University of Oxford
Machine Learning seminars
9 March 2017

Neural-Symbolic Systems for Verification, Run-Time Monitoring and Learning

Artur d'Avila Garcez
City, University of London
a.garcez@city.ac.uk

Neural-Symbolic Systems

Cognitive Science

↓

Logic

Learning

Neural Computation

↑

Neuroscience

One Structure for Learning and Reasoning

In AI: KR+ML

Why Neurons and Symbols?

"We need a language for describing the alternative algorithms that a network of neurons may be implementing" L. Valiant

(New) Logic + Neural Computation

GOAL: Learning from experience and reasoning about what has been learned in an uncertain environment in a computationally efficient way.

Neural-Symbolic Methodology

high-level symbolic representations
(abstraction, recursion, relations, modalities)

translations

low level, efficient neural structures
(with the same, simple architecture throughout)

Analogy: low-level implementation (machine code) of high-level representations (e.g. java, system requirements)

A Foundational Approach

(as opposed to the neuroscience or the engineering approach)

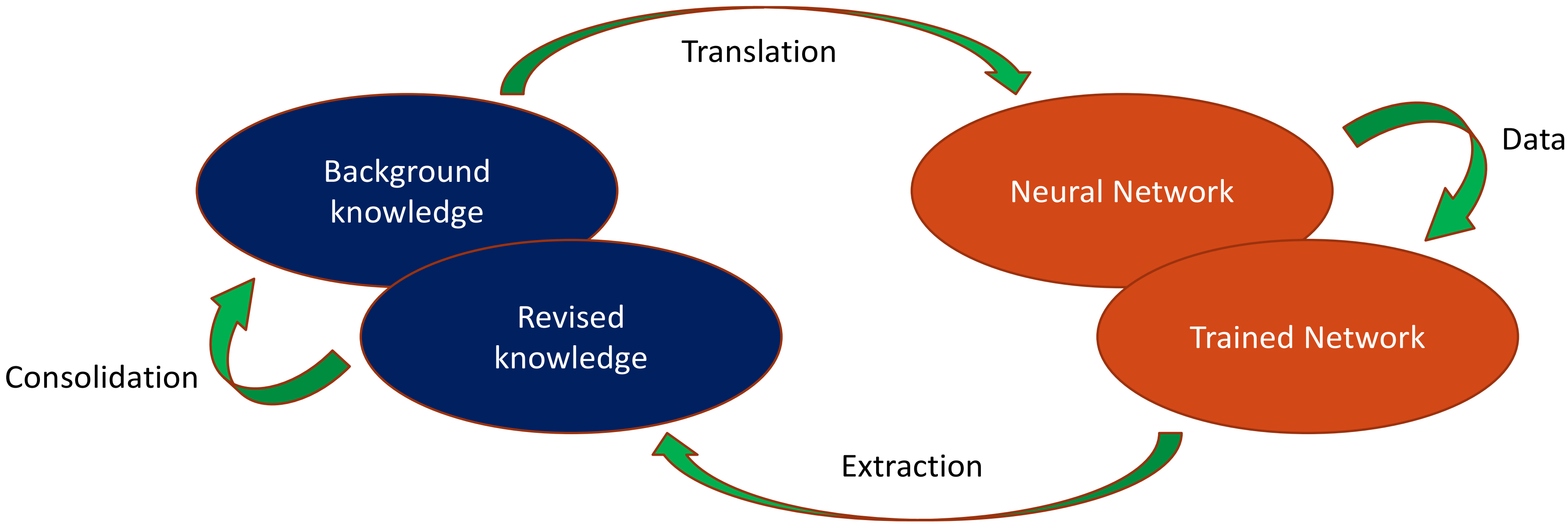
One Structure for Learning and Reasoning:

Take different tasks, consider what they have in common, formalize, evaluate and repeat

KEY: controlling the inevitable accumulation of errors (robustness)

Applications: training in simulators, robocup, evolution of software models, bioinformatics, power plant fault diagnosis, semantic web (ontology learning), general game playing, visual intelligence, finance, explainable AI for personal development.

Neural-Symbolic Learning Cycle



Connectionist Inductive Logic Programming (CILP) System

A Neural-Symbolic System for Integrated Reasoning and Learning (neural nets + logic programming)

- Knowledge Insertion, Revision (Learning) and Extraction

  (based on Towell and Shavik, Knowledge-Based Artificial Neural Networks. AIJ 70:119-165, 1994)

CILP = backpropagation with background knowledge (BK)

- Applications: DNA Sequence Analysis, Power Systems Fault Diagnosis

  CILP test set performance is comparable to backprop.

  CILP test set performance on small training sets is comparable to KBANN and better than backprop.

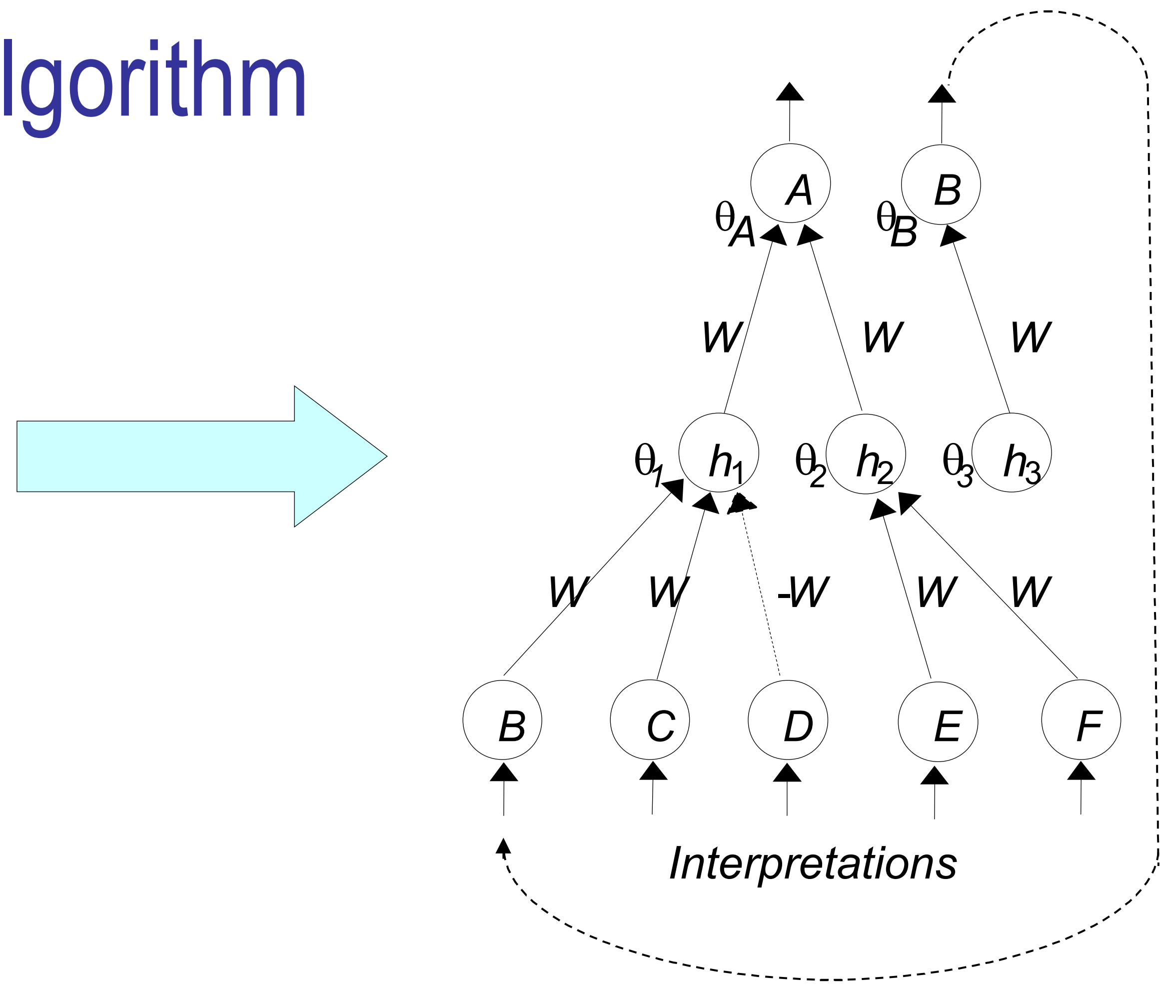
  CILP training set performance is better than backprop. and KBANN

CILP Translation Algorithm

r_1: $A \leftarrow B, C, \sim D;$

r_2 : $A \leftarrow E, F;$

r_3 : $B \leftarrow$



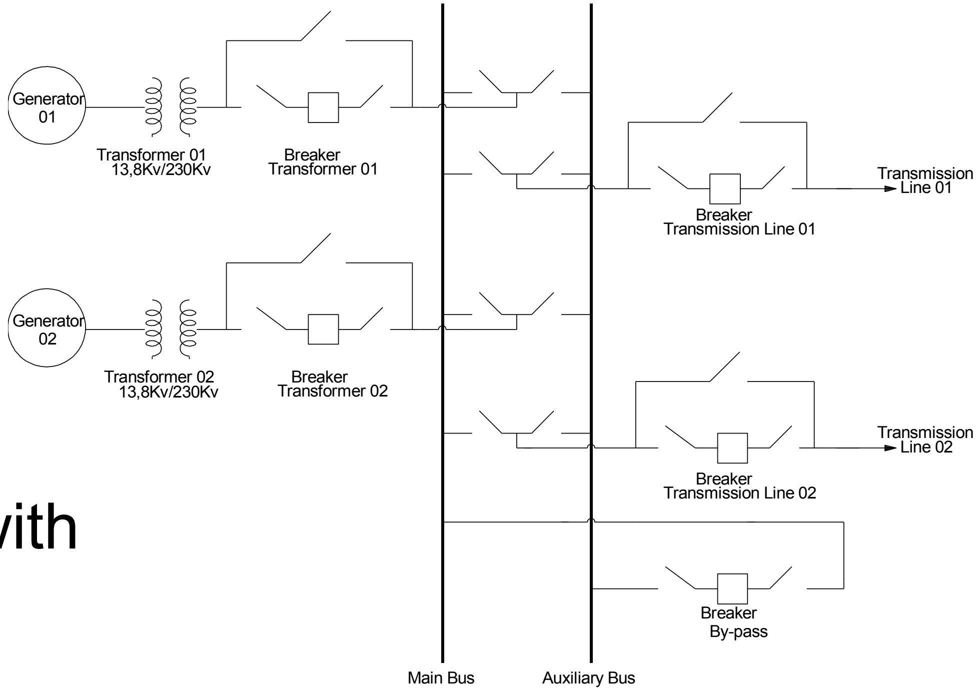
THEOREM: For any logic program P there exists a neural network N such that N computes P

based on Holldobler and Kalinke's translation, but extended to sigmoid neurons (backprop) and hetero-associative networks

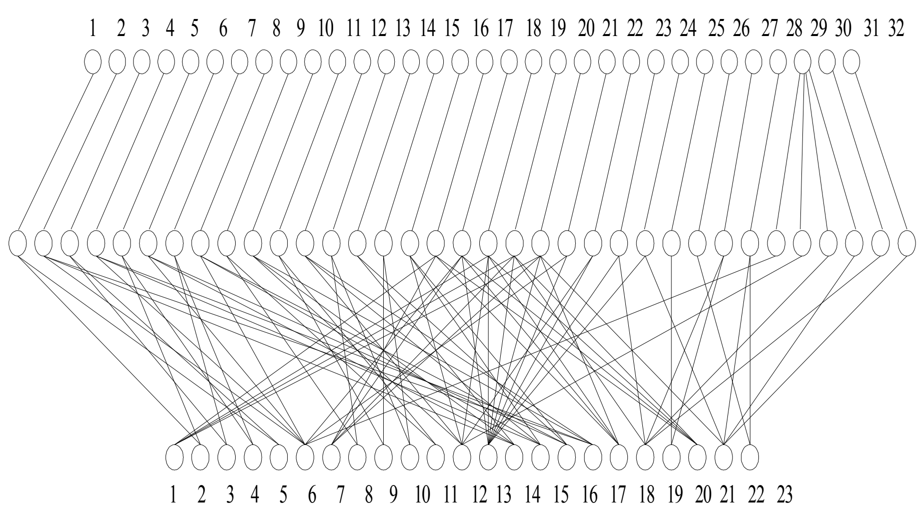
Holldobler and Kalinke, Towards a Massively Parallel Computational Model for Logic Programming. ECAI Workshop Combining Symbolic and Connectionist Processing , 1994.

Power Plant Fault Diagnosis

First real-world application of CILP



Mapping 23 alarms to 32 faults, with 35 rules (with errors) in the BK



Power Plant Fault Diagnosis

Background Knowledge (35 rules with errors)

278 examples of single and multiple faults

Fault(ground,close-up,line01,no-bypass) IF
Alarm(instantaneous,line01) AND
Alarm(ground,line01)

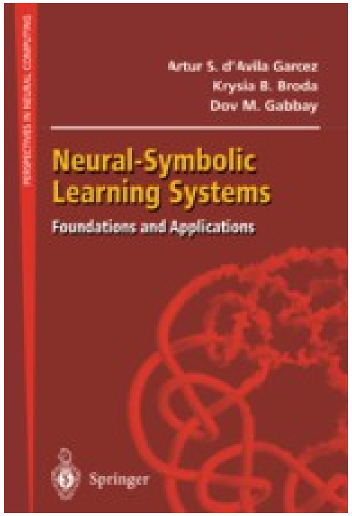
There is a fault at transmission line 01, close to the power plant generator, due to an over-current in the ground line of transmission line 01, which occurred when the system was not using the bypass circuit.

Power Plant Fault Diagnosis (results)

CILP achieves accuracy comparable to that of networks trained with backprop. or KBANN with the same BK, but it learns faster than both, and it performs better on smaller training sets (human-like computing?).

We attribute this to the soundness of the CILP translation (i.e. the above theorem; KBANN isn't provably sound).

For details: Garcez, Broda and Gabbay, Neural-Symbolic Learning Systems, Springer, 2002.



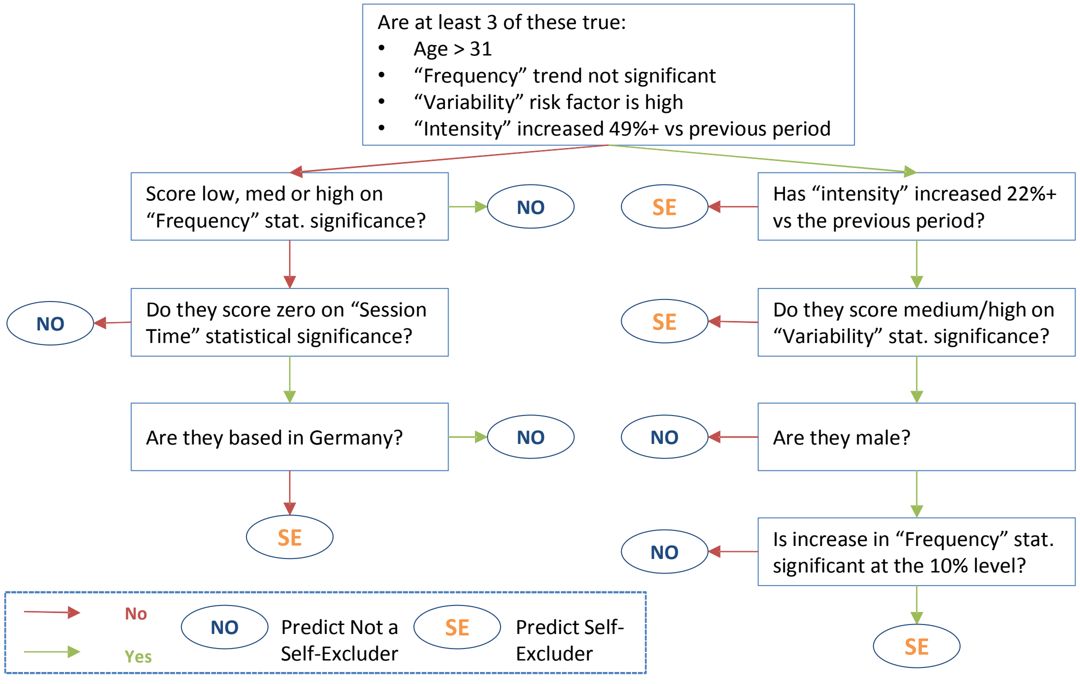
CILP Rule Extraction

- Knowledge is extracted by querying/sampling the trained network;
- A partial ordering helps guide the search, reducing complexity on the average case;
- A proof of soundness guarantees that the rules approximate the behaviour of the network;
- Rule simplification and visualization techniques help experts validate the rules;

CILP Rule Extraction (2)

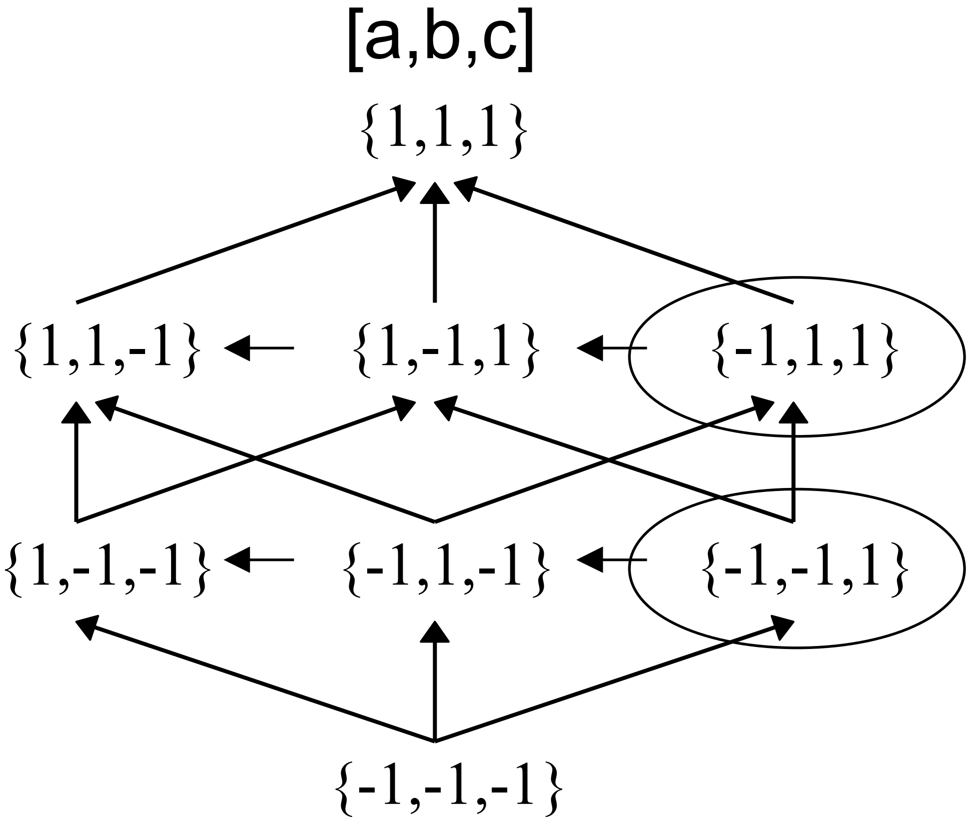
- Extracted rules can be visualized in the form of a state transition diagram (to follow)

- Alternatively, use (unsound but efficient) TREPAN-like rule extraction and variations of it...



C. Percy, A. S. d'Avila Garcez, S. Dragicevic, M. Franca, G. Slabaugh and T. Weyde. The Need for Knowledge Extraction: Understanding Harmful Gambling Behavior with Neural Networks, In Proc. ECAI 2016, The Hague, September 2016.

CILP Extraction Algorithm (discrete case)



$2(a, b, c) \rightarrow h_1$

$b, c \rightarrow h_1$
$a, c \rightarrow h_1$
$a, b \rightarrow h_1$

$a \rightarrow h_0$
$b \rightarrow h_0$
$c \rightarrow h_0$

$1\ (a, b, c) \rightarrow h_0$

THEOREM: CILP rule extraction is sound

Challenge: efficient extraction of sound, readable knowledge from large-scale networks (100's of neurons; 1000's of connections)

CILP extensions (richer knowledge)

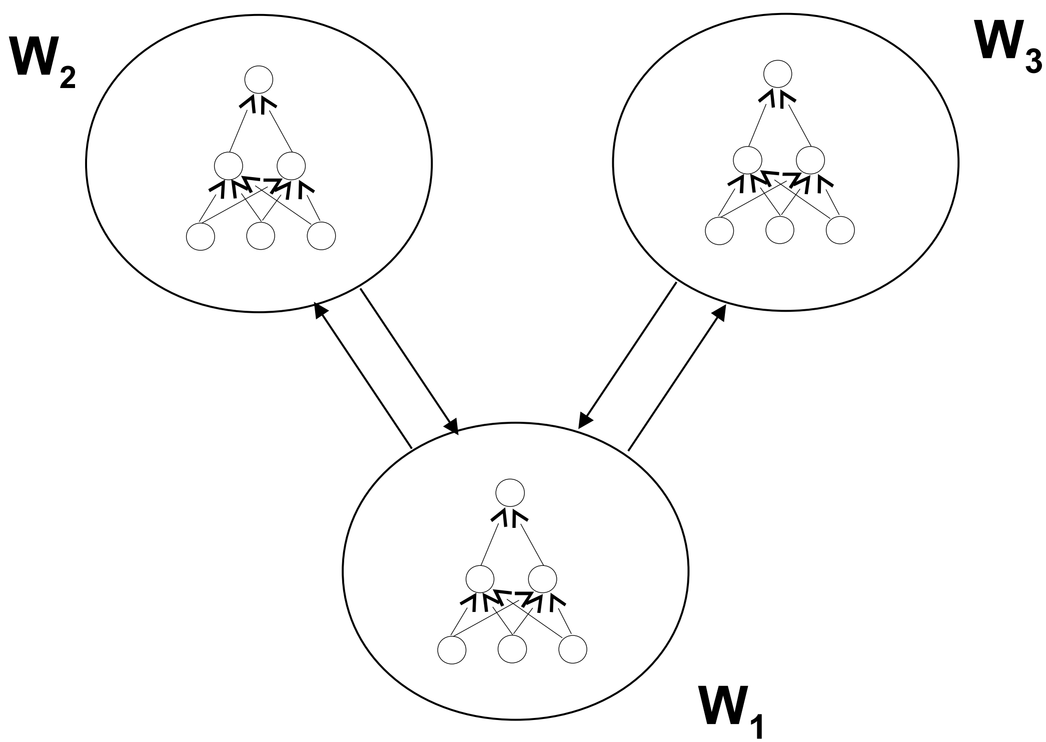
- The importance of non-classical reasoning: preferences, nonmonotonic, modal, temporal, epistemic, intuitionistic, abductive reasoning, value-based argumentation (dialogues), etc.

- New applications: normative reasoning (robocup), temporal logic learning with model checking, software model adaptation (business process evolution from text, e.g. email), training and assessment in simulators (driving test), visual intelligence (action classification in video), semantic web...

CILP network ensembles (deep structures)

Modularity for learning; accessibility relations for modal, temporal reasoning, disjunctive information, etc.

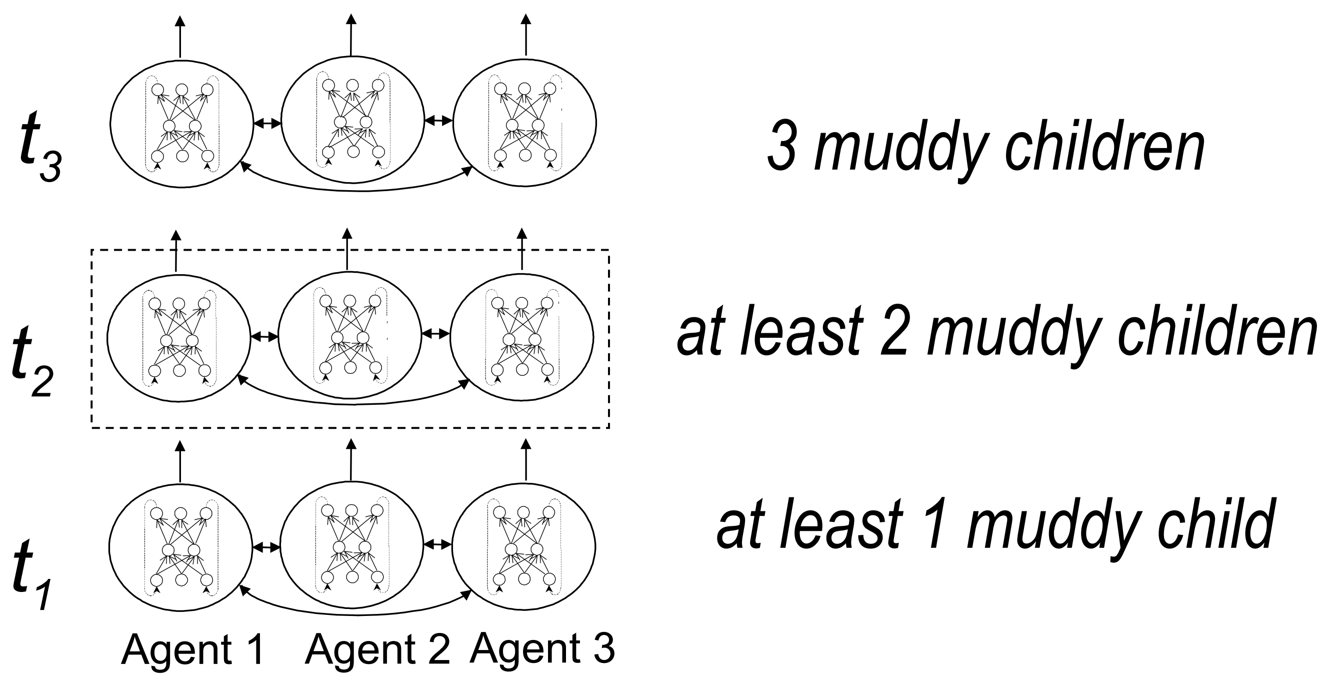
Good trade-off between expressiveness and computation



THEOREM: For any modal, temporal, epistemic, etc. program *P* there exists an ensemble of neural networks *N* such that *N* computes *P*.

Connectionist Temporal Reasoning and Learning

The muddy children puzzle (children are playing in a garden; at least one of them is muddy; they can see if the others are muddy, but not themselves; a caretaker asks: do you know if you're muddy?). A full solution to the puzzle can only be given by a two-dimensional network ensemble.



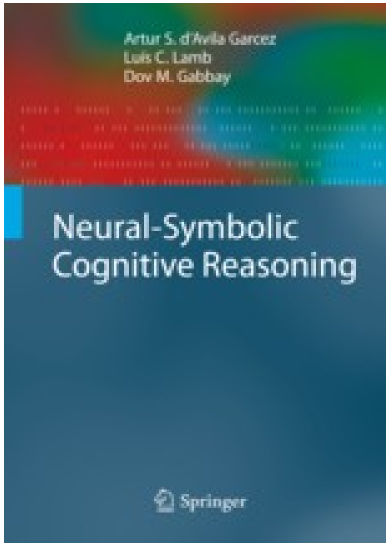
Learning with modal background knowledge is faster and offers better accuracy than learning by examples only (93% vs. 84% average test set accuracy)

Three wise men, kings and hats, etc...

- Various such logic puzzles and riddles can be useful at helping us understand the capabilities and limitations of neural models

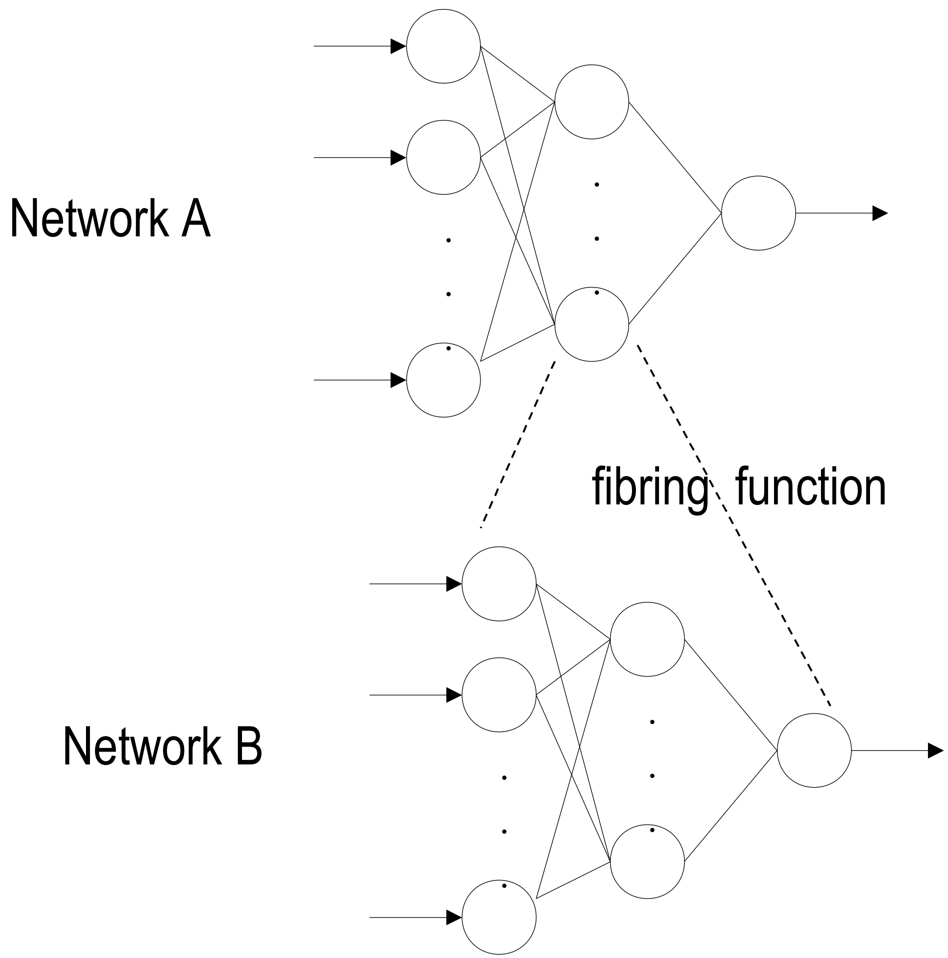
A certain king wishes to test his three wise men. He arranges them in a circle so that they can see and hear each other. They are all perceptive, truthful and intelligent, and this is common knowledge in the group. It is also common knowledge among them that there are three red hats and two white hats, and five hats in total. The king places a hat on the head of each wise man in a way that they are not able to see the colour of their own hats, and then asks each one whether they know the colour of the hats on their heads.

For details: Garcez, Lamb and Gabbay, Neural-Symbolic Cognitive Reasoning, Springer, 2009.



Combining (Fibring) Networks

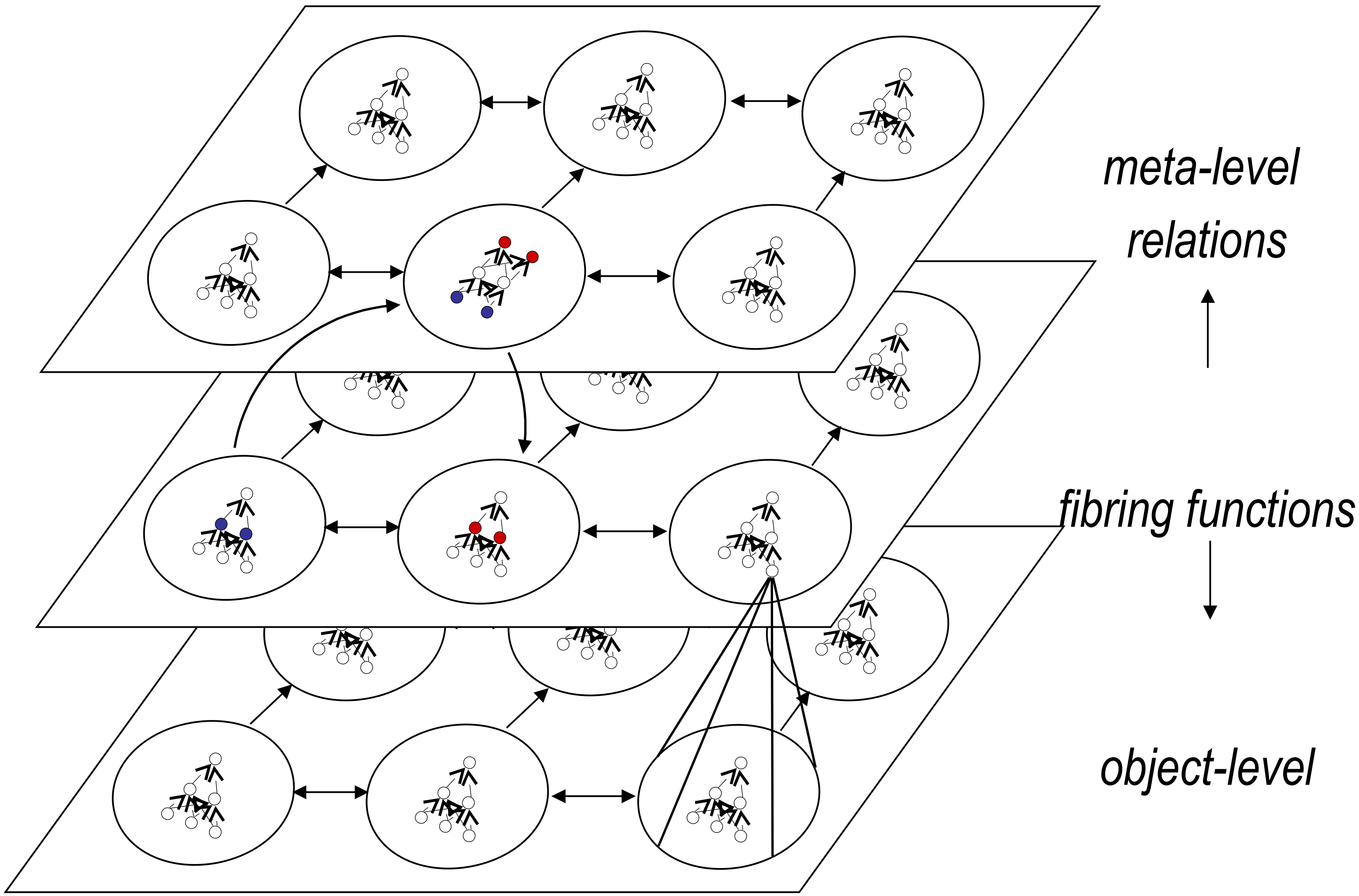
A neuron that is a network! Neuromodulation?



Expressiveness to represent first-order logic, i.e. relational knowledge

Garcez and Gabbay, Fibring Neural Networks, In Proc. AAAI 2004

CILP Cognitive Model: Fibred Network Ensembles



Applications (1)

Training and Assessment in Simulators

• Learning from observation of experts and trainees at task execution, and reasoning online to provide feedback to the user

• System seeks to adapt in real-time to the skills of the user, whether an experienced driver or a learner.

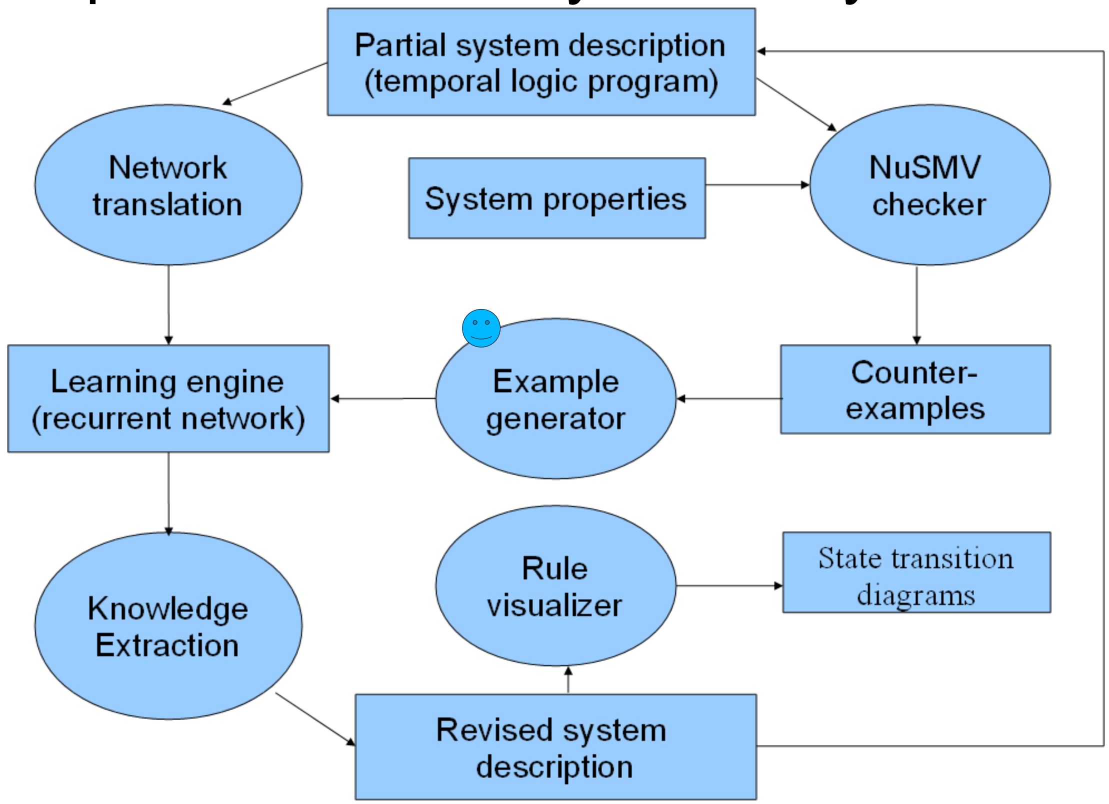
• To do so, it uses temporal knowledge insertion and extraction from stacks of RBMs and RTRBMs

L. de Penning, A. d'Avila Garcez, L. Lamb and J. J. Meyer. A Neural-Symbolic Cognitive Agent for Online Learning and Reasoning. IJCAI'11, July 2011

Applications (2)

Software Model Verification and Adaptation

Verification: NuSMV
Adaptation: Neural-Symbolic System



Borges, Garcez, Lamb. Learning and Representing Temporal Knowledge in Recurrent Networks. IEEE TNN 22(12):2409 - 2421, Dec 2011.

See also: F. Vaandrager, Model learning, CACM, Feb 2017.

V&A applied to Pump System

The pump system controls the levels of water in a mine to avoid the risk of overflow; an initial, partial system description is available.

State variables: *CrMeth* (level of methane is critical)
HiWat (level of water is high)
PumpOn (pump is turned on)

Safety property in LTL: $G\neg(CrMeth \wedge HiWat \wedge PumpOn)$

Partial system spec (background knowledge; s = sensor):

$$\bigcirc CrMeth \leftarrow sCMOn.$$
$$\bigcirc CrMeth \leftarrow CrMeth, \sim sCMOff.$$
$$\bigcirc HiWat \leftarrow sHiW.$$
$$\bigcirc HiWat \leftarrow CrMeth, \sim sLoW.$$
$$\bigcirc PumpOn \leftarrow TurnPOn.$$
$$\bigcirc PumpOn \leftarrow CrMeth, \sim TurnPOff.$$

Verification (NuSMV) and example generation

New Counter-example		
t	State	Input
1	$\{\sim CrMeth, \sim HiWat \sim PumpOn\}$	$sCMOn$
2	$\{CrMeth, \sim HiWat, \sim PumpOn\}$	$TurnPOn$
3	$\{CrMeth, \sim HiWat, PumpOn\}$	$sHiW$
4	$\{CrMeth, HiWat, PumpOn\}$	–

A training example:
sCMOn → *TurnPOn* → *sHiW* → *¬PumpOn*

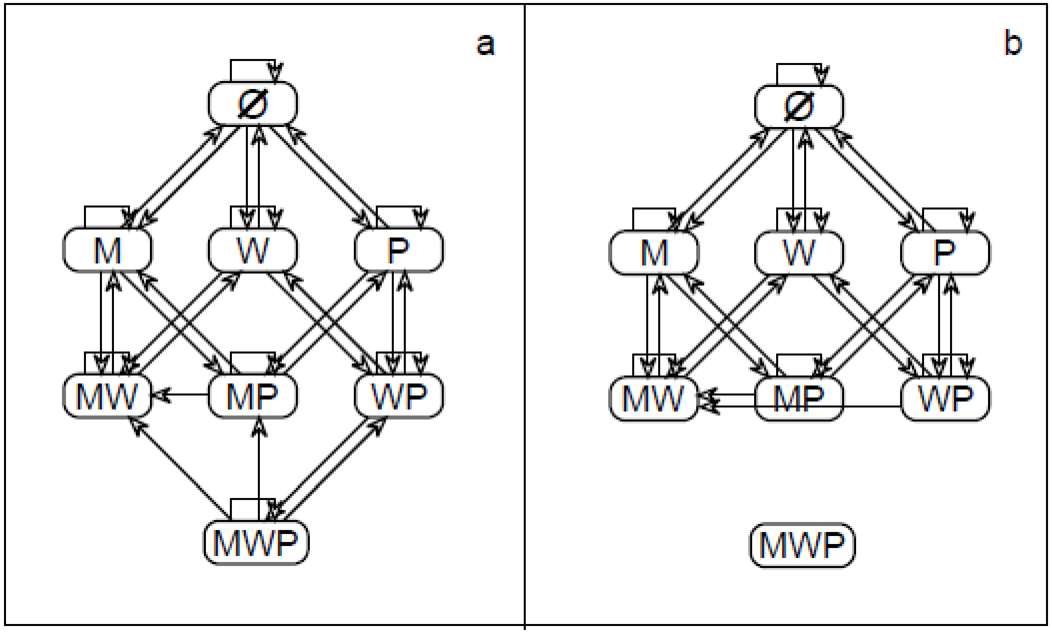
Corresponding to new rule: If methane is critical then turn the pump on, unless the water level is high...

Repeat the process until the property is (hopefully) satisfied (i.e. no counter-example is generated)

Neural network is three-valued {-1,0,1} CILP network, similar to NARX, trained with standard backprop.

Network Visualization

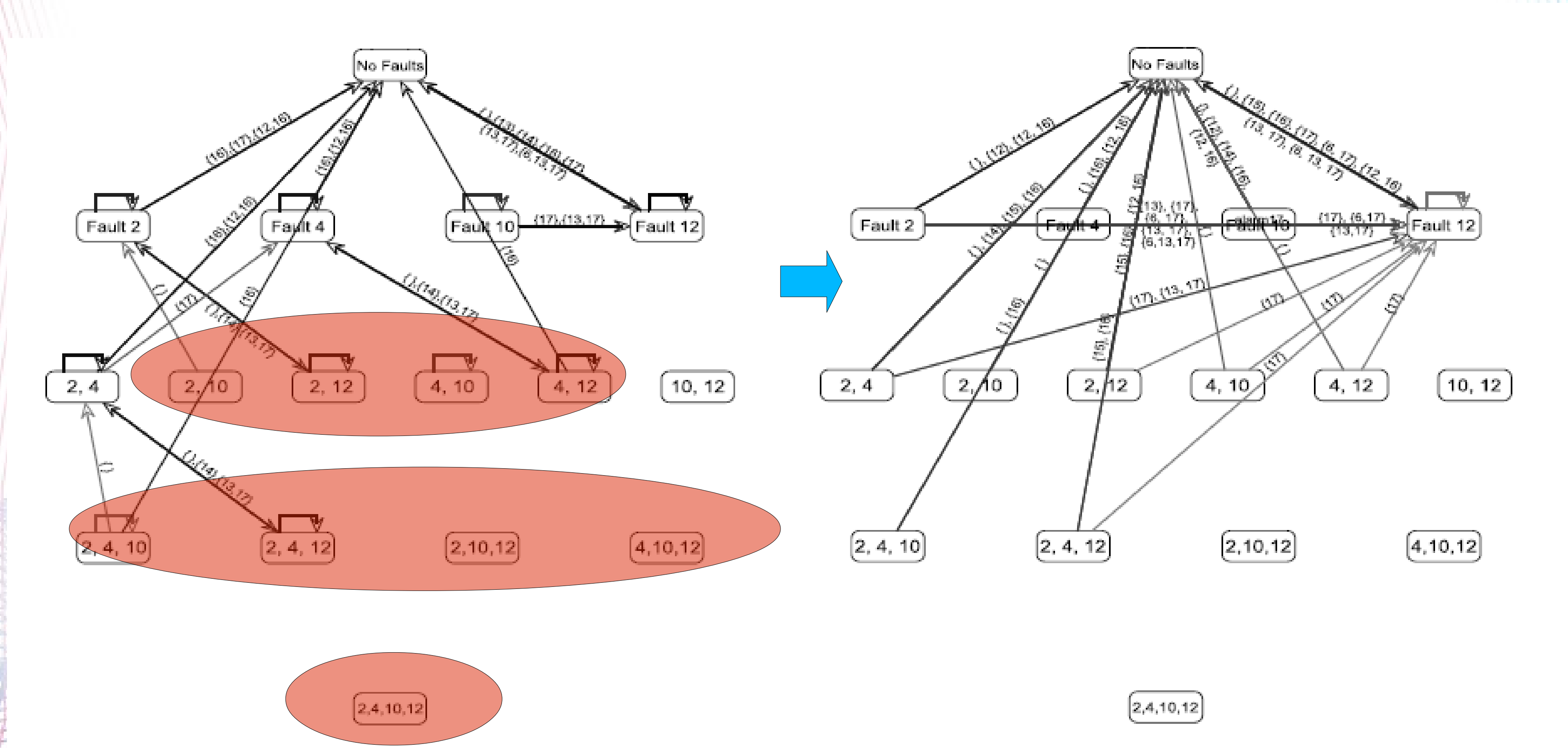
CrMeth = M (level of methane is critical)
HiWat = W (level of water is high)
PumpOn = P (pump is turned on)



Power Plant Fault Diagnosis (real problem; ongoing validation)

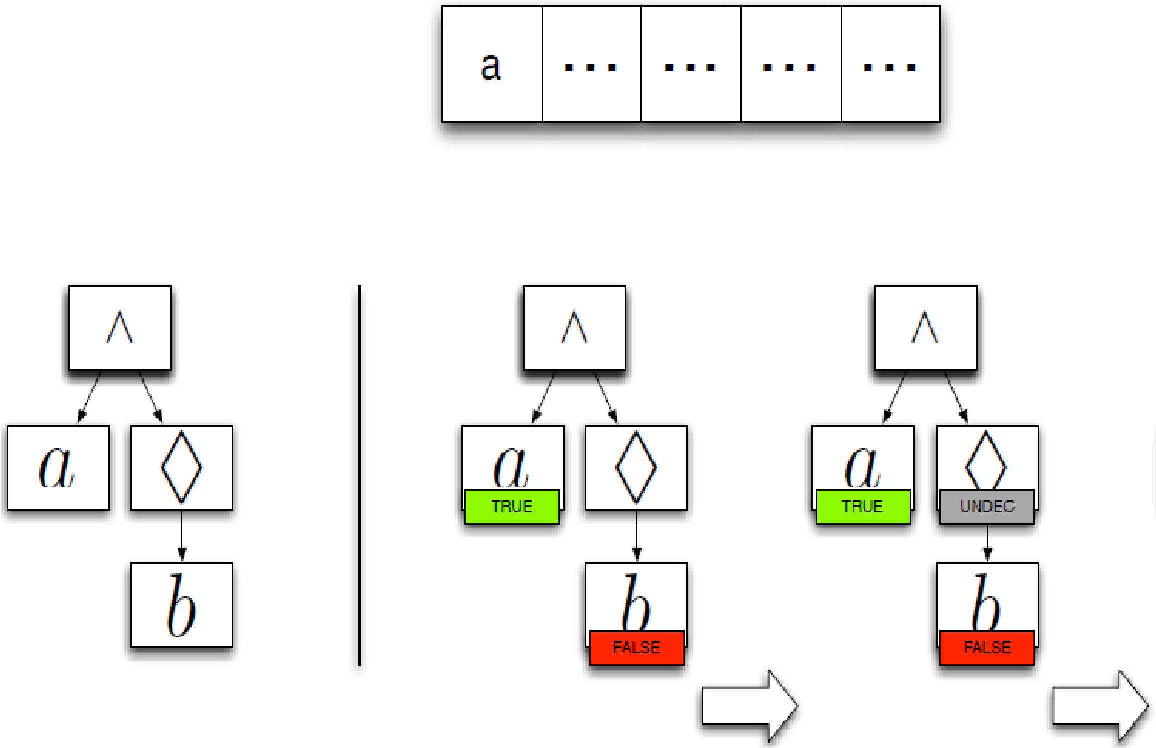
Safety property: $G\neg(Fault(\_,\_,line1,bypass) \wedge Fault(\_,\_,line2,bypass))$

(diagrams are annotated with alarms which trigger derived faults)



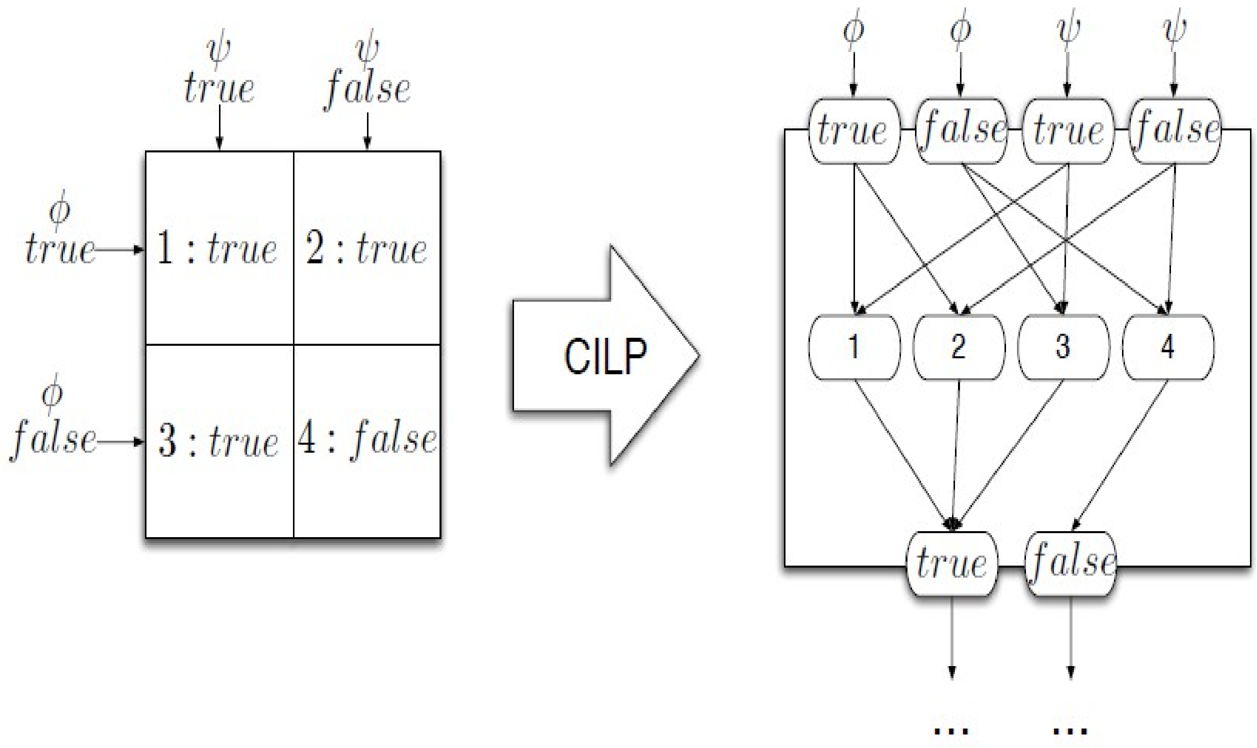
Run-time Monitoring

- So far, LTL property is outside the neural net
- Let's consider property adaptation next.



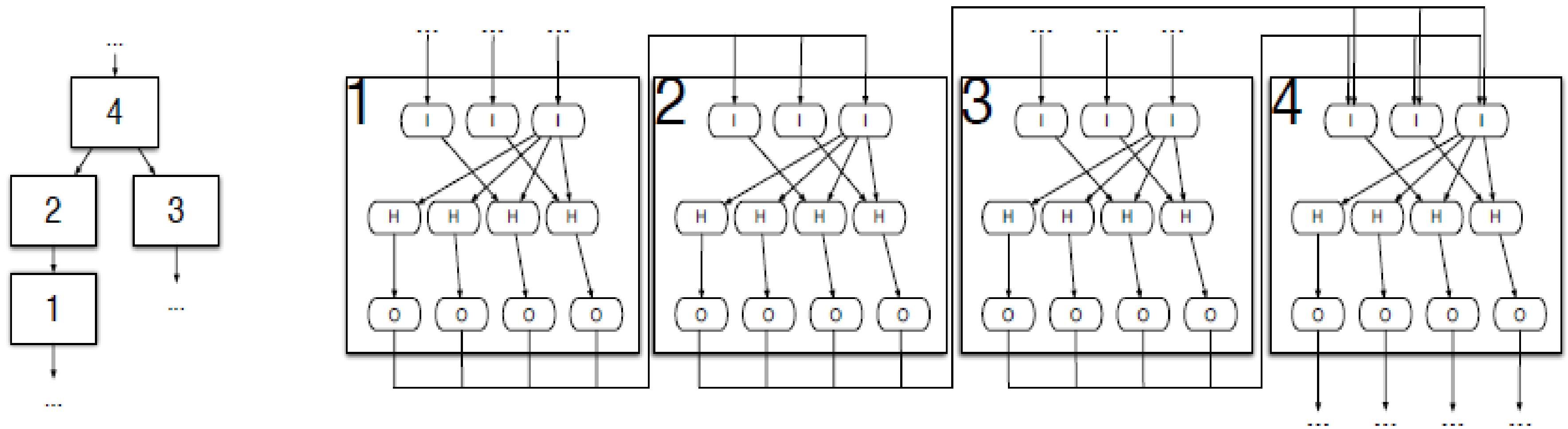
Neural Encoding

- Every tree node implements a truth-table for one operator
- Every truth-table can be represented in a CILP neural net



Run-Time Neural Monitor

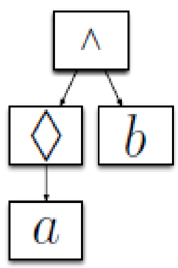
- The tree structure is “flattened” into an ensemble of CILP networks

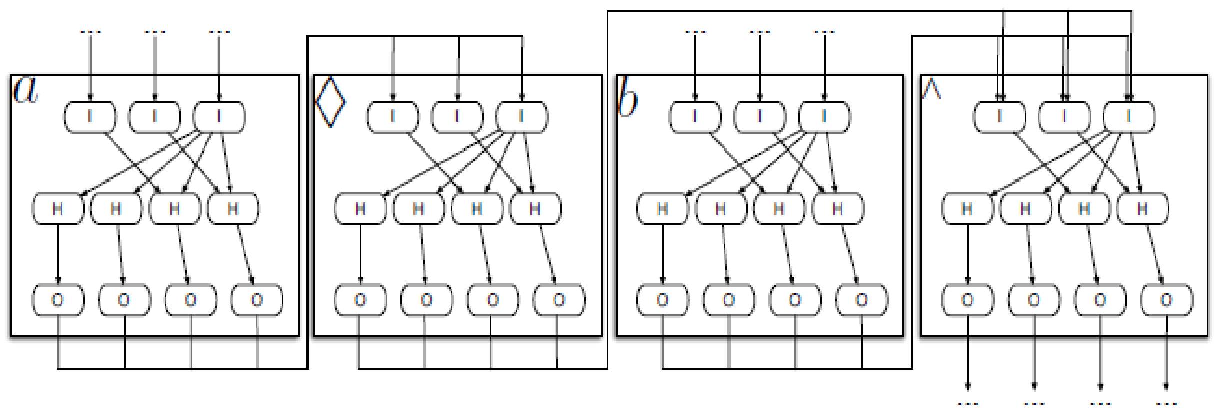


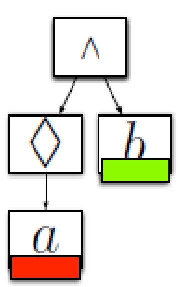
Property monitoring

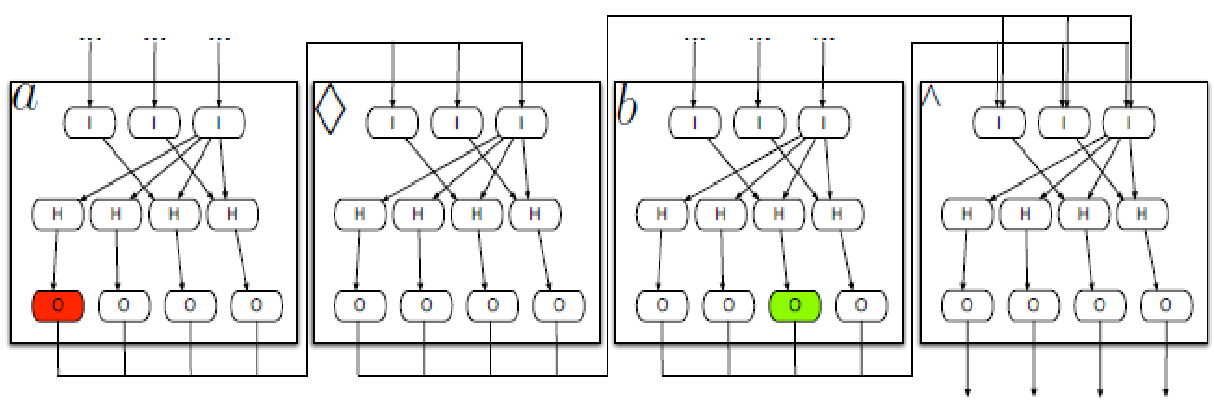
$\Diamond a \wedge b$

b	...	...	...	...

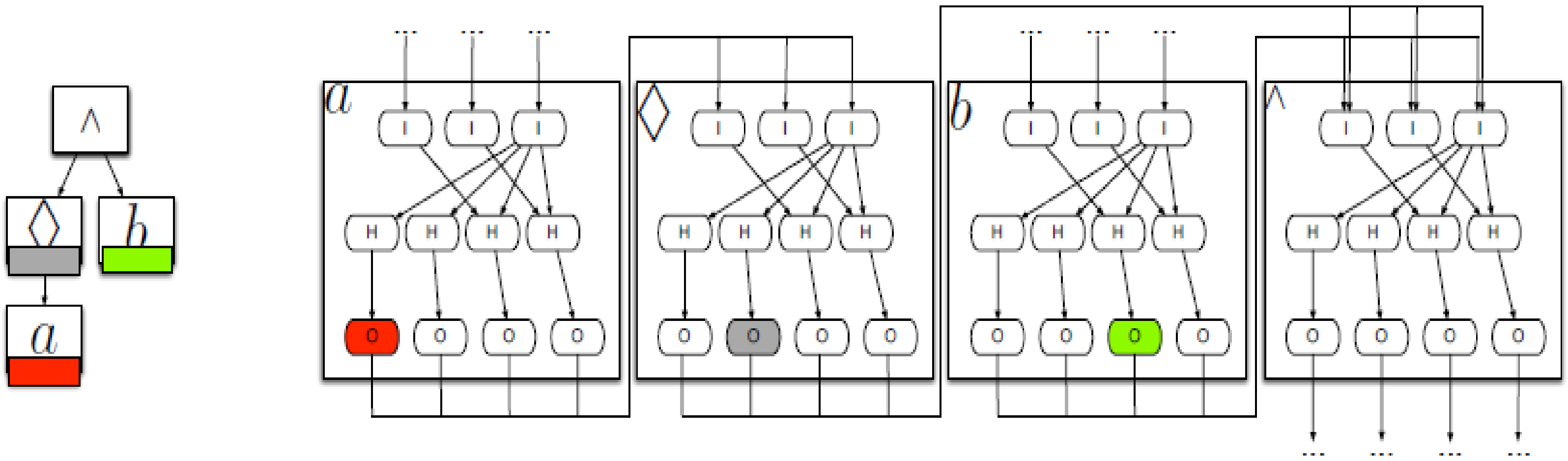


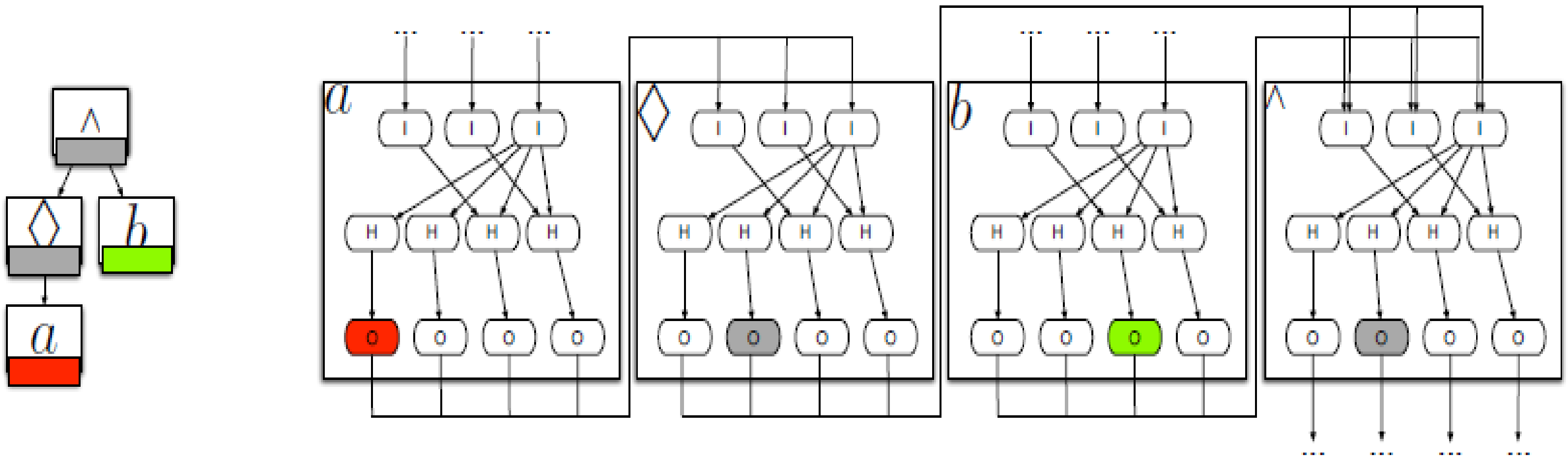






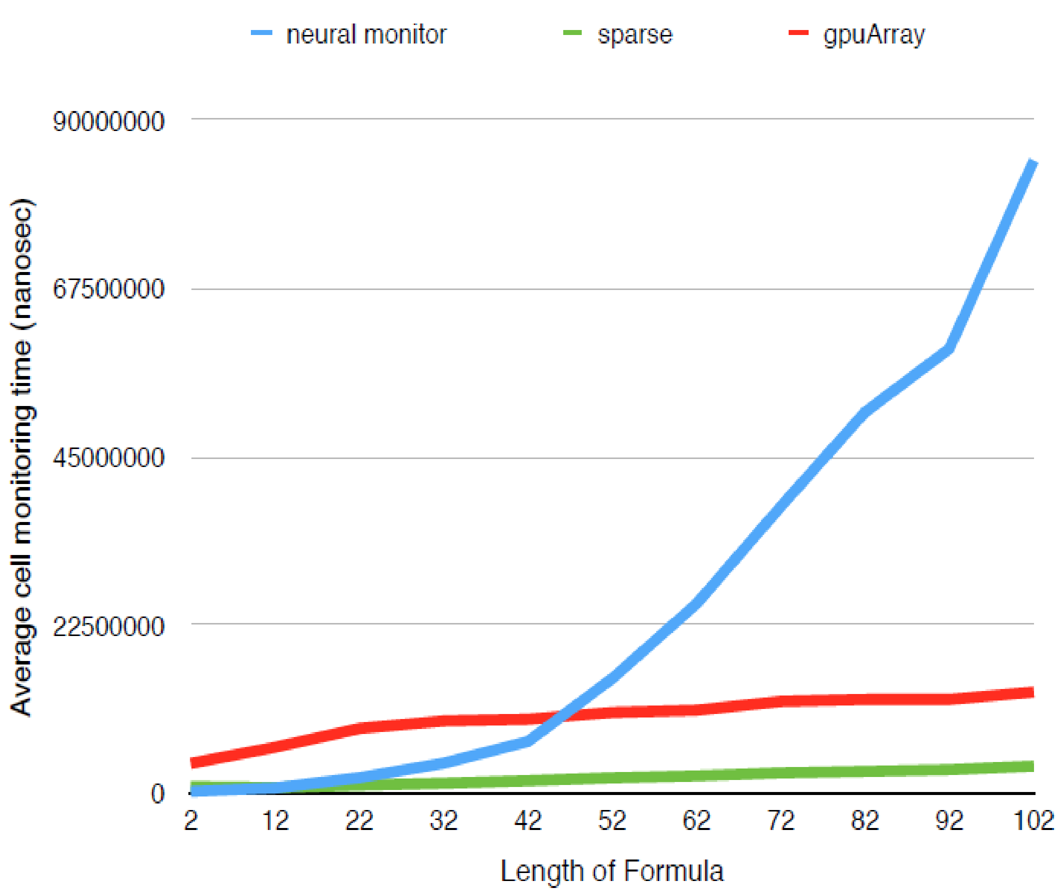
Monitor verdict = stable output





Performance

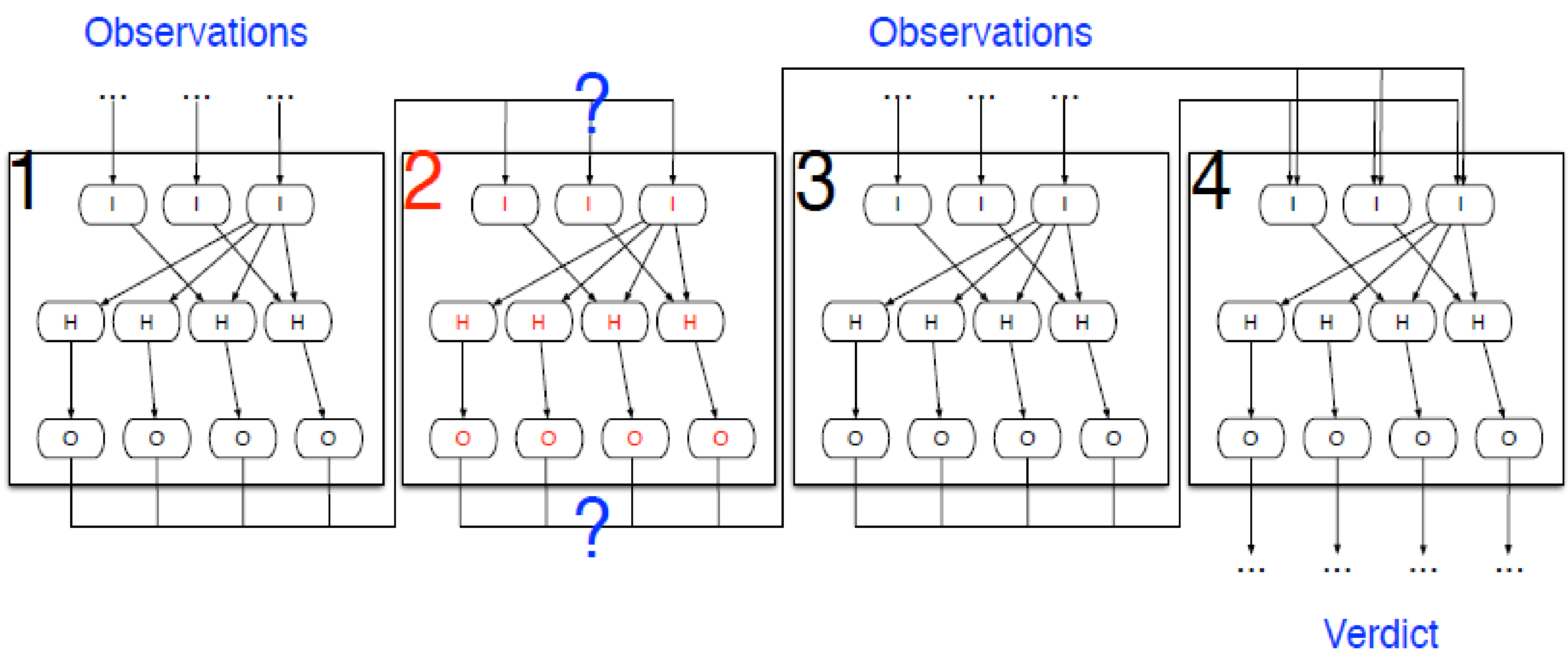
Bottleneck is matrix multiplication. Matrix growth is quadratic w.r.t. length of property. But matrices are sparse (with constant number of non-zero elements per row; tree branching factor is constant)



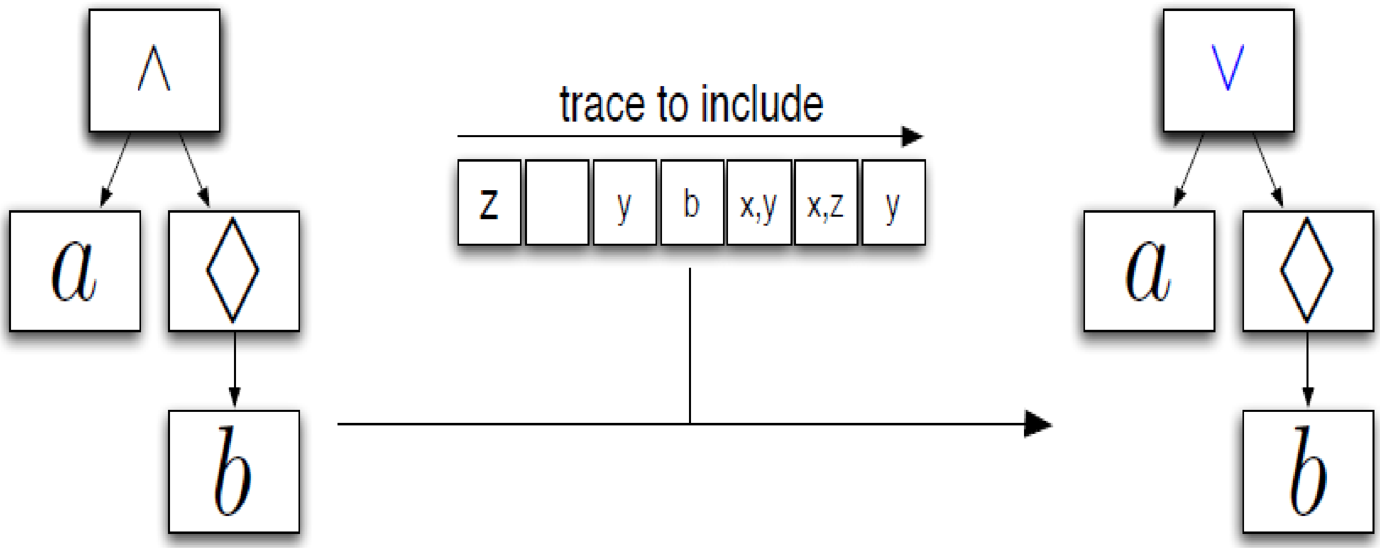
Learning = property adaptation

Local Training

Propagate from observations to verdict and backpropagate label to abduce local input-output patterns (e.g. for network 2).



Adaptation: bending the rules



A. Perotti, G. Boella and A. S. d'Avila Garcez, Runtime Verification Through Forward Chaining. In Proc. RV'15, September 2015.

A. Perotti, A. S. d'Avila Garcez and Guido Boella. Neural-Symbolic Monitoring and Adaptation. In Proc. IJCNN 2015, July 2015.

Current/Future Work

- So far:
  - Loose integration of NNs and Model Checking
  - NNs for modelling/adapting system properties
- Current: Verification of trained networks used e.g. for controller synthesis
  - c.f. Katz et al. Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks (Feb 2017) (Extension of Simplex to ReLUs)
- Next:
  - Tight integration of NNs and Model Checking? (a model checker that is a neural net)

Recent developments in Neural-Symbolic Computing

- Knowledge Extraction from Deep Nets:
  - S. Tran and A. S. d'Avila Garcez. Deep Logic Networks: Inserting and Extracting Knowledge from Deep Belief Networks. IEEE TNNLS, Nov, 2016
- Relational (full FOL) Learning in Tensor Networks (with Tensorflow implementation):
  - L. Serafini, I. Donadello and A. S. d'Avila Garcez. Learning and Reasoning in Logic Tensor Networks: Theory and Application to Semantic Image Interpretation. In ACM SAC 2017, April 2017.
- Applications of knowledge extraction in industry: understanding pathways to harm in gambling and reducing harm from gambling (c.f. BetBuddy.com)
- Neural-Symbolic Computing in sports, education, health and work: Illuminoo.com

Conclusion: Why Neurons and Symbols

To study the statistical nature of learning and the logical nature of reasoning.

To provide a unifying foundation for robust learning and efficient reasoning.

To develop effective computational systems for integrated reasoning and learning.

Thank you!