Y,Z)	gives(john,josephine,book)
buys(X,Y) → owns(X,Y)	(∃X) buys(john,X)
owns(X,Y) → can-sell(X,Y)	owns(josephine,ball)

SHRUTI

rules: $\forall(p_1(\dots) \wedge \dots \wedge p_n(\dots) \rightarrow (\exists Y_1, \dots, Y_k) p(\dots))$

facts and queries: $\exists(q(\dots))$

Some restrictions:

- No function symbols other than constants.
- In facts or queries each variable occurs only once.
- Every variable which occurs at least twice in the hypothesis of some rule, must also occur in the consequence of the rule. Furthermore, it must be instantiated in the moment when the consequence is being unified with a query.
- The number of applications of some rule within a derivation is globally bounded.

Logic programs

A (*normal logic*) *program* P is a finite set of *clauses* of the form

$$\forall(A \leftarrow L_1 \wedge \cdots \wedge L_n),$$

in short

$$A \leftarrow L_1, \ldots, L_n,$$

over some first-order language, where A is an atom, all L_i are literals.
(A *head*, $L_1, \ldots, L_n$ *body* of the clause.)

P is called *definite*, if P does not contain negation.

B_P: set of all ground instances of atoms (Herbrand base).
$I_P = 2^{B_P}$: set of all interpretations (complete lattice wrt. $\subseteq$).
ground(P): set of all ground instances of clauses in P.

Fixed point semantics

P program. Define $T_P : I_P \to I_P$ by:

$T_P(I)$ is set of all $A \in B_P$ for which there is a clause $A \leftarrow L_1, \ldots, L_n$ in ground(P) with $I \models L_1 \wedge \cdots \wedge L_n$.

Properties of the *single-step operator* T_P:

- Models of P are pre-fixed points of T_P.
  (Fixed points are called *supported models*.)
- T_P is *monotonic* and *Scott continuous* if P is definite.
- T_P is in general not monotonic if P is not definite.

Scott continuity

P definite program. Then:

- $T_P(I) \subseteq T_P(J)$ for all $I \subseteq J$ (monotonicity).
- $\sup T_P(I_n) = T_P(\sup I_n)$ for all directed (I_n).

I.e. T_P *Scott continuous*. Subbase of the Scott topology on I_P:

$\{\mathcal{G}(A) : A \in B_P\}$ mit $\mathcal{G}(A) = \{I \in I_P : I \models A\}$.

Theorem

X complete lattice (Scott-Ershov domain).
$f : X \to X$ Scott continuous.
Then f has a least fixed point.

The least fixed point of T_P (Herbrand case) yields *denotational semantics* of P.

Negation: Cantor topology

P not definite, then T_P in general not monotonic.

Theorem not applicable.

$\mathcal{G} = \{\mathcal{G}(A) : A \in B_P\} \cup \{\mathcal{G}(\neg A) : A \in B_P\}$ with $\mathcal{G}(L) = \{I \in I_P : I \models L\}$
Subbase of the *Cantor topology* Q on I_P.

B_P countable, then (I_P, Q) homeomorphic to the *Cantor set* $\mathcal{C}$ on $\mathbb{R}$.

Batarekh & Subrahmanian 1989 (*query topology*)
Seda 1995 (*atomic topology*)

Results

$\lim T_P^n(I)$ is model of P, if existent.
(Hitzler & Seda 1997)

$\lim T_P^n(I)$ is supported model of P, if existent and T_P continuous.

Semi-syntactic characterization of continuity.
(Seda 1995)

Generalized metric treatment of fixed-point semantics.
(Hitzler & Seda, TCS 2003)

Consequence operators

Truth values $\mathcal{T} = \{t_1, \dots, t_n\}$.

Interpretations are functions $I : B_P \to \mathcal{T}$.

$I_{P,n} = I_P$ set of all interpretations.

$\mathcal{B}_A$ set of all atoms in bodies of clauses in ground(P) with head $A \in B_P$.

$T : I_P \to I_P$ *consequence operator* for P, if for all $I \in I_P$ and all $A \leftarrow \texttt{body}$ in P we have that $T(I)(A) \leftarrow I(\texttt{body})$ holds via truth table.

T *local* if $T(I)(A) = T(K)(A)$ for all $A \in B_P$ and all $I, K \in I_P$ which agree on $\mathcal{B}_A$.

T_P is a local consequence operator.

Other examples: Operators as defined by Fitting (1985,199x) in three- or four-valued logic.

Cantor topology $\mathcal{Q}$

$\mathcal{Q}$ is the product topology on $I_P = \mathcal{T}^{B_P}$,
where $\mathcal{T} = \{t_1, \ldots, t_n\}$ carries the discrete topology.

$\mathcal{Q}$ is totally disconnected, compact, Hausdorff, second countable, dense in itself.

$\mathcal{Q}$ is metrizable and homeomorphic to the Cantor set. (B_P is countable.)

Continuity in $\mathcal{Q}$

Consequence operator T on I_P is *locally finite*, if for all $A \in B_P$ and all $I \in I_P$ there exists a finite set $S \subseteq \mathcal{B}_A$ with $T(J)(A) = T(I)(A)$ for all $J \in I_P$ which agree with I on S.

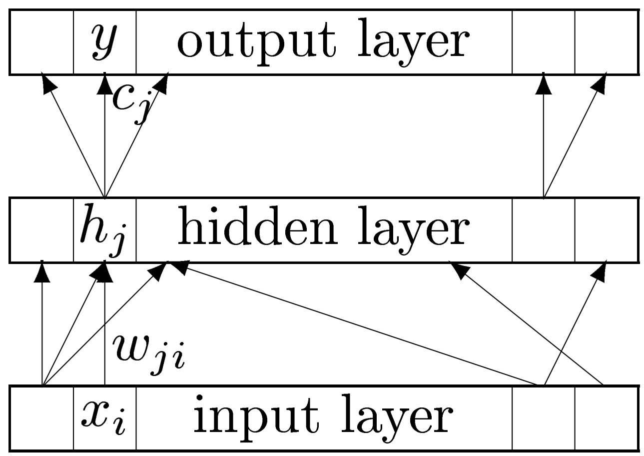
Theorem

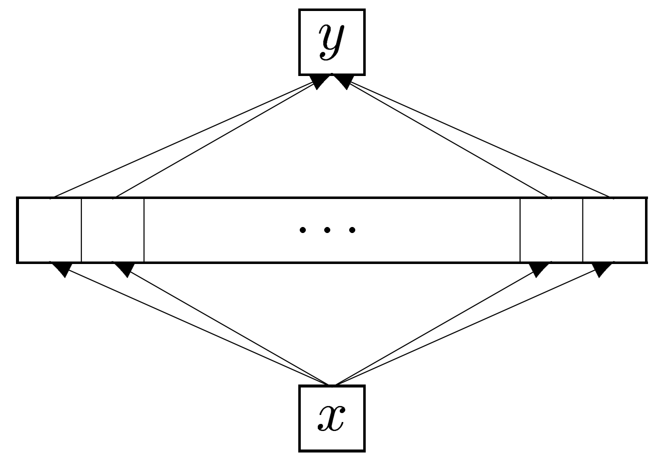
A local consequence op. is locally finite iff it is continuous in $\mathcal{Q}$.

Sufficient:

- P is *covered*, i.e. does not contain any *local variables* (occuring in some body, but not in corresponding head).

3-layer feedforward nets (3lfn)





x_i inputs; y output; w_{ji}, c connection weights

I/O-function:

$$y = f(x_1, \ldots, x_r) = \sum_j c_j \sigma \left(\sum_i w_{ji} x_i - \theta_j \right)$$

θ_j thresholds

σ activation function (e.g. sigmoidal or gaussian bell)

Idea
(Hölldobler, Kalinke, Störr 1994/1999)

Representation of a logic program P
by representing its single-step operator T_P.

Yields:
Representation of a program with infinite ground instantiation.

Hölldober, Kalinke & Störr 1999

Sought: suitable imbedding of I_P into $\mathbb{R}$.

Hölldobler, Kaline & Störr 1999: special case of acyclic programs with injective level-mapping.

Via: metric by Fitting (1994) and representation as 4-adic numbers.

Approximation of T_P only.
Pure existence proof. No idea how to construct networks.

Continuity

Theorem (Funahashi 1989, simplified version):

σ sigmoidal
$K \subseteq \mathbb{R}$ compact,
$f : K \to \mathbb{R}$ continuous,
$\varepsilon > 0$.

Then there exists 3lfn with sigmoidal σ and I/O-function $\bar{f} : K \to \mathbb{R}$ with

$$\max_{x \in K} \left\{ d\left(f(x), \bar{f}(x) \right) \right\} < \varepsilon;$$

d metric which induces natural topology on $\mathbb{R}$.

I.e. every continuous function $f : K \to \mathbb{R}$ can be uniformly approximated by I/O-functions of 3lfns.

Approximation of continuous consequence operators

Theorem (Hitzler & Seda 2001)

Let P be a logic program, T be a locally finite consequence operator, and ι be a homeomorphism from $(I_{P,n}, \mathcal{Q})$ to $\mathcal{C}$.
Then $\iota(T)$ can be uniformly approximated by I/O-functions of 3lfns.

This holds *mutatis mutandis* e.g. for radial basis function networks (activation function is gaussian).

ι normally given via some enumeration (injective level mapping) $l : B_P \to \mathbb{N}$ and some corresponding p-adic expansion.

Measurability

Satz (Hornik, Stinchcombe, White 1989, simplified version)

$\sigma : \mathbb{R} \to (0,1)$ monotonic increasing, onto.
$f : \mathbb{R} \to \mathbb{R}$ Borel measurable,
μ Borel probability measure on $\mathbb{R}$,
$\varepsilon > 0$.

Then there is a 3lfn with sigmoidal activation function σ and I/O-function $\bar{f} : \mathbb{R} \to \mathbb{R}$ with

$$\varrho_\mu(f, \bar{f}) = \inf\left\{\delta > 0 : \mu\left\{x : \left|f(x) - \bar{f}(x)\right| > \delta\right\} < \delta\right\} < \varepsilon.$$

I.e. the set of I/O-functions which can be computed using 3lfns is dense with respect to ϱ_μ in the set of all Borel measurable functions $f : \mathbb{R} \to \mathbb{R}$.

Measurable consequence operators

Satz (Hitzler & Seda 2000)

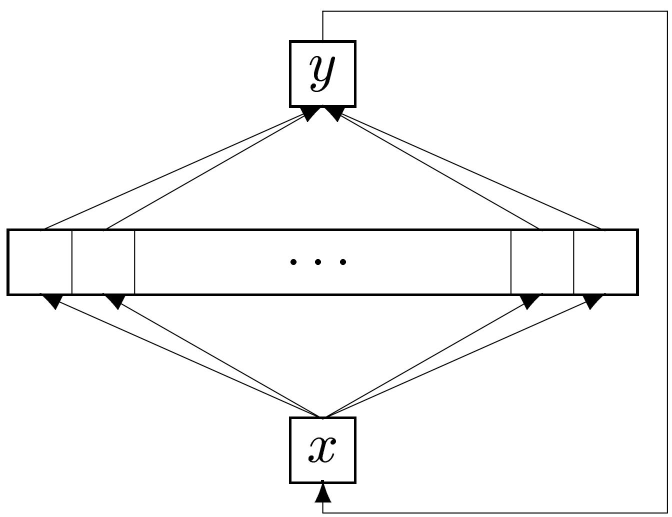
Local consequence operators are always measurable with respect to $\sigma(\mathcal{Q})$.

But:

Approximation by 3lfns is only *almost everywhere*.

Cantor set has measure 0.

Rekursive architecture



Results do not indicate how approximating networks could be constructed.

But results on the behaviour of these networks can be obtained.

Hölldobler, Störr und Kalinke 1999 use recursive architecture (left).

Network iterates consequence operator.

Recursive architecture

T locally finite consequence operator.
f I/O-function of approximating network.

For all $I \in I_P$ and all $n \in \mathbb{N}$ we have

$$|f^n(\iota(I)) - \iota(T^n(I))| \leq \varepsilon \frac{1-\lambda^n}{1-\lambda}.$$

Need: λ Lipschitz-constant of F, the continuation of $\iota(T)$ to $[0,1]$.
ε bound on approximation error.

Recursive architecture

If F is a contraction on $[0,1]$, then $\left(F^k(\iota(I))\right)$ converges for all I to the unique fixed point x of F and there is $m \in \mathbb{N}$ such that for all $n \geq m$ we have

$$|f^n(\iota(I)) - x| \leq \varepsilon \frac{1}{1-\lambda}.$$

Furthermore, T is a contraction on the complete space I_P (with suitable metric), and we have $\iota(M) = x$ for the unique fixed point M of T.

Recursive architecture

Assume there is $I \in I_P$ such that $T^n(I)$ converges in $\mathcal{Q}$ to a fixed point M of T.

Then for every $\delta > 0$ there exists some $n \in \mathbb{N}$ and a network with $|f^n(\iota(I)) - \iota(M)| < \delta$.

Acyclic programs

A logic program P is *acyclic* if there exists a level mapping $l : B_P \to \mathbb{N}$ such that for all $A \in B_P$ and all $B \in \mathcal{B}_A$ we have $l(A) > l(B)$.

Let $d : I_P \times I_P \to \mathbb{R}$ be defined by $d(I, J) = 2^n$, where n is least such that I and J disagree on some atom of level n.

d ist a complete metric on I_P.

Let P be acyclic and T be a local consequence operator for P. Then T is a contraction with respect to d and $T^n(I)$ converges in $\mathcal{Q}$ for all $I \in I_P$ to the unique fixed point of T.

More general programs

For much more general programs we know about metrics with respect to which T_P is a contraction. But these metrics are in general not imbeddable into the reals.

(Hitzler & Seda, TCS 2003)
(Hitzler & Seda 2001)

Non-monotonic reasoning

Fixed points of opertor GL_P yield the *stable models* of P.
(as in Answer Set Programming)

[DK89] Phan M. **Dung** and Kanchana **Kanchanasut**, A fixpoint approach to declarative semantics of logic programs. Proc. NACLP'89, 1989.

Program transformation $P \mapsto \mathrm{fix}(P)$.
Complete unfolding through positive body literals.

[Wen02] Matthias **Wendt**, Unfolding the well-founded semantics, Journal of Electrical Engineering 2002.

Shows $\mathsf{GL}_P(I) = T_{\mathrm{fix}(P)}(I)$ for all interpretations I.

$\leadsto$ Allows to carry over results. (Bader & Hitzler, in preparation)

Towards constructing approximating networks

(Bader 2003), (Bader & Hitzler JAL 200x)

Represent/approximate graph of T_P
by attractor of iterated function system on $\mathbb{R}^2$.

Encode iterated function system
by recurrent radial basis function network.

Towards constructing approximating networks

(Bader & d'Avila Garcez, in preparation)

Using fibred networks.

Fibred networks:
Talk by Artur d'Avila Garcez, tomorrow, 15:20 hrs, GRU 350.

Towards constructing approximating networks

(Bader & Hitzler, in preparation)

If $\iota(T_P)$ continuous, then also uniformly continuous.

This means: Changes between $\iota(I)$ and $\iota(T_P(I))$ up to some given $\varepsilon > 0$ are caused (only) by atoms of level smaller than some $l(\varepsilon)$.

If we know $l(\varepsilon)$, we can explicitly read off the necessary parameters for an approximating RBF-network.

Closing

Neural-symbolic integration:
We still don't know how to do it.

(But we're getting closer.)

The Fixpoint Completion

Quasi-interpretation Q: set of clauses of form $A \leftarrow \neg B_1, \dots, \neg B_m$.

Program P: set of (ground) clauses of form

$$A \leftarrow A_1, \dots, A_n, \neg B_1, \dots, \neg B_m.$$

$T'_P(Q)$ set of $A \leftarrow \mathtt{body}_1, \dots, \mathtt{body}_n, \neg B_1, \dots, \neg B_m$
where $A \leftarrow A_1, \dots, A_n, \neg B_1, \dots, \neg B_m$ in P
and $A_i \leftarrow \mathtt{body}_i$ in Q for all i.

$T'_P \uparrow \omega = \mathrm{lfp}(T'_P) = \mathrm{fix}(P)$ quasi-interpretation.

The Gelfond-Lifschitz Operator

$T_P(I)$ set of all A
with $A \leftarrow L_1, \ldots, L_n$ in P and $I \models L_1, \ldots, L_n$.

P/I set of all $A \leftarrow A_1, \ldots, A_n$
with $A \leftarrow A_1, \ldots, A_n, \neg B_1, \ldots, \neg B_m$ in P
and $I \not\models B_1, \ldots, I \not\models B_m$.

$\mathrm{GL}_P(I) = \mathrm{lfp}(T_{P/I})$. (Gelfond & Lifschitz 1988)

For all interpretations I: $\mathrm{GL}_P(I) = T_{\mathrm{fix}(P)}(I)$. [Wen02]

In fact even: $\Psi_P(I) = \Phi_{\mathrm{fix}(P)}(I)$.

Self-Similarity

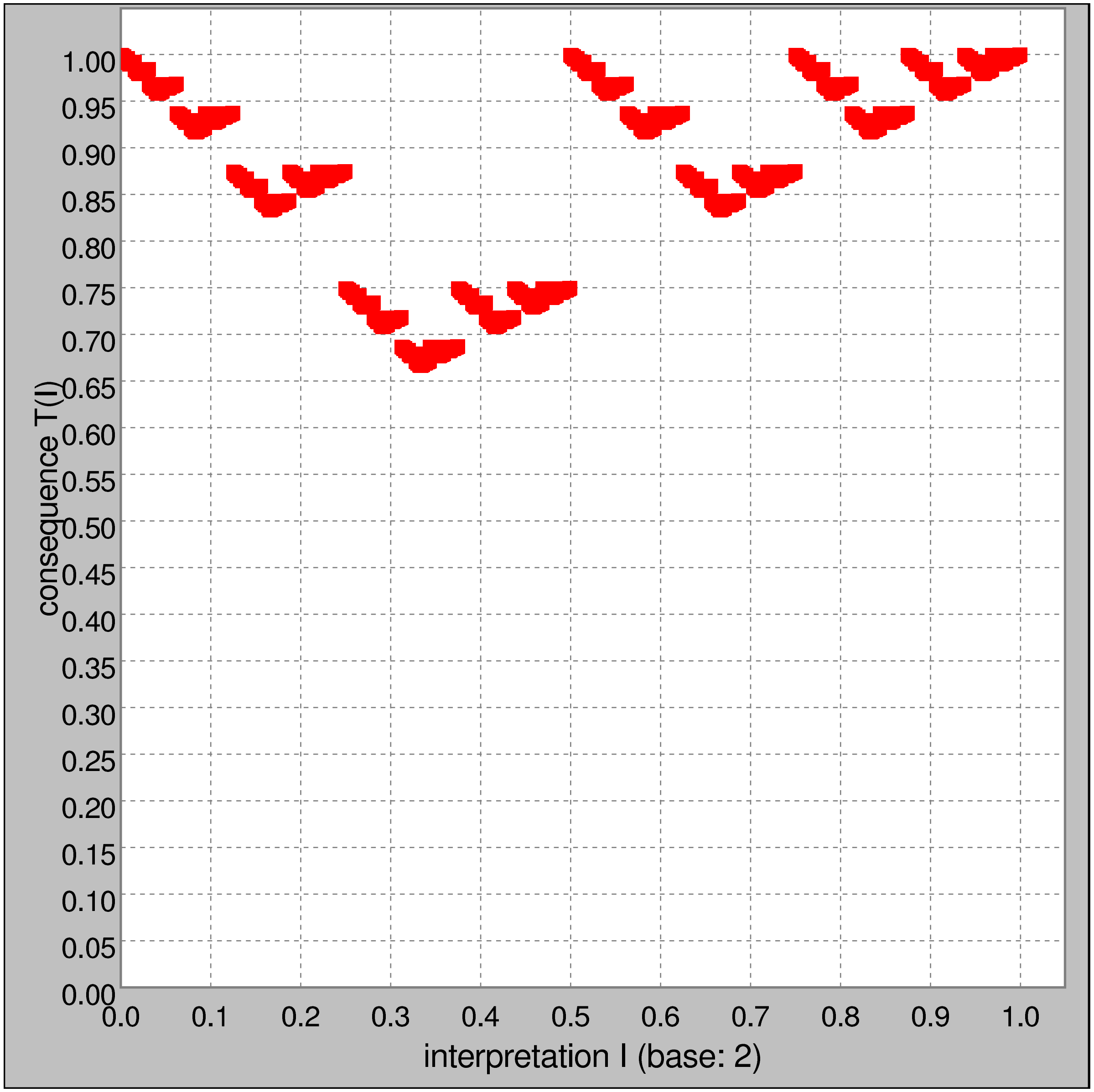
An obseration by Sebastian Bader.

Graph of T_P visualized via embedding into $[0,1] \times [0,1]$ using p-adic numbers.

$R : I_P \to \mathbb{R} : I \mapsto \sum_{A \in I} B^{-l(A)}$, where $l : B_P \to \mathbb{N}$ injective, $B > 2$.

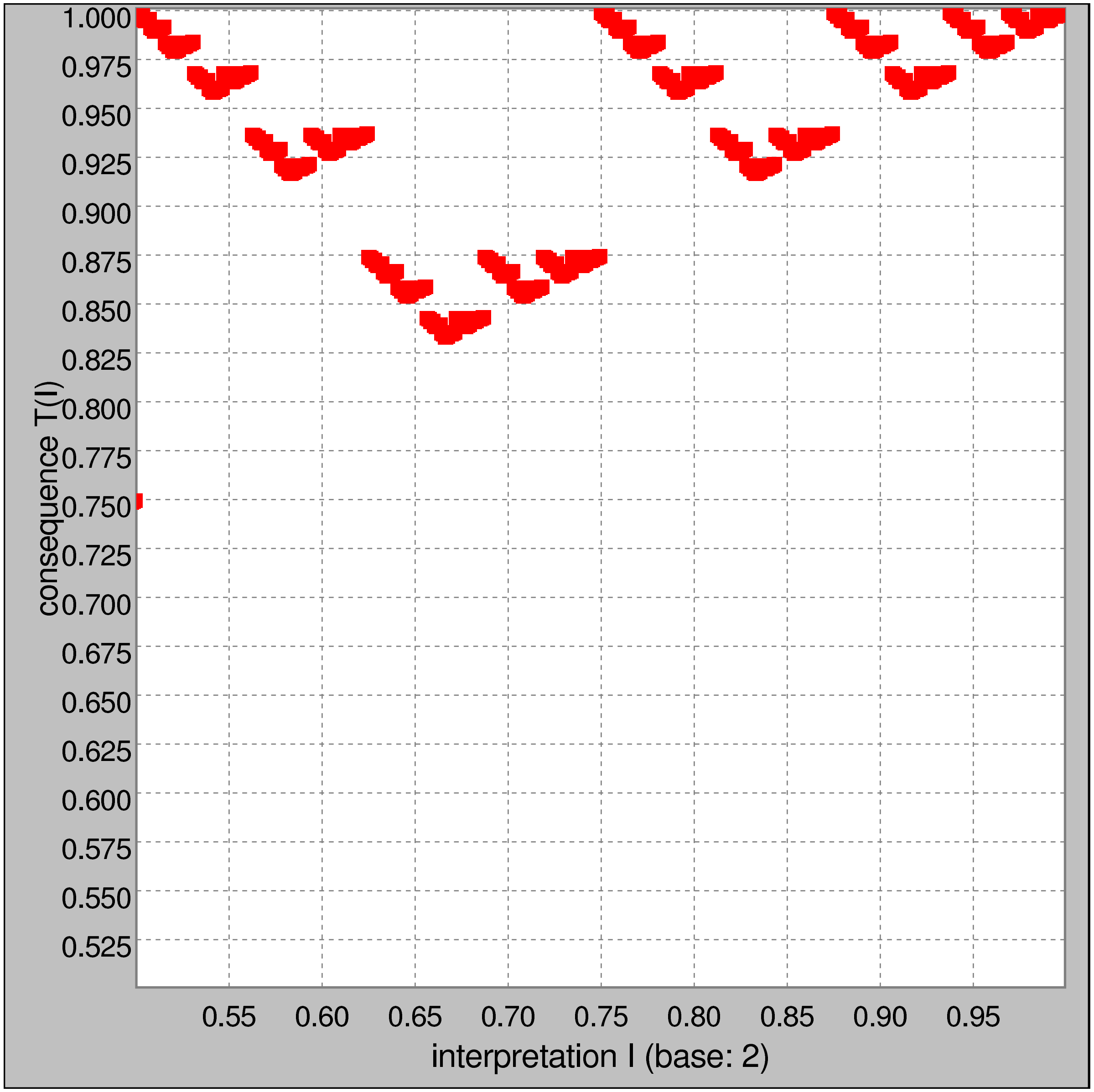
Graph shows self-similarity.

(The following pictures were provided by Sebastian Bader.)



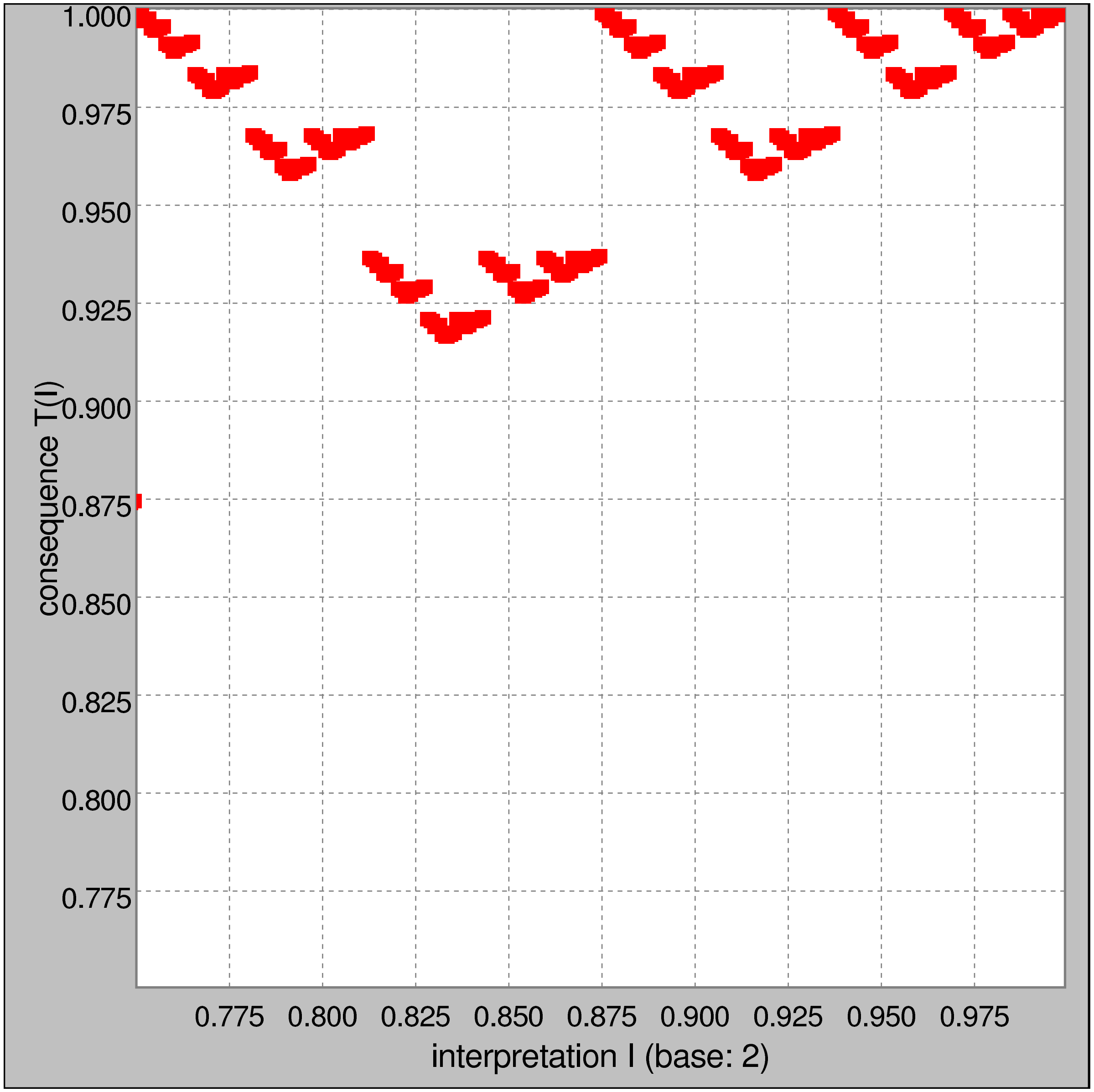
p(0).
p(s(X)) :– p(X).
p(X) :– not p(X).

p(s(s(s(s(s(s(s(s(s(s(0))))))))))) => 11
p(s(s(s(s(s(s(s(s(s(0)))))))))) => 10
p(s(s(s(s(s(s(s(s(0))))))))) => 9
p(s(s(s(s(s(s(s(0)))))))) => 8
p(s(s(s(s(s(s(0))))))) => 7
p(s(s(s(s(s(0)))))) => 6
p(s(s(s(s(0))))) => 5
p(s(s(s(0)))) => 4
p(s(s(0))) => 3
p(s(0)) => 2
p(0) => 1



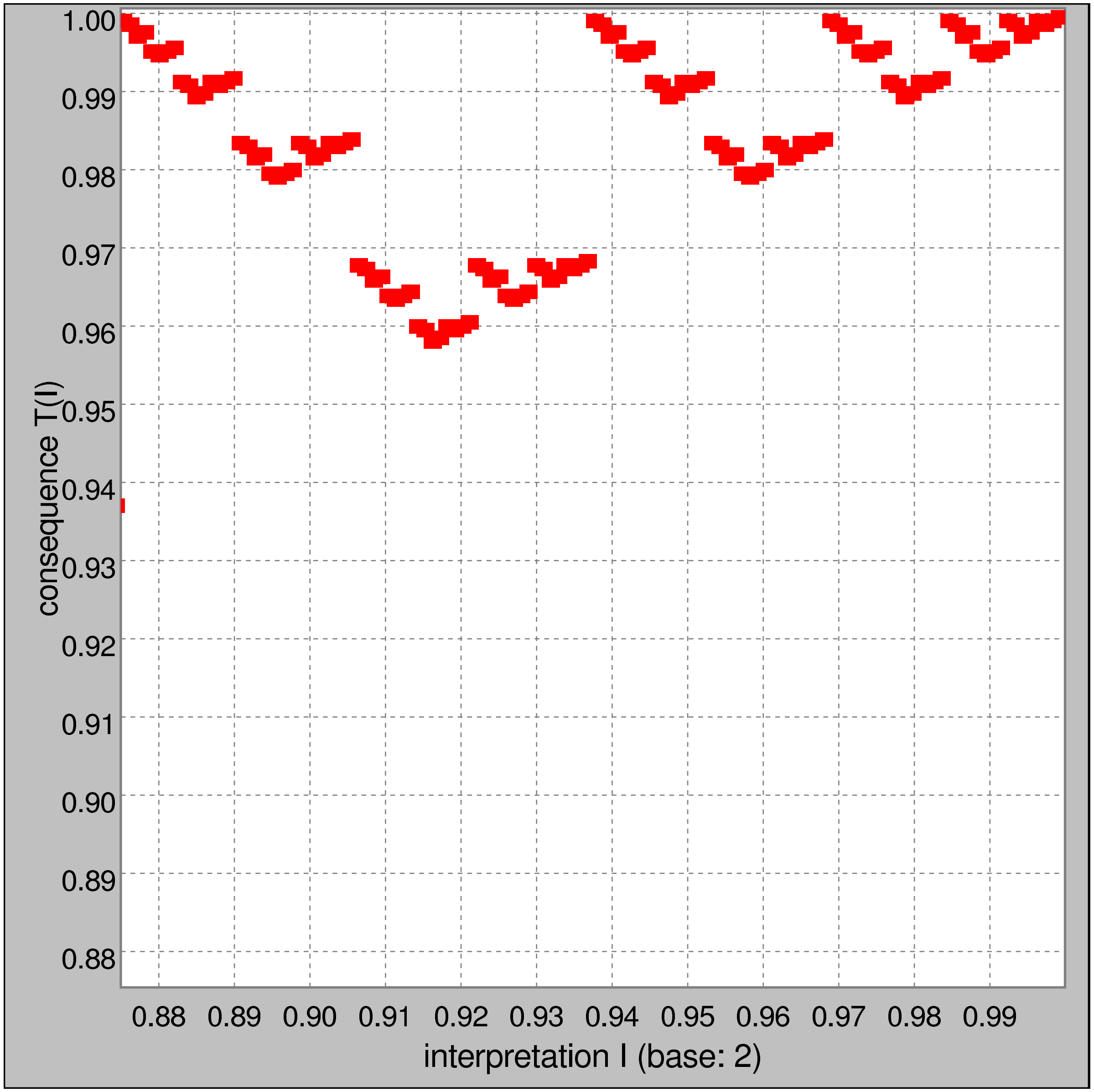

```
p(0).
p(s(X)) :- p(X).
p(X) :- not p(X).
```

```
p(s(s(s(s(s(s(s(s(s(s(0)))))))))) => 11
p(s(s(s(s(s(s(s(s(s(0))))))))) => 10
p(s(s(s(s(s(s(s(s(0)))))))) => 9
p(s(s(s(s(s(s(s(0))))))) => 8
p(s(s(s(s(s(s(0)))))) => 7
p(s(s(s(s(s(0))))) => 6
p(s(s(s(s(0)))) => 5
p(s(s(s(0)))) => 4
p(s(s(0))) => 3
p(s(0)) => 2
p(0) => 1
```



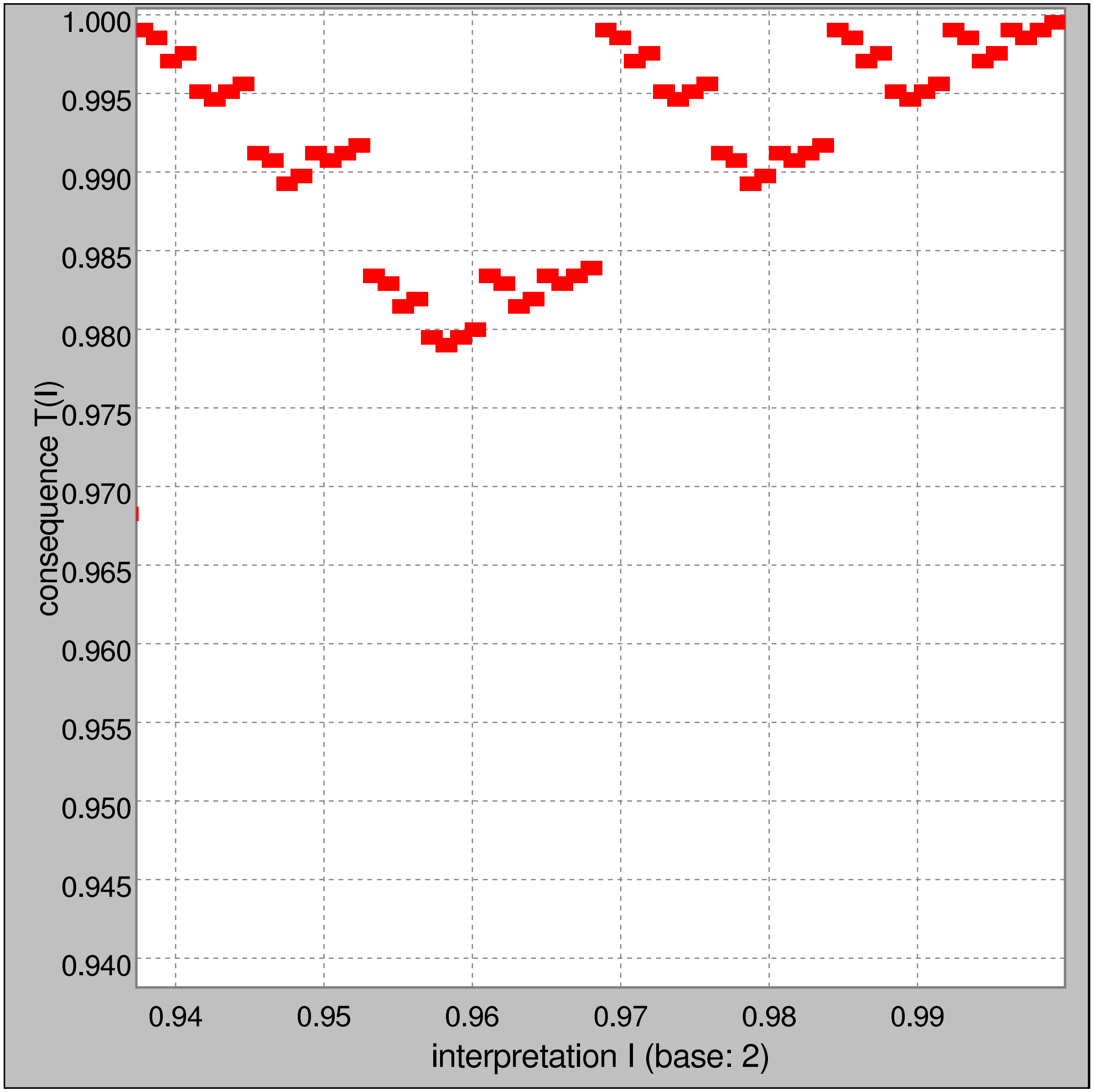

```
p(0).
p(s(X)) :- p(X).
p(X) :- not p(X).
```

```
p(s(s(s(s(s(s(s(s(s(s(0))))))))))) => 11
p(s(s(s(s(s(s(s(s(s(0)))))))))) => 10
p(s(s(s(s(s(s(s(s(0))))))))) => 9
p(s(s(s(s(s(s(s(0)))))))) => 8
p(s(s(s(s(s(s(0))))))) => 7
p(s(s(s(s(s(0)))))) => 6
p(s(s(s(s(0))))) => 5
p(s(s(s(0)))) => 4
p(s(s(0))) => 3
p(s(0)) => 2
p(0) => 1
```




```
p(0).
p(s(X)) :- p(X).
p(X) :- not p(X).
```

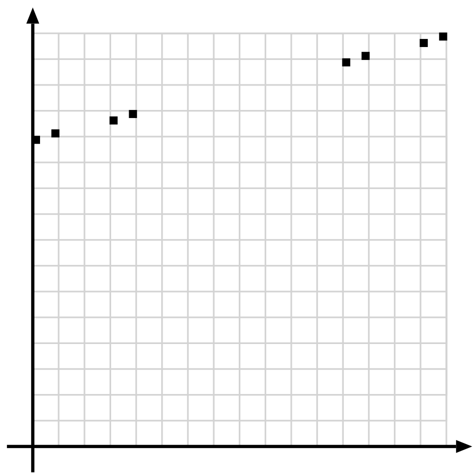
```
p(s(s(s(s(s(s(s(s(s(s(0))))))))))) => 11
p(s(s(s(s(s(s(s(s(s(0)))))))))) => 10
p(s(s(s(s(s(s(s(s(0))))))))) => 9
p(s(s(s(s(s(s(s(0)))))))) => 8
p(s(s(s(s(s(s(0))))))) => 7
p(s(s(s(s(s(0)))))) => 6
p(s(s(s(s(0))))) => 5
p(s(s(s(0)))) => 4
p(s(s(0))) => 3
p(s(0)) => 2
p(0) => 1
```



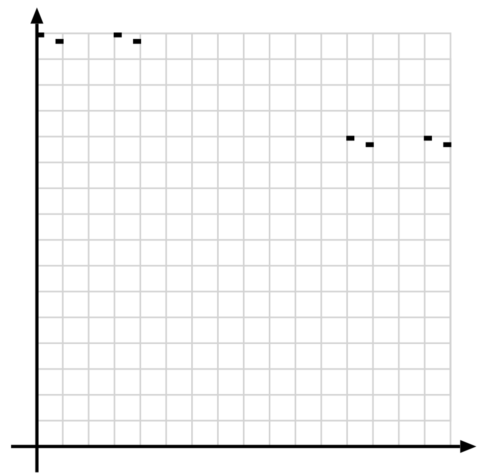

```
p(0).
p(s(X)) :- p(X).
p(X) :- not p(X).
```

```
p(s(s(s(s(s(s(s(s(s(s(0))))))))))) => 11
p(s(s(s(s(s(s(s(s(s(0)))))))))) => 10
p(s(s(s(s(s(s(s(s(0))))))))) => 9
p(s(s(s(s(s(s(s(0)))))))) => 8
p(s(s(s(s(s(s(0))))))) => 7
p(s(s(s(s(s(0)))))) => 6
p(s(s(s(s(0))))) => 5
p(s(s(s(0)))) => 4
p(s(s(0))) => 3
p(s(0)) => 2
p(0) => 1
```

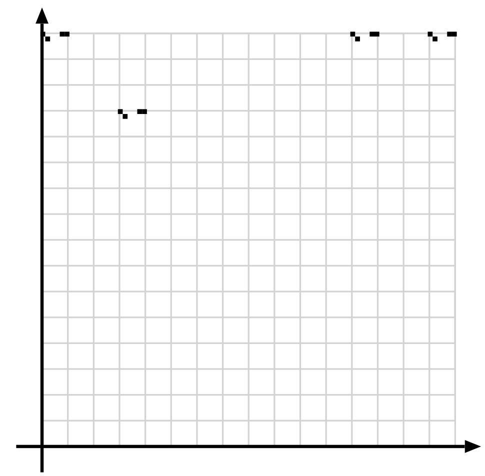
Examples of graphs of logic programs



```
n(0).
n(s(X))←n(X).
```

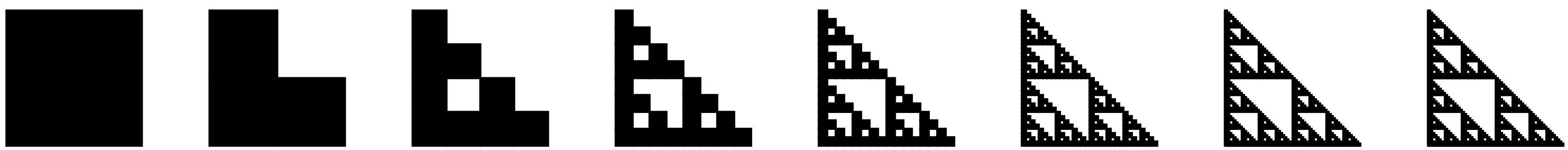


```
e(0).
e(s(X))←not e(X).
o(X)←not e(X).
```



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

(Hyperbolic) Iterated function systems (IFSs)



Space $\mathcal{H}$: Compact subsets of $\mathbb{R}^2$ with *Hausdorff metric*.
Set $\Omega = \{\omega_i\}$ of *contraction mappings* on $\mathbb{R}^2$.
$\bigcup \Omega(A) = \bigcup_i \omega_i(A)$ contraction on $\mathcal{H}$ with unique fixed point (*attractor*).

First representation theorem

P logic program. $R : I_P \to \mathbb{R}$ p-adic embedding.

$\left(\mathbb{R}^2, d, \Omega = \left\{\left(\omega_i^1, \omega_i^2\right)\right\}\right)$ hyperbolic IFS, attractor A.

Then

$$\mathrm{graph}(R(T_P)) = A$$

iff

$$\pi_1(A) = \mathrm{range}(R) \text{ and}$$
$$R(T_P)\left(\omega_i^1(a)\right) = \omega_i^2(a) \text{ for all } a \in \mathrm{graph}(R(T_P)) \text{ and all } i.$$

Second representation theorem

(Bader & Hitzler 2003, to appear in JAL)

P logic program with Lipschitz-continuous $R(T_P)$.
Then there exists IFS with attractor graph($R(T_P)$).

Idea: Set $\omega_i^2(x) = R(T_P)(\omega_i^1(x))$.
Choose $\omega_i^1(x)$ such that it generates range(R). This is possible with arbitrarily small contraction, the necessary size of which can be determined by the Lipschitz constant of $R(T_P)$.

Concrete approximation by interpolation

$a \in \mathbb{N}$ *accuracy.* (JAL to appear)

l injective level mapping (enumeration of B_P).

Interpolation points: $(R(I), R(T_P(I)))$, where $I \in D = \{A \mid l(A) < a\}$.

IFS with $\Omega_a = \{(\omega_i^1, \omega_i^2)\}$, where

$$\omega_i^1(x) = \frac{1}{B^a} x + d_i^1$$

$$\omega_i^2(x) = \frac{1}{B^a} + R(T_P)\left(d_i^1\right) - \frac{R(T_P)(0)}{B^a}$$

Attractors A_a are graphs of continuous functions.

$(A_a)_a$ converges in function space (with sup-metric) to $R(T_P)$
if $R(T_P)$ Lipschitz-continuous.

Encoding as radial basis function network