NeuroSymbolic Artificial Intelligence

Between probability and fuzzyness

Luciano Serafini

Fondazione Bruno Kessler
Trento, Italy

November 7, 2023

Table of Contents

FONDAZIONE BRUNO KESSLER
FUTURE BUILT ON KNOWLEDGE
Future Artificial Intelligence Research

Introduction

Symbolic Knowledge Representation

Agents's knowledge is represented by logical theories (set of sentences of a logical language) stating properties about relational structures

- declarative semantics gives a meaning to each sentence, supporting compositionality $\Longrightarrow$ reasoning on logical consequence
- appropriate to represent knowledge about structurally intricated domains, which require complex combinatorial reasoning.

Introduction

Symbolic Knowledge Representation

Agents's knowledge is represented by logical theories (set of sentences of a logical language) stating properties about relational structures

- declarative semantics gives a meaning to each sentence, supporting compositionality $\Longrightarrow$ reasoning on logical consequence
- appropriate to represent knowledge about structurally intricated domains, which require complex combinatorial reasoning.

SubSymbolic Knowledge Representation

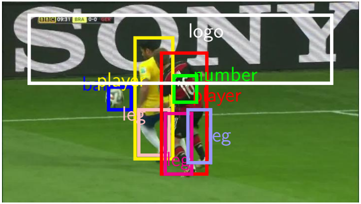
Agent's knowledge is represented by means of real (high complex) functions (e.g., regression, classification, clustering, generative models).

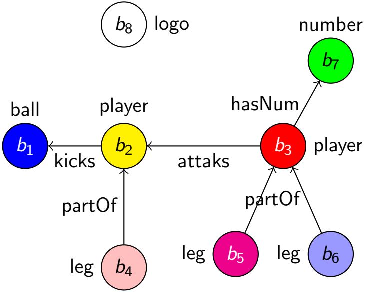
- differentiability w.r.t, a set of parameters $\Longrightarrow$ automatic learning by gradient-based optimization.
- appropriate represents knowledge about regularities on large and continuous dataspaces.

Structured Domains with Continuous Data

- Semantic image understanding with background knowledge

$$\forall x, y(kicks(x, y) \rightarrow player(x) \wedge ball(y))$$
$$\forall xy(color(jersy(x), z) \wedge color(jersy(y), z) \rightarrow team(x) = team(y))$$





I. Donadello, L. Serafini, and A. S. d'Avila Garcez (2017). "Logic Tensor Networks for Semantic Image Interpretation". In: *IJCAI*, pp. 1596–1602

Structured Domains with Continuous Data

- Neuro-Sybolic verification of deep neural networks
  - robustness to adversarial attack

$$\boldsymbol{x} \approx \boldsymbol{y} \rightarrow N_{traffic\_light}(\boldsymbol{x} \mid \boldsymbol{\theta}) \approx N_{traffic\_light}(\boldsymbol{y} \mid \boldsymbol{\theta})$$

  - Fairness: the output of a neural network N is not influenced by a sensitive feature (e.g., gender)

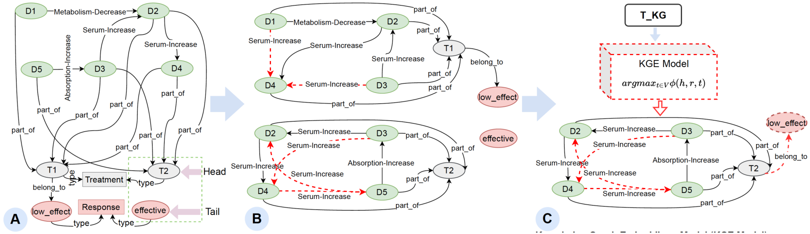
$$\boldsymbol{x}_{-gender} = \boldsymbol{y}_{-gender} \rightarrow N_{loan}(\boldsymbol{x} \mid \boldsymbol{\theta}) = N_{loan}(\boldsymbol{y} \mid \boldsymbol{\theta})$$

Neuro-Symbolic Assertion Language inspired Hoare logic [Hoare, 1969]

Xuan Xie, Kristian Kersting, and Daniel Neider (2022). "Neuro-Symbolic Verification of Deep Neural Networks". In: *Thirty-First International Joint Conference on Artificial Intelligence*. IJCAI, pp. 3622–3628

Structured Domains with Continuous Data

- Generate knowledge graph embeddings that are consistent with background ontologies



A Rivas et al. (2023). "A Neuro-Symbolic System over Knowledge Graphs for Link Prediction". In: *Semantic Web*

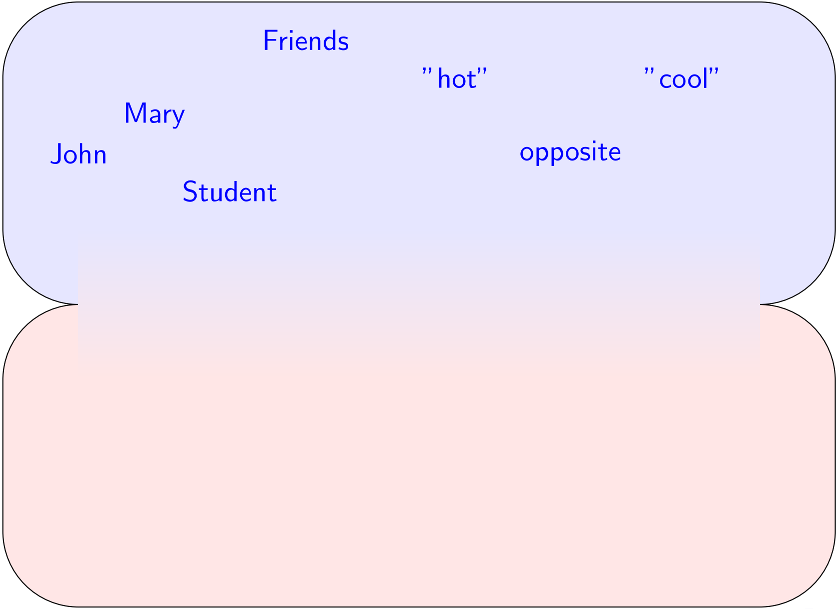
Table of Contents

Grounding symbols to reals

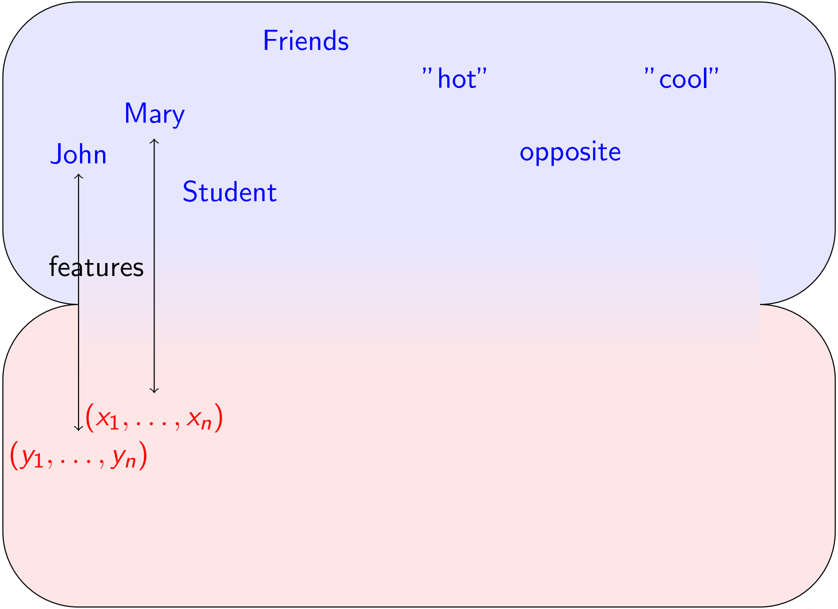
- an agents sees the external world via perceptions
- learned real functions associated to each perception an internal representation (class, cluster, embedding, . . .) of the observed phenomena
- we could use real functions to solve the grounding problem by . . .
- **grounding each internal symbol to a real function**[1]

[1] A constant value is 0-ary real function

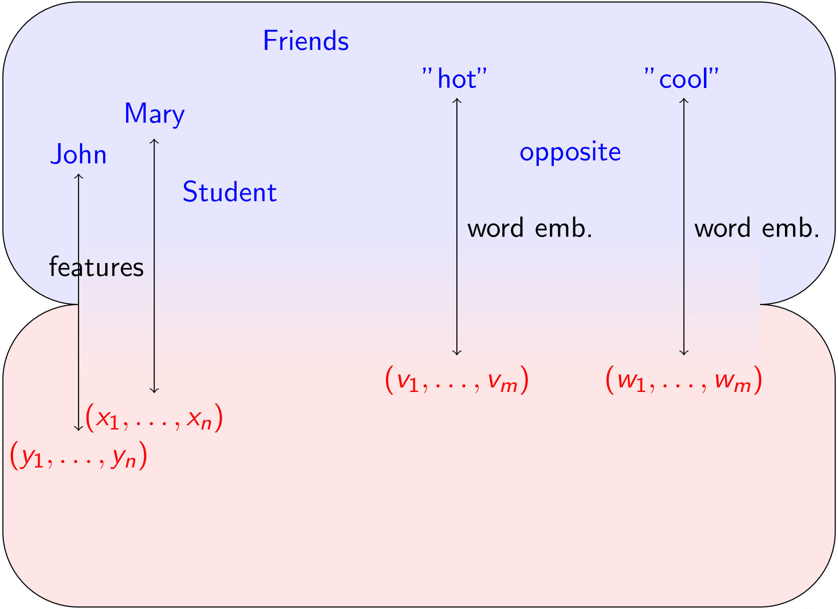
Grounding symbols to reals



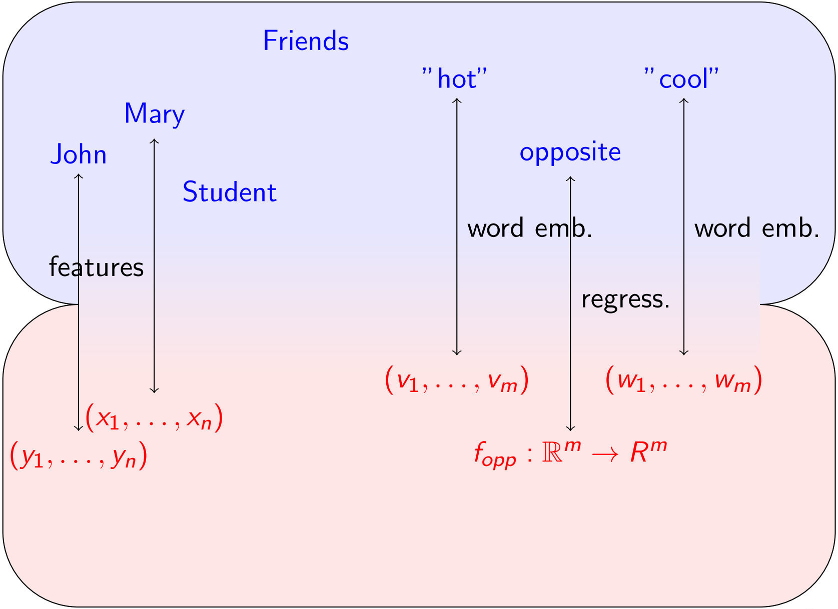
Grounding symbols to reals



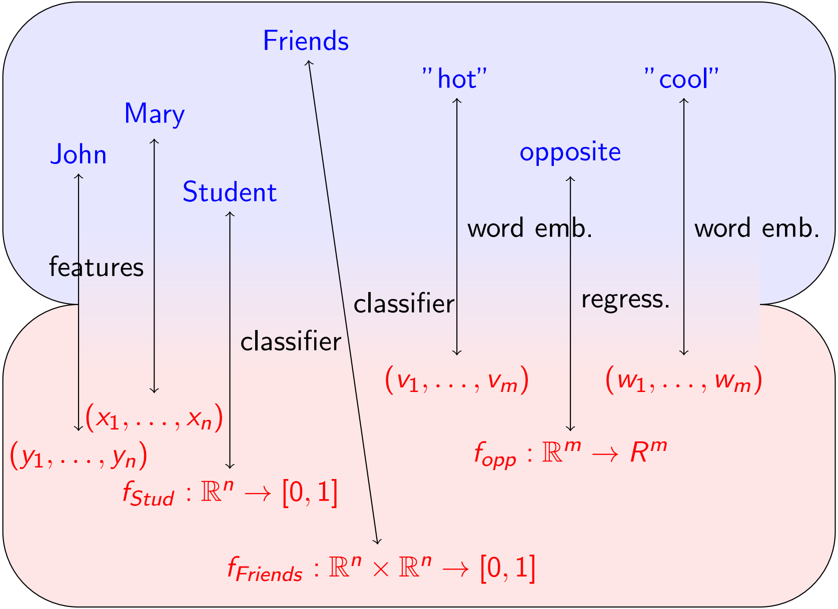
Grounding symbols to reals



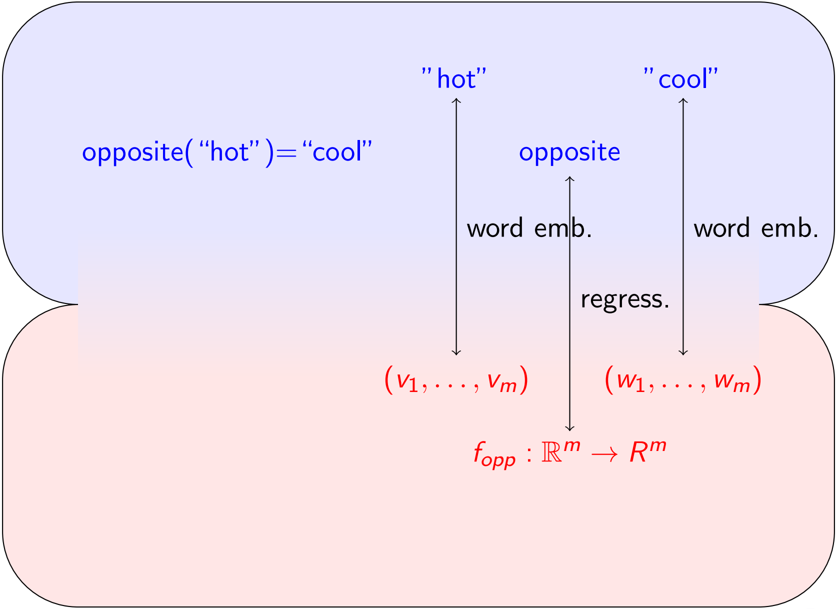
Grounding symbols to reals



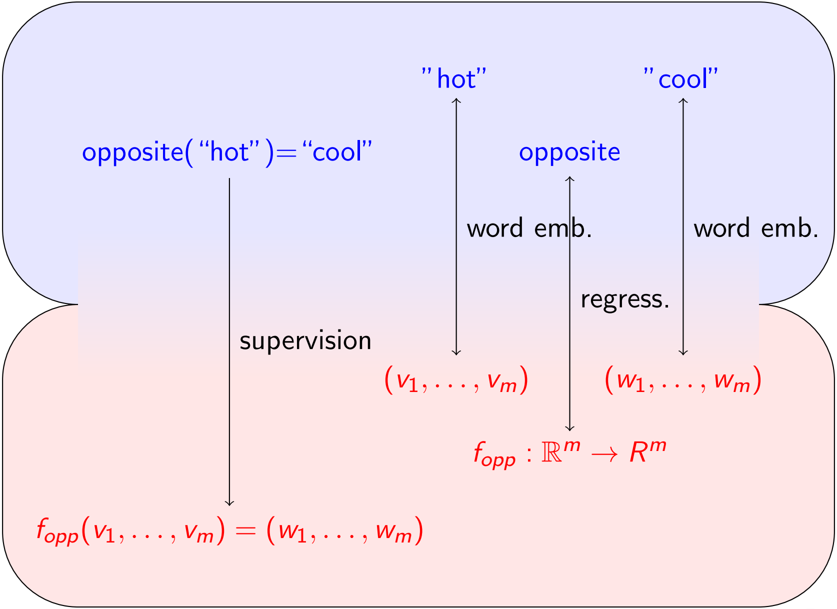
Grounding symbols to reals



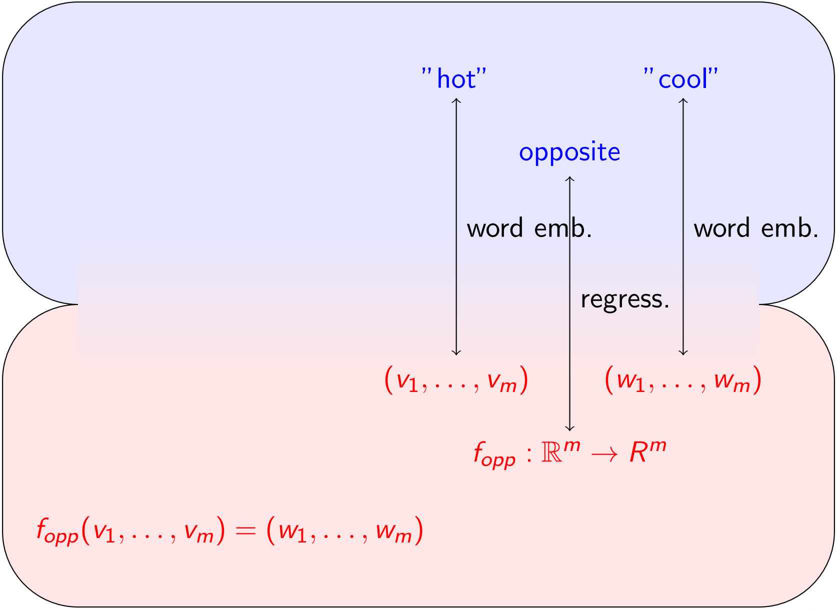
Propagating knowledge via Sim. Gr.



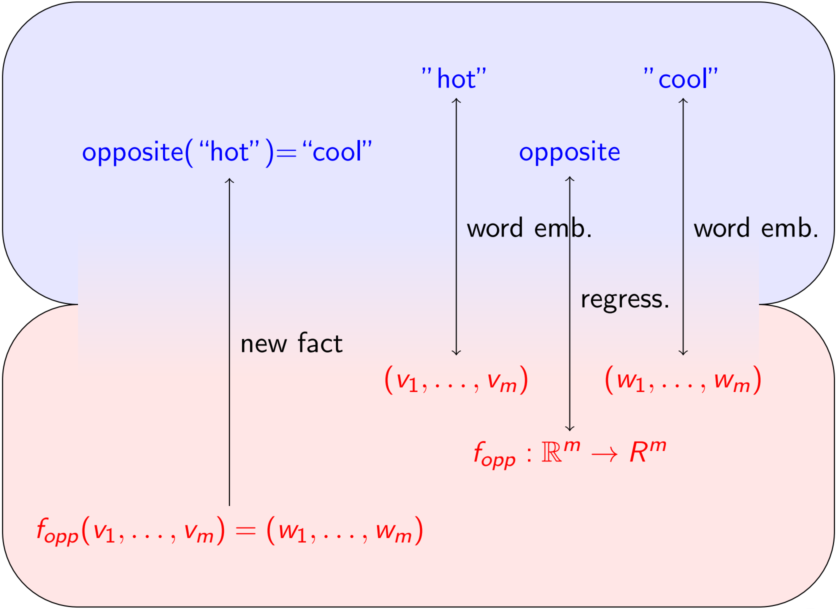
Propagating knowledge via Sim. Gr.



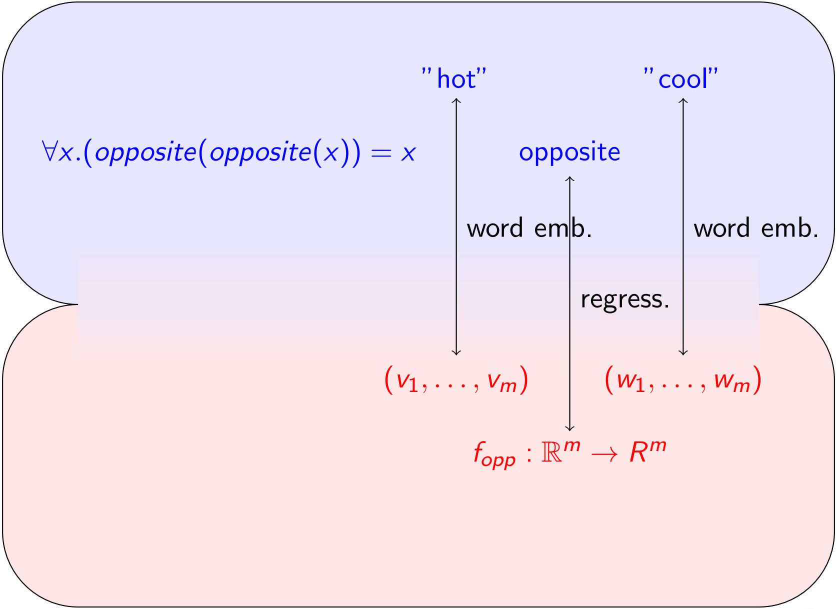
Propagating knowledge via Sim. Gr.



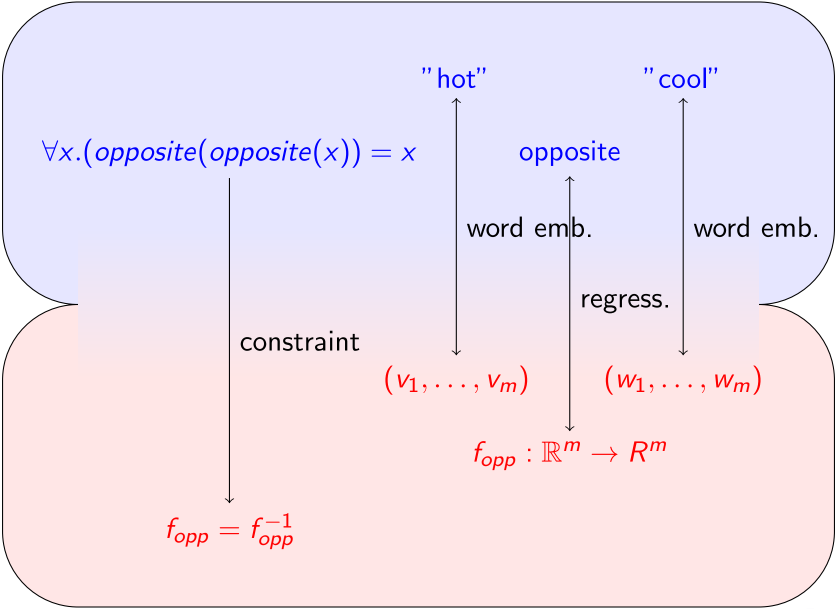
Propagating knowledge via Sim. Gr.



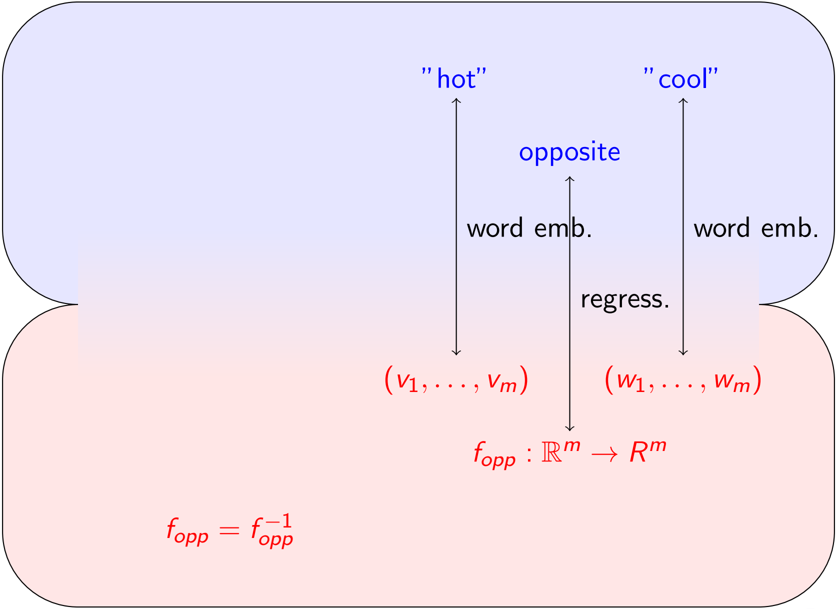
Propagating knowledge via Sim. Gr.



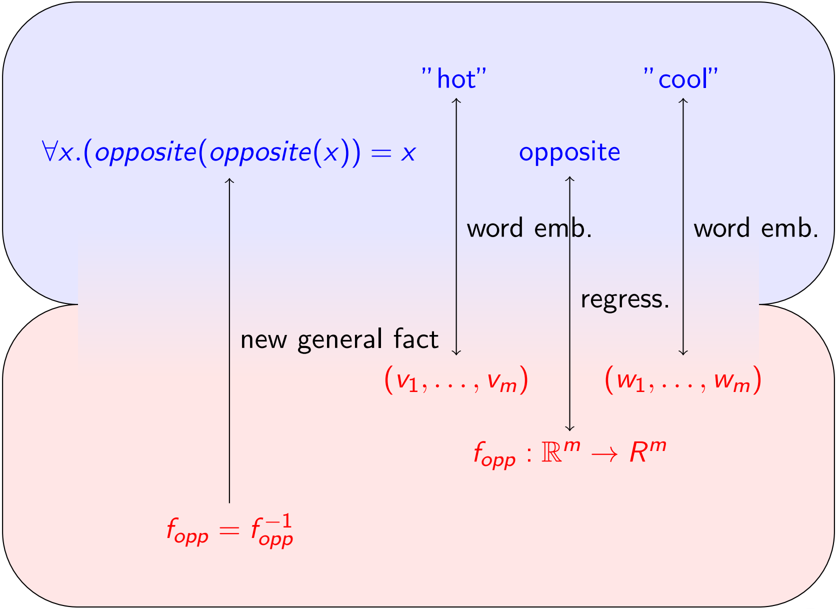
Propagating knowledge via Sim. Gr.



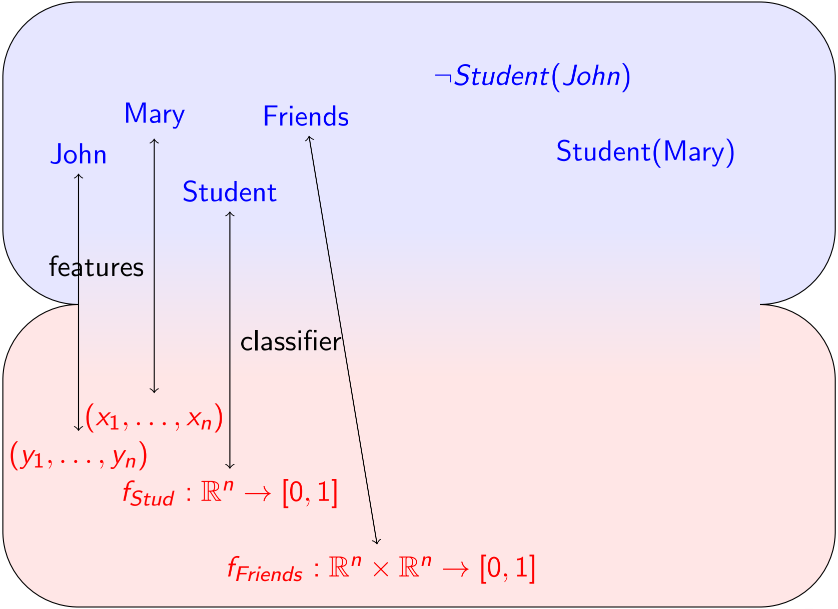
Propagating knowledge via Sim. Gr.



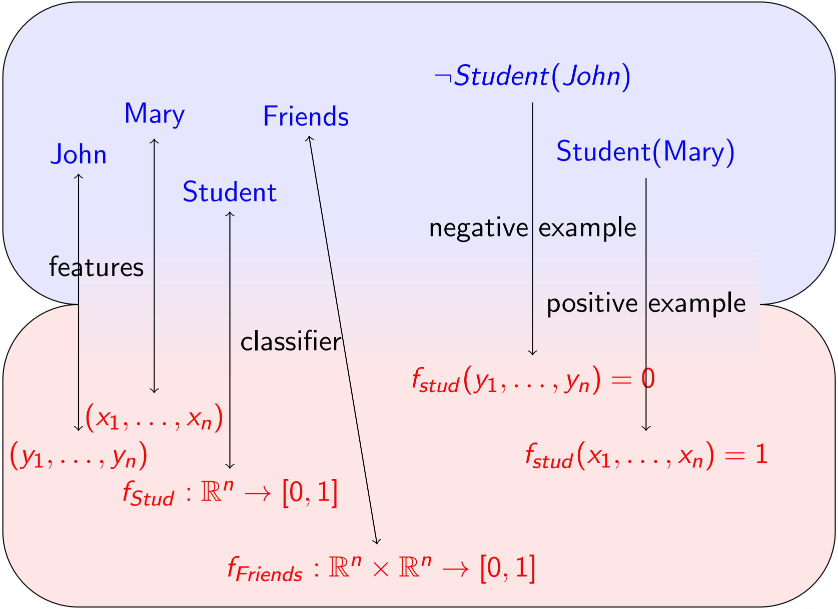
Propagating knowledge via Sim. Gr.



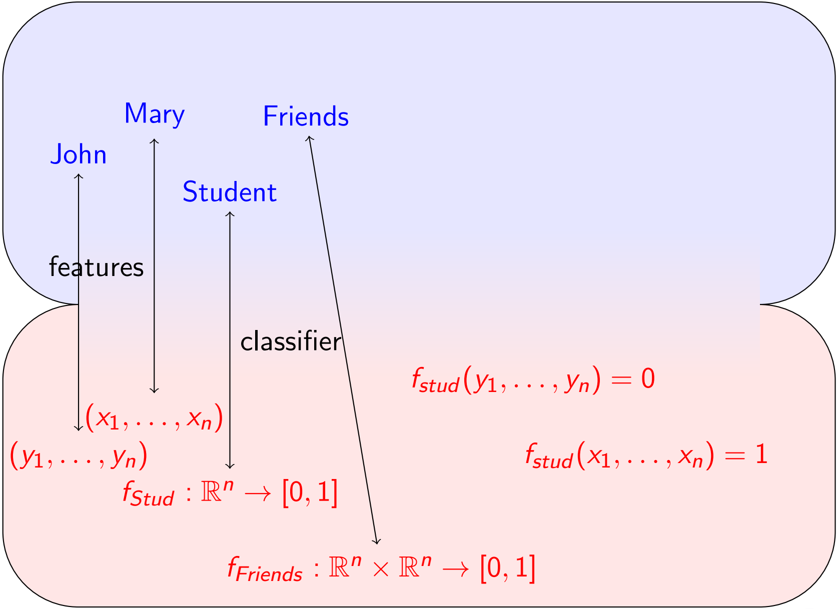
Propagating knowledge via Sim. Gr.



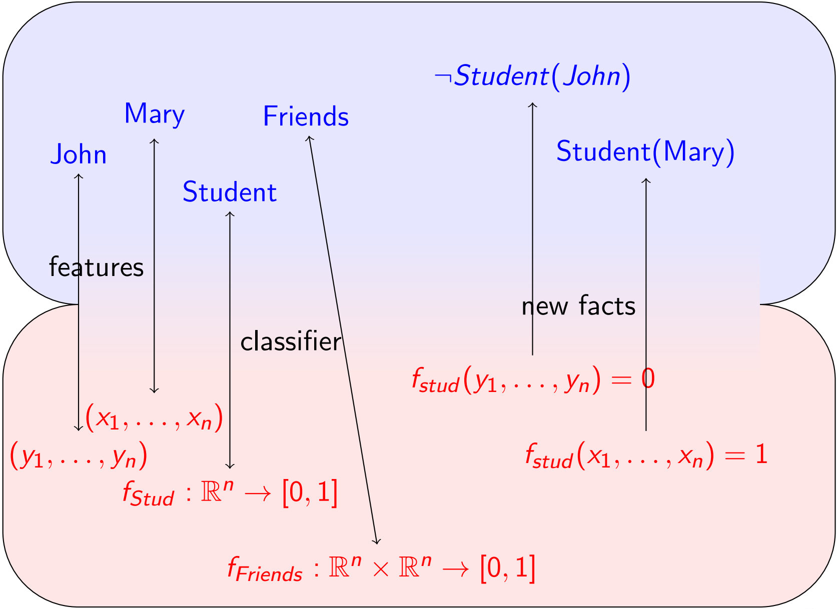
Propagating knowledge via Sim. Gr.



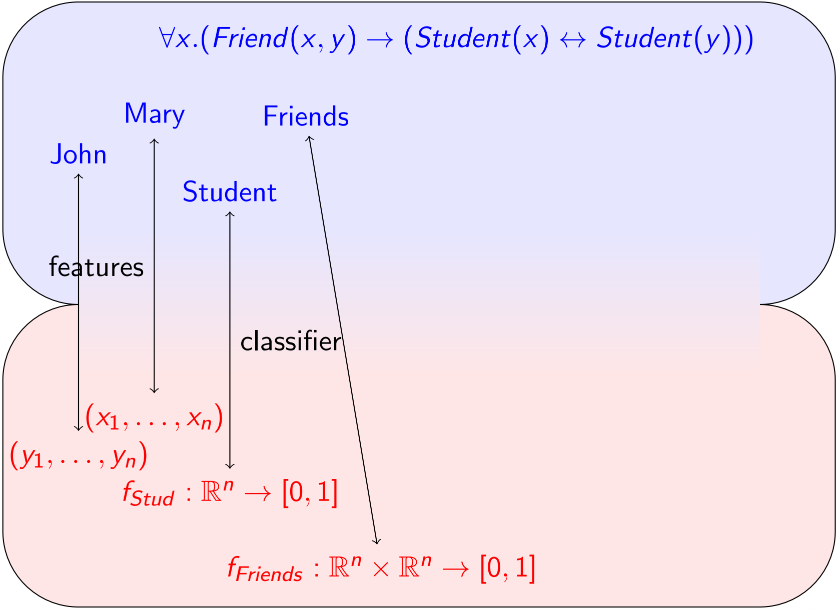
Propagating knowledge via Sim. Gr.



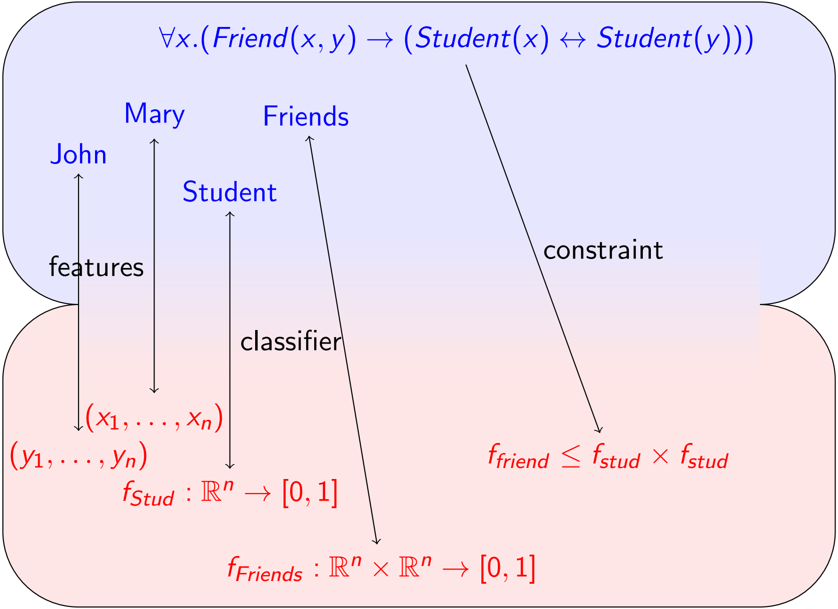
Propagating knowledge via Sim. Gr.



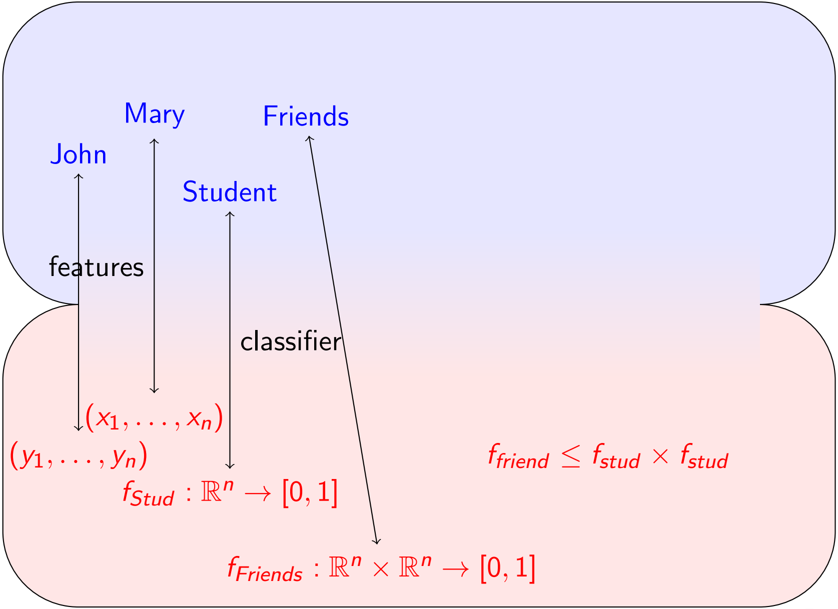
Propagating knowledge via Sim. Gr.



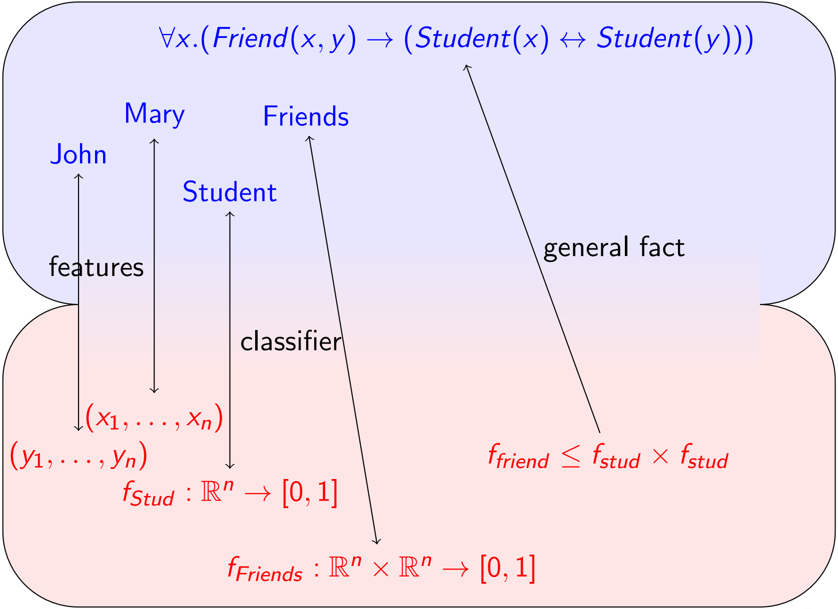
Propagating knowledge via Sim. Gr.



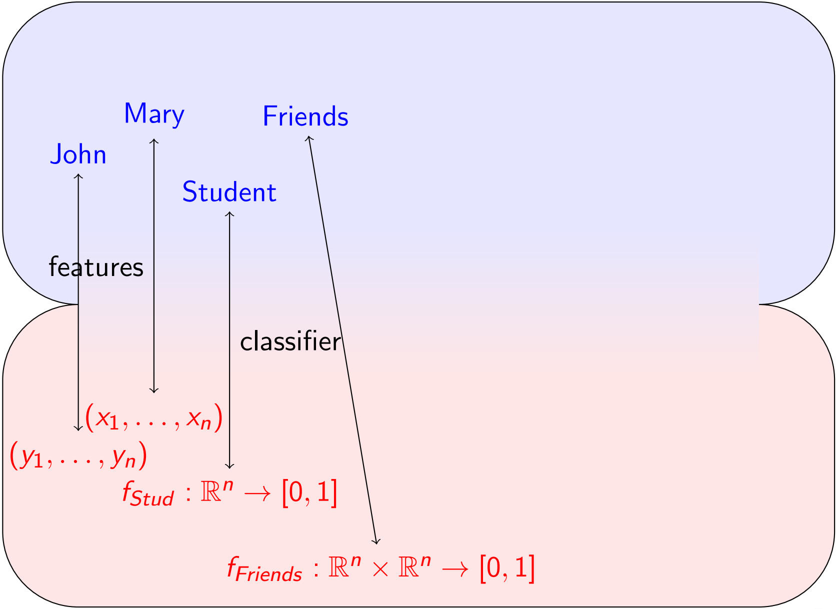
Propagating knowledge via Sim. Gr.



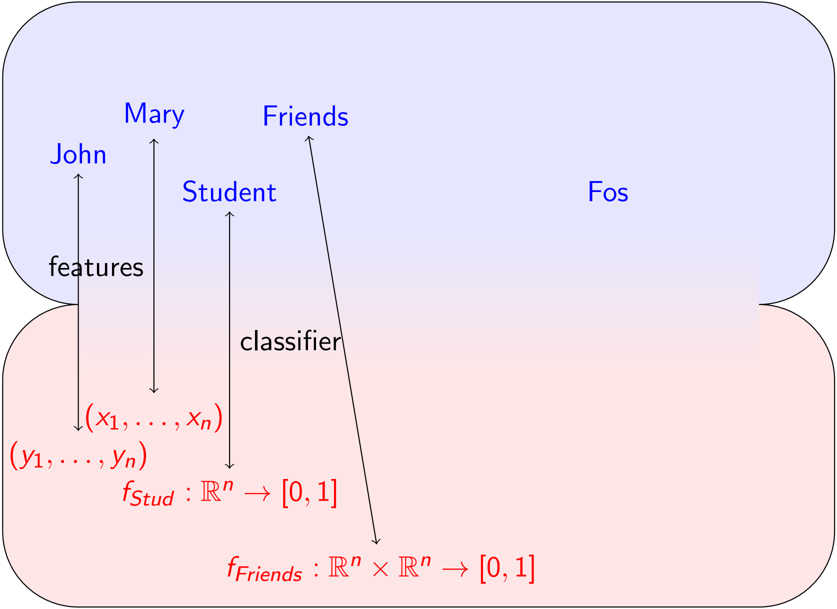
Propagating knowledge via Sim. Gr.



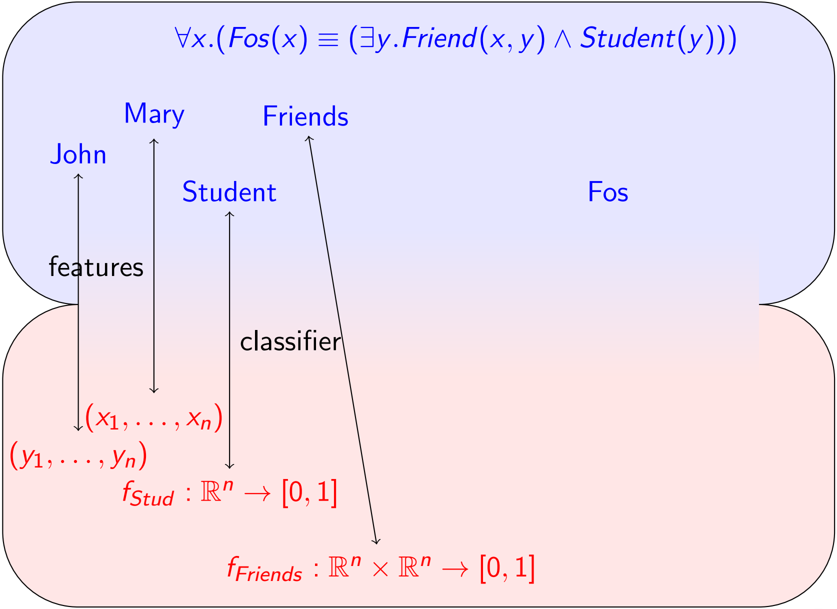
Propagating knowledge via Sim. Gr.



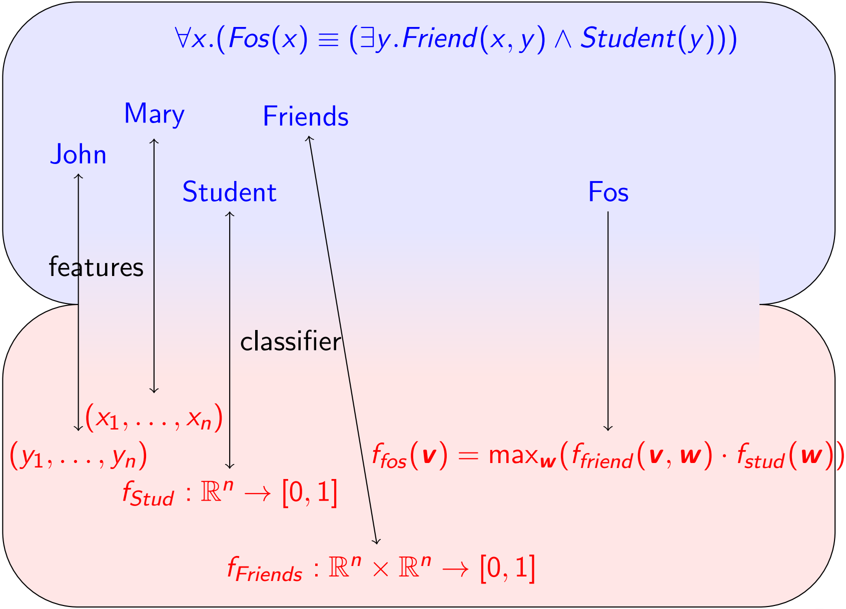
Propagating knowledge via Sim. Gr.



Propagating knowledge via Sim. Gr.



Propagating knowledge via Sim. Gr.



NeuroSymbolic AI

- Neurally inspired models represent knowledge in terms of real valued models. From a very abstract viewpoint a neural model of a parametric function

$$f(\cdot \mid \boldsymbol{\theta}_f) : S_i(\mathbb{R}) \longrightarrow S_0(\mathbb{R}) \qquad \boldsymbol{\theta}_f \in \boldsymbol{\Theta}_f$$

  where $S_i(\mathbb{R})$ and $S_o(\mathbb{R})$ is structured data on real numbers, which are the input and output of the model $f(\cdot \mid \boldsymbol{\theta}_f)$.

NeuroSymbolic AI

- Neurally inspired models represent knowledge in terms of real valued models. From a very abstract viewpoint a neural model of a parametric function

$$f(\cdot \mid \boldsymbol{\theta}_f) : S_i(\mathbb{R}) \longrightarrow S_0(\mathbb{R}) \qquad \boldsymbol{\theta}_f \in \boldsymbol{\Theta}_f$$

  where $S_i(\mathbb{R})$ and $S_o(\mathbb{R})$ is structured data on real numbers, which are the input and output of the model $f(\cdot \mid \boldsymbol{\theta}_f)$.
- symbolic models Represent knowledge by means of a theory which is a set of sentences of a logical language in a signature Σ, closed under logical consequence:

$$T \subset \mathcal{L}(\Sigma)$$

NeuroSymbolic AI

- Neurally inspired models represent knowledge in terms of real valued models. From a very abstract viewpoint a neural model of a parametric function

$$f(\cdot \mid \boldsymbol{\theta}_f) : S_i(\mathbb{R}) \longrightarrow S_0(\mathbb{R}) \qquad \boldsymbol{\theta}_f \in \boldsymbol{\Theta}_f$$

  where $S_i(\mathbb{R})$ and $S_o(\mathbb{R})$ is structured data on real numbers, which are the input and output of the model $f(\cdot \mid \boldsymbol{\theta}_f)$.
- symbolic models Represent knowledge by means of a theory which is a set of sentences of a logical language in a signature Σ, closed under logical consequence:

$$T \subset \mathcal{L}(\Sigma)$$

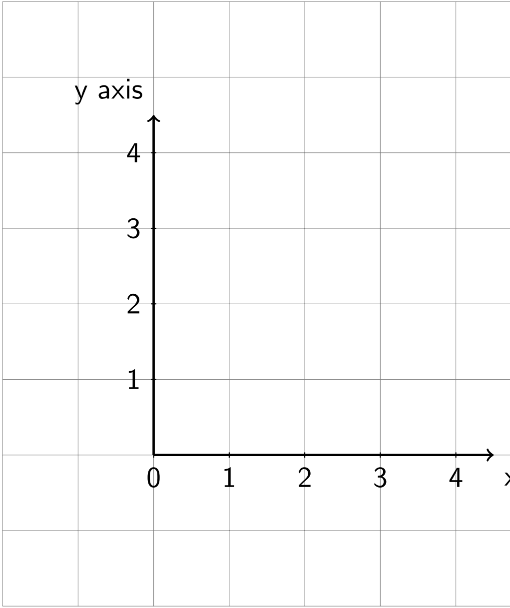
- Neuro-Symbolic models combine neural models and symbolic models by interpreting (some of) the symbols $\sigma \in \Sigma$ is a neural model

$$\langle T, \mathcal{G} \rangle \qquad T \subseteq \mathcal{L}(\Sigma) \qquad \mathcal{G} : \sigma \mapsto f_\sigma(\cdot \mid \boldsymbol{\theta}_\sigma) \qquad \sigma \in \Sigma$$

Neuro Symbolic tasks

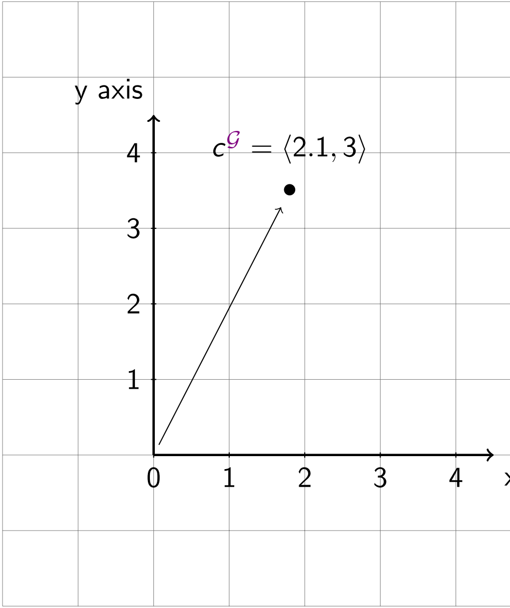
- **learning grounding** Given some knowledge about a set of symbols Σ, learn the groundings of some or all symbolis in Σ.
  - in ML this is analogous to training
  - in KR this is analogous to maximum satisfiability
- **answering queries** Given some knolwedge about a set of grounded symbols Σ, determine the truth value of a query ϕ.
  - in ML and KR this is called inference.

Grounding FOL Signature in $\mathbb{R}^2$



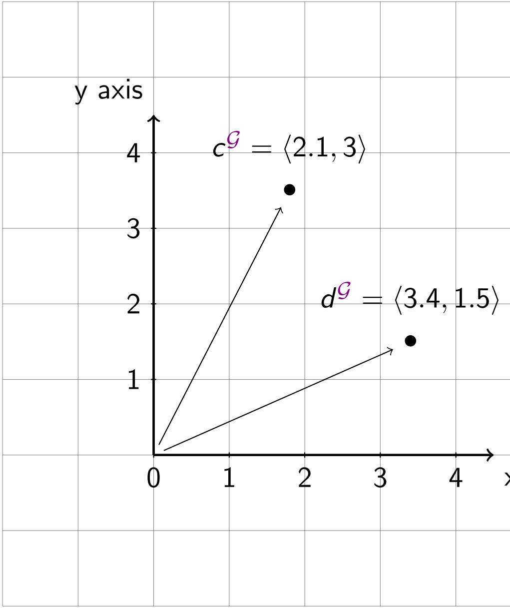
Grounding FOL Signature in $\mathbb{R}^2$

- $c^{\mathcal{G}} = \langle 2.1, 3 \rangle$



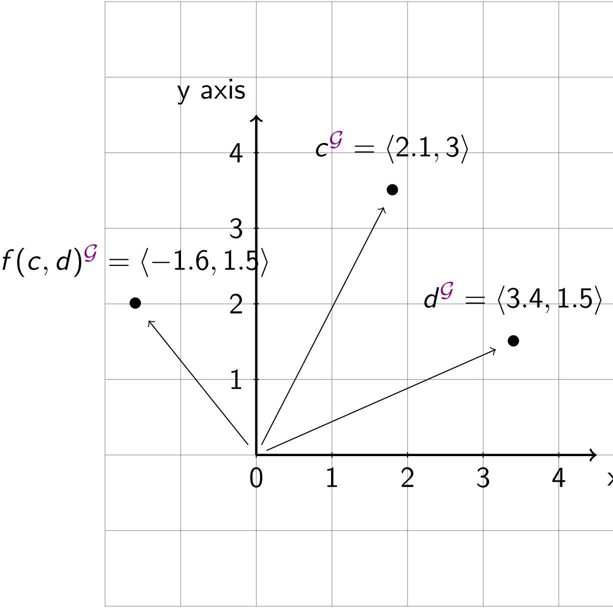
Grounding FOL Signature in $\mathbb{R}^2$

- $c^{\mathcal{G}} = \langle 2.1, 3 \rangle$
- $d^{\mathcal{G}} = \langle 3.4, 1.5 \rangle$



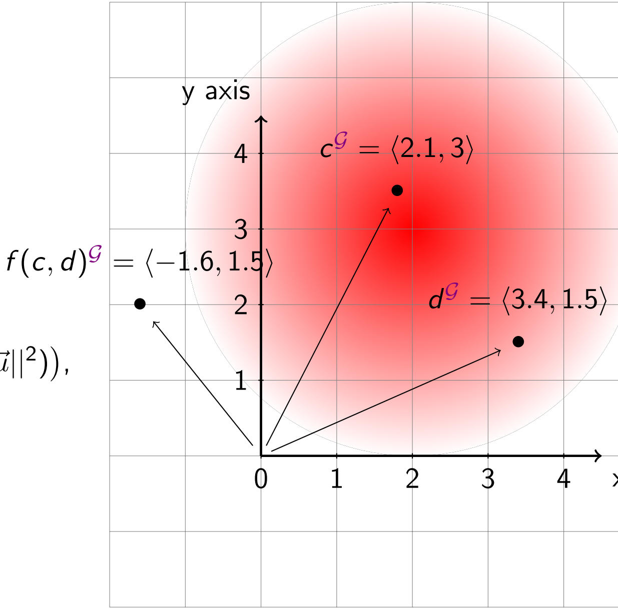
Grounding FOL Signature in $\mathbb{R}^2$

- $c^{\mathcal{G}} = \langle 2.1, 3 \rangle$
- $d^{\mathcal{G}} = \langle 3.4, 1.5 \rangle$
- $f^{\mathcal{G}} : \vec{x}, \vec{y} \mapsto \vec{x} - \vec{y}$



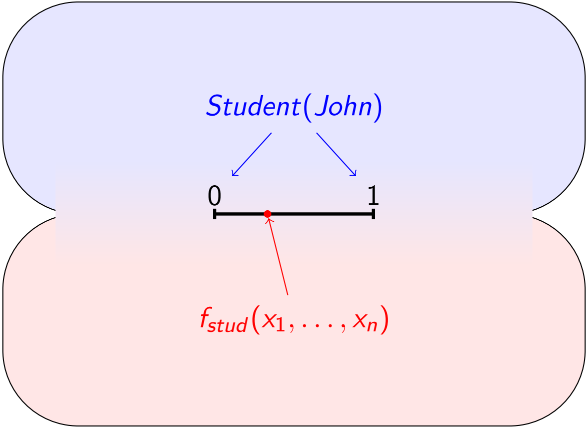
Grounding FOL Signature in $\mathbb{R}^2$

- $c^{\mathcal{G}} = \langle 2.1, 3 \rangle$
- $d^{\mathcal{G}} = \langle 3.4, 1.5 \rangle$
- $f^{\mathcal{G}} : \vec{x}, \vec{y} \mapsto \vec{x} - \vec{y}$
- $P^{\mathcal{G}} : \vec{x} \mapsto \exp\left((-||\vec{x} - \vec{\mu}||^2)\right)$, with $\mu = (2, 3)$



Discrete vs continous

. . . but we ignore one important aspect . . .



- how do we interpret a value in $[0, 1]$ as the truth of a proposition?
- two options:
  - extends the st of truth values to the whole interval $[0, 1]$: fuzzy logic
  - interprets the values in [0,1] as the probability of truth: probability logic

Table of Contents

NeSy based on Fuzzy Logic

Definition (Grounding of formulas)

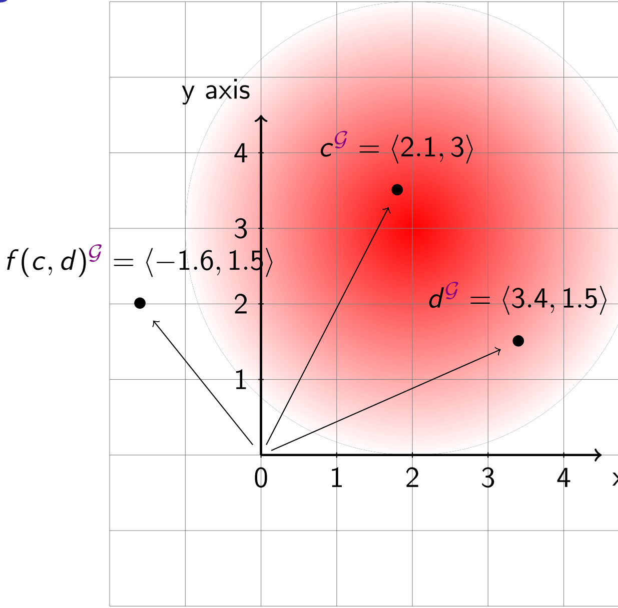
The grounding of formulas is recursively defined according to their structure, and the fuzzy semantics of connectives.

- $P(t_1, \ldots, t_n)^{\mathcal{G}} = P^{\mathcal{G}}(t_1^{\mathcal{G}}, \ldots, t_n^{\mathcal{G}})$
- $(\phi \wedge \psi)^{\mathcal{G}} = \max(\phi^{\mathcal{G}} + \psi^{\mathcal{G}} - 1, 0)$
- $(\phi \rightarrow \psi)^{\mathcal{G}} = min(1 - \phi^{\mathcal{G}} + \psi^{\mathcal{G}}, 1)$
- $(\phi \vee \psi)^{\mathcal{G}} = \min(\phi^{\mathcal{G}} + \psi^{\mathcal{G}}, 1)$
- $(\neg\phi)^{\mathcal{G}} = 1 - \phi^{\mathcal{G}}$

Samy Badreddine et al. (2022). "Logic tensor networks". In: *Artificial Intelligence* 303, p. 103649

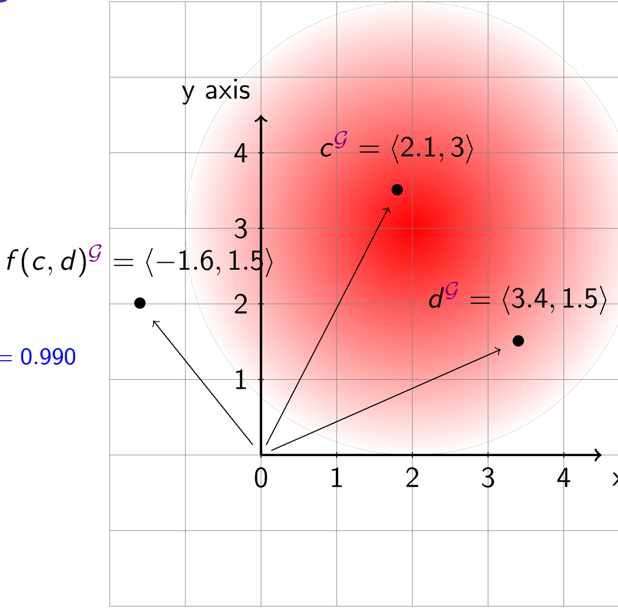
Grounding FOL Signature in $\mathbb{R}^2$

- $c^{\mathcal{G}} = \langle 2.1, 3 \rangle$
- $d^{\mathcal{G}} = \langle 3.4, 1.5 \rangle$
- $f^{\mathcal{G}} : \vec{x}, \vec{y} \mapsto \vec{x} - \vec{y}$
- $P^{\mathcal{G}} : \vec{x} \mapsto \exp\left(-(\vec{x} - \vec{\mu})^2\right)$



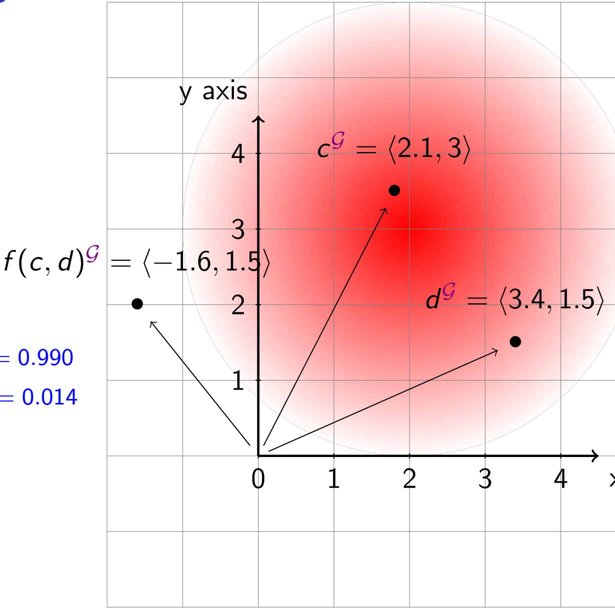
Grounding FOL Signature in $\mathbb{R}^2$

- $c^{\mathcal{G}} = \langle 2.1, 3 \rangle$
- $d^{\mathcal{G}} = \langle 3.4, 1.5 \rangle$
- $f^{\mathcal{G}} : \vec{x}, \vec{y} \mapsto \vec{x} - \vec{y}$
- $P^{\mathcal{G}} : \vec{x} \mapsto \exp\left(-(\vec{x} - \vec{\mu})^2\right)$
- $P(c)^{\mathcal{G}} = exp(-||c^{\mathcal{G}} - \vec{\mu}||^2) = 0.990$



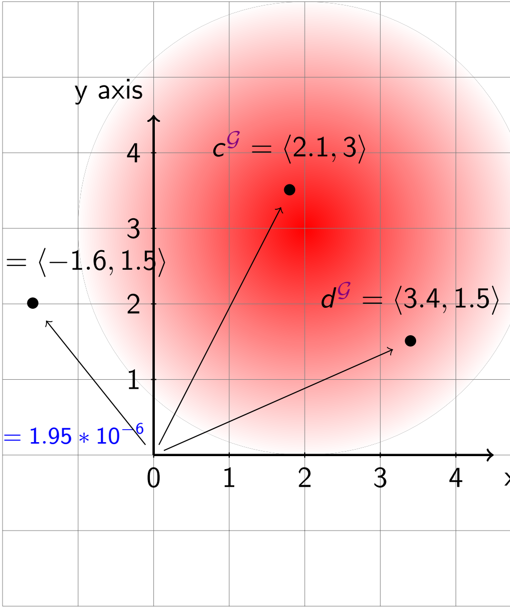
Grounding FOL Signature in $\mathbb{R}^2$

- $c^{\mathcal{G}} = \langle 2.1, 3 \rangle$
- $d^{\mathcal{G}} = \langle 3.4, 1.5 \rangle$
- $f^{\mathcal{G}} : \vec{x}, \vec{y} \mapsto \vec{x} - \vec{y}$
- $P^{\mathcal{G}} : \vec{x} \mapsto \exp\left(-(\vec{x} - \vec{\mu})^2\right)$
- $P(c)^{\mathcal{G}} = exp(-||c^{\mathcal{G}} - \vec{\mu}||^2) = 0.990$
- $P(d)^{\mathcal{G}} = exp(-||d^{\mathcal{G}} - \vec{\mu}||^2) = 0.014$

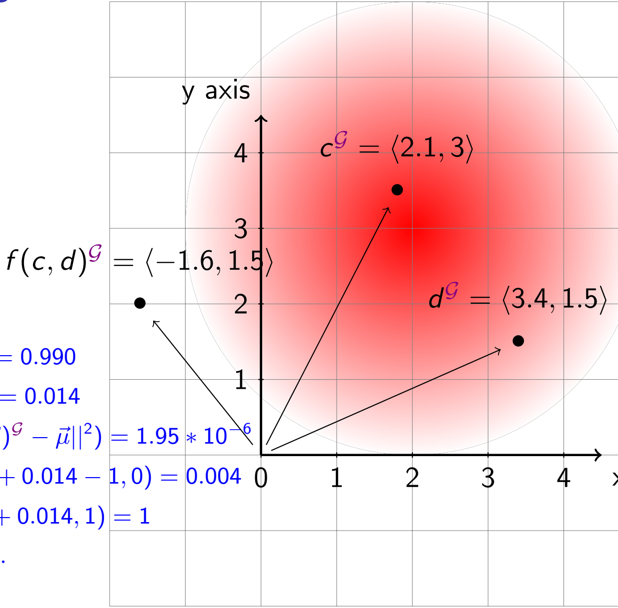


Grounding FOL Signature in $\mathbb{R}^2$

- $c^{\mathcal{G}} = \langle 2.1, 3 \rangle$
- $d^{\mathcal{G}} = \langle 3.4, 1.5 \rangle$
- $f^{\mathcal{G}} : \vec{x}, \vec{y} \mapsto \vec{x} - \vec{y}$
- $P^{\mathcal{G}} : \vec{x} \mapsto \exp\left(-(\vec{x} - \vec{\mu})^2\right)$
- $P(c)^{\mathcal{G}} = exp(-||c^{\mathcal{G}} - \vec{\mu}||^2) = 0.990$
- $P(d)^{\mathcal{G}} = exp(-||d^{\mathcal{G}} - \vec{\mu}||^2) = 0.014$
- $P(f(c, d))^{\mathcal{G}} = exp(-||f(c, d)^{\mathcal{G}} - \vec{\mu}||^2) = 1.95 * 10^{-6}$



Grounding FOL Signature in $\mathbb{R}^2$



- $c^{\mathcal{G}} = \langle 2.1, 3 \rangle$
- $d^{\mathcal{G}} = \langle 3.4, 1.5 \rangle$
- $f^{\mathcal{G}} : \vec{x}, \vec{y} \mapsto \vec{x} - \vec{y}$
- $P^{\mathcal{G}} : \vec{x} \mapsto \exp\left(-(\vec{x} - \vec{\mu})^2\right)$
- $P(c)^{\mathcal{G}} = exp(-||c^{\mathcal{G}} - \vec{\mu}||^2) = 0.990$
- $P(d)^{\mathcal{G}} = exp(-||d^{\mathcal{G}} - \vec{\mu}||^2) = 0.014$
- $P(f(c,d))^{\mathcal{G}} = exp(-||f(c,d)^{\mathcal{G}} - \vec{\mu}||^2) = 1.95 * 10^{-6}$
- $P(c) \wedge P(d)^{\mathcal{G}} = \max(0.990 + 0.014 - 1, 0) = 0.004$
- $P(c) \vee P(d)^{\mathcal{G}} = \min(0.990 + 0.014, 1) = 1$
- $\neg P(f(c,d)) \rightarrow P(d))^{\mathcal{G}} = \dots$

Grounding FOL quantifiers

In fuzzy semantics

The semantics of $\forall x \phi(x)$ and $\exists x \phi(x)$ is given in terms of min and *max* aggregators

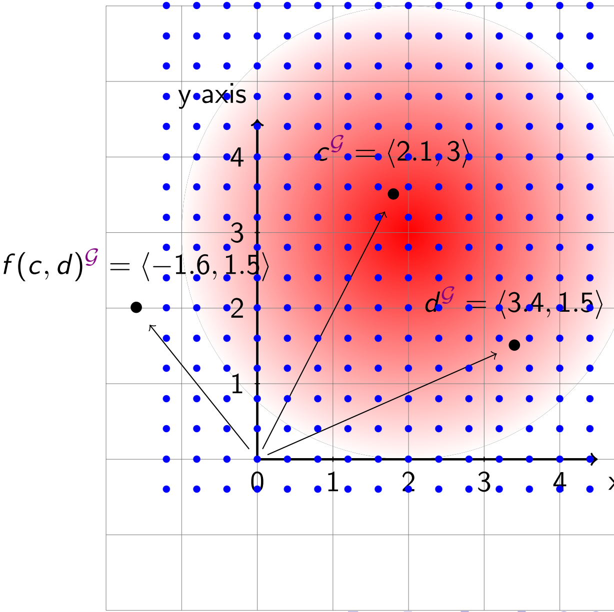
$$(\forall x \phi(x))^{\mathcal{G}} = \min_{\mathrm{x} \in \mathbb{R}^k} \phi(\mathrm{x})^{\mathcal{G}}$$

$$(\exists x \phi(x))^{\mathcal{G}} = \max_{\mathrm{x} \in \mathbb{R}^k} \phi(\mathrm{x})^{\mathcal{G}}$$

- $\min_{\mathrm{x} \in \mathbb{R}^k} \phi^{\mathcal{G}}(\mathrm{x})$ can not be computed directly, as it involves an uncountably infinite number of instances.
- It could be solved analytically, but this involves human intervention
- we approximate the semantics of quantifiers by domain sampling

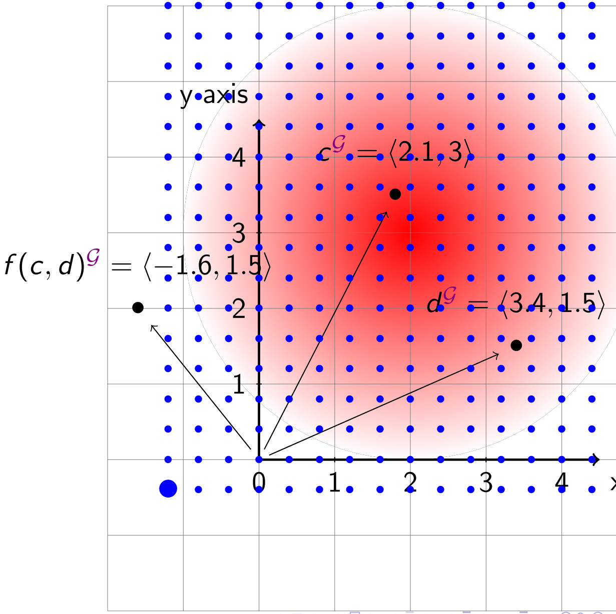
Grounding FOL Signature in $\mathbb{R}^2$

- $c^{\mathcal{G}} = \langle 2.1, 3\rangle$
- $d^{\mathcal{G}} = \langle 3.4, 1.5\rangle$
- $f^{\mathcal{G}} : \vec{x}, \vec{y} \mapsto \vec{x} - \vec{y}$
- $P^{\mathcal{G}} : \vec{x} \mapsto \exp - ((\vec{x} - \vec{\mu})^2)$



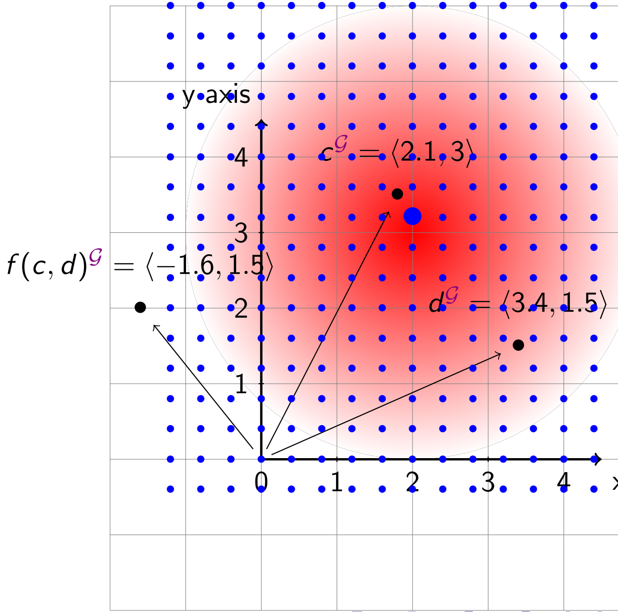
Grounding FOL Signature in $\mathbb{R}^2$

- $c^{\mathcal{G}} = \langle 2.1, 3 \rangle$
- $d^{\mathcal{G}} = \langle 3.4, 1.5 \rangle$
- $f^{\mathcal{G}} : \vec{x}, \vec{y} \mapsto \vec{x} - \vec{y}$
- $P^{\mathcal{G}} : \vec{x} \mapsto \exp - \left((\vec{x} - \vec{\mu})^2\right)$
- $(\forall x P(x))^{\mathcal{G}} = 0.000\ldots$



Grounding FOL Signature in $\mathbb{R}^2$

- $c^{\mathcal{G}} = \langle 2.1, 3 \rangle$
- $d^{\mathcal{G}} = \langle 3.4, 1.5 \rangle$
- $f^{\mathcal{G}} : \vec{x}, \vec{y} \mapsto \vec{x} - \vec{y}$
- $P^{\mathcal{G}} : \vec{x} \mapsto \exp - ((\vec{x} - \vec{\mu})^2)$
- $(\forall x P(x))^{\mathcal{G}} = 0.000\ldots$
- $(\exists x P(x))^{\mathcal{G}} \approx 0.96$



Neuro Symbolic tasks

- **learning grounding** Let T be a set of formulas on a signature Σ and $\mathcal{G}$ a grounding of Σ on the set of parameters $\boldsymbol{\theta}_\Sigma$.

$$\boldsymbol{\theta}^*_\Sigma = \underset{\boldsymbol{\theta}_\Sigma \in \boldsymbol{\Theta}_\Sigma}{\operatorname{argmax}} \, \mathcal{G}\left(\bigwedge_{\phi \in \mathcal{T}} \phi \;\middle|\; \boldsymbol{\theta}_\Sigma \right)$$

- **answering queries** Let ϕ be a closed formula (query) the answer is the truth value computed as:

$$\mathcal{G}(\phi \mid \boldsymbol{\theta}^*_\phi)$$

Table of Contents

Problog

```
  0.1 :: burglary.
  0.7 :: hears_alarm(mary).
  0.2 :: earthquake.
  0.4 :: hears_alarm(john).
alarm :- earthquake.
alarm :- burglary.
calls(X) :- alarm, hears_alarm(X).
call :- calls(X).
```

Domain

All the constants appearing in the program

$$D = \{mary, john\}$$

Herbrand base

All the variable free atoms of the program, + all the atoms obtained by replacing the variables of an atom with elements of the domain.

$$H = \left\{ \begin{array}{c} burglary, earthquake, alarm, call \\ hears\_alarm(mary), hears\_alarm(john), \\ calls(mary), calls(john) \end{array} \right\}$$

Problog semantics

Possible world

let: $\boldsymbol{x} = (x_1, \dots, x_n)$, $\boldsymbol{x}_i \subseteq \boldsymbol{x}$ Variables
$\boldsymbol{a} = (a_1, \dots, a_n)$, $\boldsymbol{a}_i \subseteq \boldsymbol{a}$ elements of the domain
a possible world ω is defined as:

$$\omega : H \to \{0, 1\}$$

such that for each $h(\boldsymbol{a})$ which is not a probabilistic fact: $\omega(h(\boldsymbol{a})) = 1$ if and only if there is a rule

$$h(\boldsymbol{x}) \text{ :- } b_1(\boldsymbol{x}_1), \dots, b_n(\boldsymbol{x}_n)$$

and

$$\omega(b_1(\boldsymbol{a}_1)) = 1, \dots \omega(b_n(\boldsymbol{a}_n)) = 1$$

Worlds: example

```
  0.1 :: burglary.
  0.2 :: earthquake.
  0.7 :: hears_alarm(mary).
  0.4 :: hears_alarm(john).
alarm :- earthquake.
alarm :- burglary.
calls(X) :- alarm, hears_alarm(X).
call :- calls(X).
```

worlds	ω_1	ω_2	ω_3	...	ω_i	...	ω_n
burglary	0	1	0	...	1	...	1
earthquake	0	0	1	...	0	...	0
hears_alarm(mary)	0	0	0	...	1	...	1
hears_alarm(john)	0	0	0	...	0	...	1
alarm	0	1	1	...	1	...	1
calls(mary)	0	0	0	...	1	...	1
calls(john)	0	0	0	...	0	...	1
call	0	0	0	...	1	...	1
worldprobabilisty				...		...	

Worlds: example

```
  0.1 :: burglary.
  0.2 :: earthquake.
  0.7 :: hears_alarm(mary).
  0.4 :: hears_alarm(john).
alarm :- earthquake.
alarm :- burglary.
calls(X) :- alarm, hears_alarm(X).
call :- calls(X).
```

worlds	ω_1	ω_2	ω_3	...	ω_i	...	ω_n
burglary	0.9	0.1	0.9	...	0.1	...	0.1
earthquake	0.8	0.8	0.2	...	0.8	...	0.8
hears_alarm(mary)	0.3	0.3	0.3	...	0.7	...	0.7
hears_alarm(john)	0.6	0.6	0.6	...	0.6	...	0.4
alarm	0	1	1	...	1	...	1
calls(mary)	0	0	0	...	1	...	1
calls(john)	0	0	0	...	0	...	1
call	0	0	0	...	1	...	1
world probabilities	0.1296	0.0144	0.0324	...	0.0336	...	0.0224

Tasks

World probability

Probabilistic facts $p_1 :: f_1, \; p_2 :: f_2, \; \ldots, p_k :: f_k$

$$P(\omega) = \prod_{\omega(f_i)=1} p_i \cdot \prod_{\omega(f_i)=0} (1 - p_i)$$

Probability of a query

Let $q \in H$ a query. The probability of the query is

$$\mathbb{E}_P(\omega(q)) = \sum_{\omega} \omega(q) P(\omega)$$

DeepProblog

A DeepProbLog program is a ProbLog program that is extended with a set of ground neural annotated disjunctions

(simplified) Neural Annotated Disjunction

Is a clause of the form:

$$nn(f_q, \boldsymbol{x}, \boldsymbol{u}) :: q(\boldsymbol{x}, u_1); \ldots ; q(\boldsymbol{x}, u_n).$$

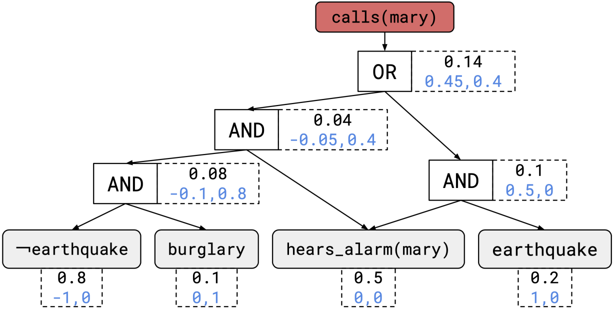
Given an object o with features $\boldsymbol{x}$, let $\boldsymbol{y} = (y_1, \ldots, y_n)$ be the output of $f_q(\boldsymbol{x})$, the effect of the Neural Annotated Disjunction is the addition of the following probabilistic facts to the Problog program

```
y_1 :: q(o,u_1).
y_2 :: q(o,u_2).
.
.
.
y_n :: q(o,u_n).
```

Robin Manhaeve et al. (2018). "DeepProbLog: Neural Probabilistic Logic

Inference in ProbLog

```
0.2 :: earthquake.
0.1 :: burglary.
alarm :- earthquake.
alarm :- burglary.
0.5 :: hears_alarm(mary).
calls(mary) :- alarm,hears_alarm(mary).
query(calls(mary))
```

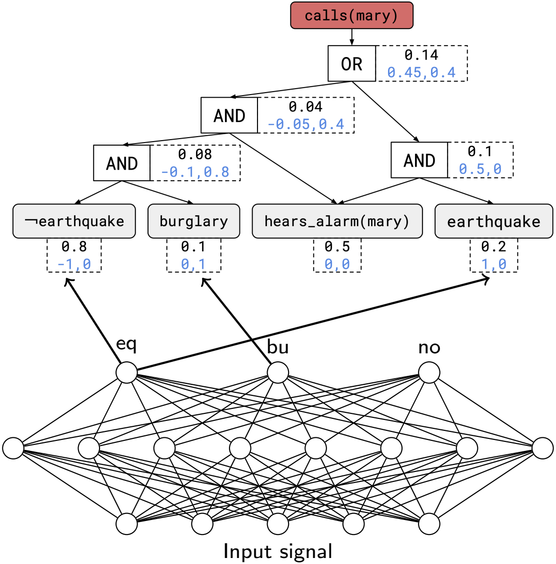


Learning in DeepProblog

```
nn(m_event,X,[eq,bu,no]) :: earthquake;burglary;none.
alarm :- earthquake.
alarm :- burglary.
0.5 :: hears_alarm(mary).
calls(mary) :- alarm,hears_alarm(mary).
query(calls(mary))
```

Learning in DeepProblog

```
nn(m_event,X,[eq,bu,no])
alarm :- earthquake.
alarm :- burglary.
0.5 :: hears_alarm(mary).
calls(mary) :- alarm,hear
query(calls(mary))
```



The SSD for a given query and the neural network that computes the truth value of the neural atoms is the neuro-symbolic architecture that can be trained end-to-end

Table of Contents

Conclusions

Challenges

- integration of large foundational models and large knowledge graphs
- scalability (especially in probabilistic-based NeSy)
- symbol discovering from supervised data
- Generative NeSy models how to use background knowledge of generative models
- Temporal NeSy models integrating temporal logic and recurrent neural architectures
- Training methods for NeSy architectures
- ...

Thanks for Listening

References I

Badreddine, Samy et al. (2022). "Logic tensor networks". In: *Artificial Intelligence* 303, p. 103649.

Donadello, I., L. Serafini, and A. S. d'Avila Garcez (2017). "Logic Tensor Networks for Semantic Image Interpretation". In: *IJCAI*, pp. 1596–1602.

Manhaeve, Robin et al. (2018). "DeepProbLog: Neural Probabilistic Logic Programming". In: *Advances in Neural Information Processing Systems*.

Rivas, A et al. (2023). "A Neuro-Symbolic System over Knowledge Graphs for Link Prediction". In: *Semantic Web*.

Xie, Xuan, Kristian Kersting, and Daniel Neider (2022). "Neuro-Symbolic Verification of Deep Neural Networks". In: *Thirty-First International Joint Conference on Artificial Intelligence*. IJCAI, pp. 3622–3628.