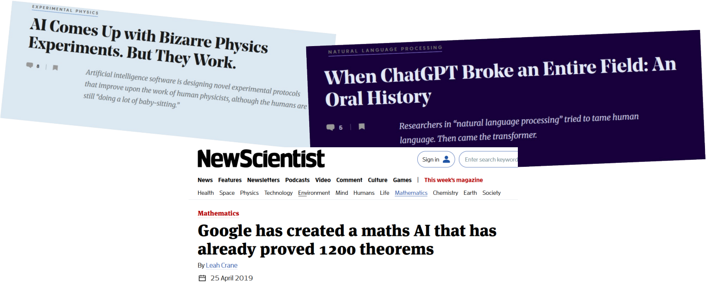


EXPERIMENTAL PHYSICS

AI Comes Up with Bizarre Physics Experiments. But They Work.

8

Artificial intelligence software is designing novel experimental protocols that improve upon the work of human physicists, although the humans are still "doing a lot of baby-sitting."

NATURAL LANGUAGE PROCESSING

When ChatGPT Broke an Entire Field: An Oral History

6

Researchers in "natural language processing" tried to tame human language. Then came the transformer.

NewScientist

Sign in | Enter search keyword

News Features Newsletters Podcasts Video Comment Culture Games | This week's magazine

Health Space Physics Technology Environment Mind Humans Life Mathematics Chemistry Earth Society

Mathematics

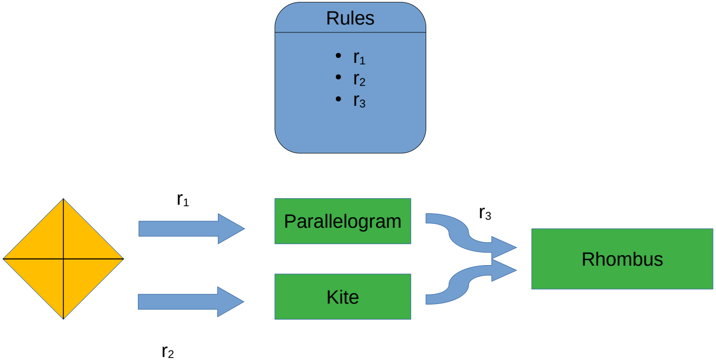
Google has created a maths AI that has already proved 1200 theorems

By Leah Crane

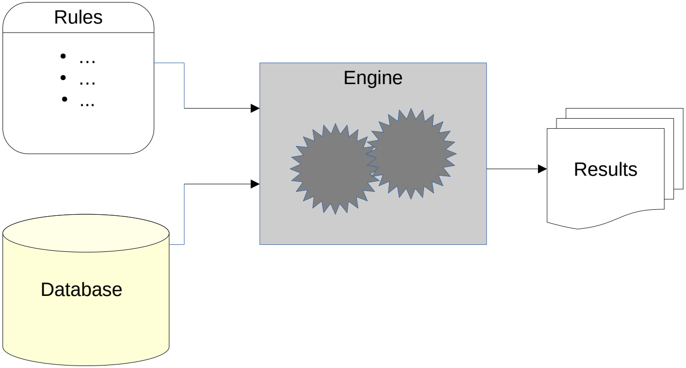
25 April 2019

Why still use symbolic artificial intelligence techniques today?

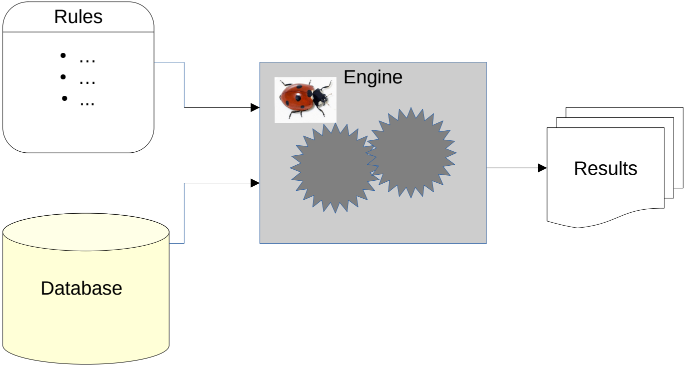
Symbolic AI - In Theory



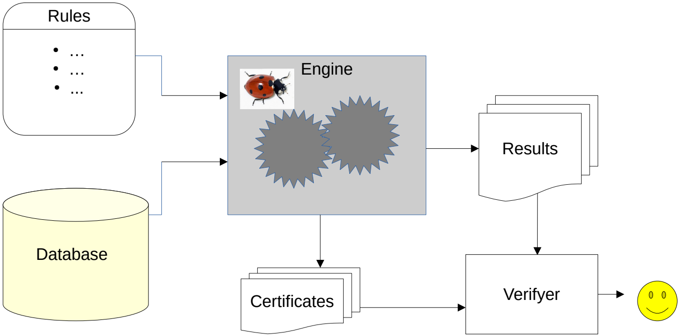
Symbolic AI - In Practice



Symbolic AI - In Practice



Verifiers to the rescue



Certificates are often available and allow verifiers with a formal correctness proof.

Verifying Datalog Reasoning with Lean

Johannes Tantow[1] Lukas Gerlach[2] Stephan Mennicke[2] Markus Krötzsch[2]

[1]TU Chemnitz

[2]Knowledge-Based Systems Group, TU Dresden

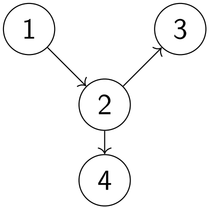
Datalog

$$
\begin{aligned}
&edge(1, 2).\\
&edge(2, 3).\\
&edge(2, 4).
\end{aligned}
$$

Johannes Tantow, Lukas Gerlach, Stephan Mennicke, Markus Krötzsch
TU Chemnitz, Knowledge-Based Systems Group, TU Dresden

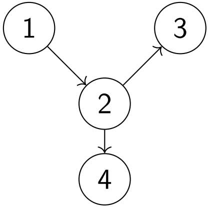
Datalog

$edge(1, 2).$
$edge(2, 3).$
$edge(2, 4).$



Datalog

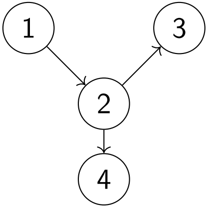
$$edge(1,2).$$
$$edge(2,3).$$
$$edge(2,4).$$

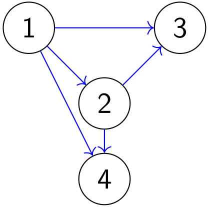


$$\begin{aligned} T(?x,?y) &\leftarrow edge(?x,?y). \\ T(?x,?z) &\leftarrow edge(?x,?y), \\ &\quad T(?y,?z). \end{aligned}$$

Datalog

$edge(1,2).$
$edge(2,3).$
$edge(2,4).$





$T(?x,?y) \leftarrow edge(?x,?y).$
$T(?x,?z) \leftarrow edge(?x,?y),$
$T(?y,?z).$

Semantic: Least model
$\bigcap_{M \text{ is model}} M$
In Lean :
$\{a \mid \forall m, isModel\ m \rightarrow a \in m\}$

Computation of datalog results

Current model:

$$edge(1, 2).$$
$$edge(2, 3).$$
$$edge(2, 4).$$

$$T(?x, ?y) \leftarrow edge(?x, ?y).$$
$$T(?x, ?z) \leftarrow edge(?x, ?y), T(?y, ?z).$$

$$T(2, 3) \leftarrow edge(2, 3).$$

Computation of datalog results

Current model:

$$edge(1,2).$$
$$edge(2,3).$$
$$edge(2,4).$$
$$\color{red}{T(2,3).}$$

$$\begin{array}{c} T(2,3) \\ \uparrow \\ edge(2,3) \end{array}$$

$$T(?x,?y) \leftarrow edge(?x,?y).$$
$$T(?x,?z) \leftarrow edge(?x,?y), T(?y,?z).$$

$T(1,3) \leftarrow$
$edge(1,2), T(2,3).$

Computation of datalog results

Current model:

$$edge(1,2).$$
$$edge(2,3).$$
$$edge(2,4).$$

$$T(2,3).$$
$$T(1,3).$$

$$\begin{tikzcd}[column sep=small]
 & T(1,3) & \\
edge(1,2) \arrow[ur] & & T(2,3) \arrow[ul] \\
 & & edge(2,3) \arrow[u]
\end{tikzcd}$$

$$\begin{aligned} T(?x,?y) &\leftarrow edge(?x,?y). \\ T(?x,?z) &\leftarrow edge(?x,?y), \\ &\qquad T(?y,?z). \end{aligned}$$

Proof trees are short certificates

Validating a correct fact

$$\begin{tikzcd}
& T(1,3) & \\
\mathit{edge}(1,2) \arrow[ur] & & T(2,3) \arrow[ul] \\
& & \mathit{edge}(2,3) \arrow[u]
\end{tikzcd}$$

Johannes Tantow, Lukas Gerlach, Stephan Mennicke, Markus Krötzsch
TU Chemnitz, Knowledge-Based Systems Group, TU Dresden

Validating a correct fact

$$\begin{tikzcd} & T(1,3) & \\ edge(1,2) \arrow[ur] & & T(2,3) \arrow[ul] \\ & & edge(2,3) \arrow[u] \end{tikzcd}$$

$$T(1,3) \leftarrow edge(1,2), T(2,3).$$
$$T(?x, ?z) \leftarrow edge(?x, ?y), T(?y, ?z).$$

Implemented and verified with partial functions like Benzaken et. al. 2017

Theorem

match r r' = true $\leftrightarrow \exists$ substitution s, $r.apply\ s = r' \leftrightarrow \exists$ grounding g, $r.apply\ g = r'$

Validating a correct fact

$$
\begin{tikzcd}[column sep=small]
 & T(1,3) & \\
edge(1,2) \arrow[ur] & & T(2,3) \arrow[ul] \\
 & & edge(2,3) \arrow[u]
\end{tikzcd}
$$

check if $edge(1,2)$ is in the database

```
class Database (τ : Signature) where
  contains: GroundAtom τ → Bool

instance univDatabase (τ : Signature) : Database τ where
  contains := fun _ => true
```

Proof graphs

$$T(?x, ?y) \leftarrow edge(?x, ?y).$$
$$U(?x, ?y) \leftarrow T(?x, ?y)$$
$$T(?x, ?z) \leftarrow edge(?x, ?y), T(?y, ?z), U(?y, ?z).$$

$$\begin{tikzcd}
 & T(1,3) & & & \\
edge(1,2) \arrow[ur] & T(2,3) \arrow[u] & U(2,3) \arrow[ul] & T(2,3) \arrow[l] & edge(2,3) \arrow[l] \\
 & edge(2,3) \arrow[u] & & &
\end{tikzcd}$$

Proof graphs

$$T(?x, ?y) \leftarrow edge(?x, ?y).$$
$$U(?x, ?y) \leftarrow T(?x, ?y)$$
$$T(?x, ?z) \leftarrow edge(?x, ?y), T(?y, ?z), U(?y, ?z).$$

$$\begin{tikzcd}
 & T(1,3) & \\
edge(1,2) \arrow[ur] & T(2,3) \arrow[u] \arrow[r] & U(2,3) \arrow[ul] \\
 & edge(2,3) \arrow[u] &
\end{tikzcd}$$

Proof graphs

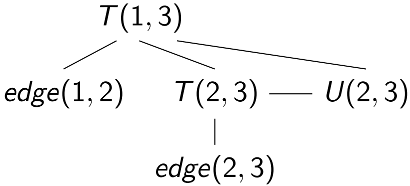
$$T(?x, ?y) \leftarrow edge(?x, ?y).$$
$$\color{red}{U(?x, ?y) \leftarrow T(?x, ?y)}$$
$$T(?x, ?z) \leftarrow edge(?x, ?y), T(?y, ?z), \color{red}{U(?y, ?z)}\color{black}{.}$$

$$\begin{tikzcd}
 & T(1,3) & \\
edge(1,2) \arrow[ur] & T(2,3) \arrow[u] \arrow[r] & U(2,3) \arrow[ul] \\
 & edge(2,3) \arrow[u] &
\end{tikzcd}$$

Next steps

1. Define a practicable graph model
2. Implement and verify depth-first search

Implementing a graph - Mathlib



Problems

1. Which rule is represented?
2. Uses an adjacency matrix

Implementing a directed graph

```
structure Graph (A) where
    vertices : List A
    predecessors : A → List A
    complete : ∀ a, (predecessors a).all
        fun x => x ∈ vertices
```

Good

Easy to use in proof and implement

Bad

Slow in practice

Implementing a directed graph

```
abbrev PreGraph (A) := Std.HashMap A (List A)
def vertices (pg : PreGraph) := pg.keys
def predecessors (pg : PreGraph) (a : A) := pg.getD a []
def complete : ∀ a, (predecessors a).all
    fun x => x ∈ vertices
```

Good

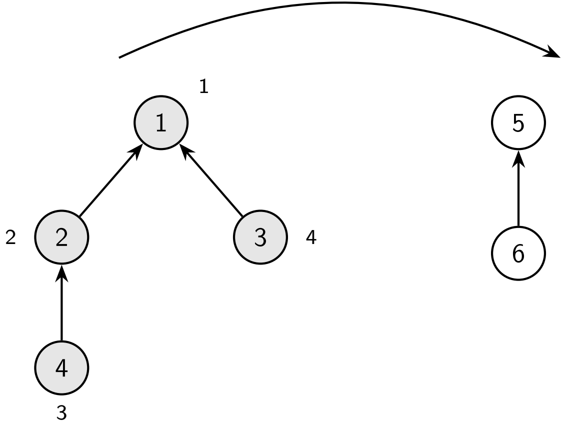
Fast in practice

Bad

Required proving the correctness of multiple Hashmap operations

Depth-first search

dfs: for each vertex v, if !visited then dfs_step(v)



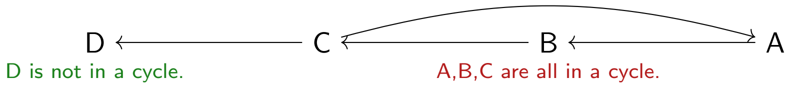
Theorem

$dfs\ G\ f = true \leftrightarrow$
$\forall p, \neg G.isCycle\ p \wedge \forall v, f\ v\ G$

Theorem?

$dfs\_step\ G\ f\ v\ visited = true \leftrightarrow$
$\ldots$

Correctness for depth first search



Correctness for depth first search

D is not in a cycle.
D can be reached from a cycle.

A,B,C are all in a cycle.
A,B,C all can be reached from a cycle.

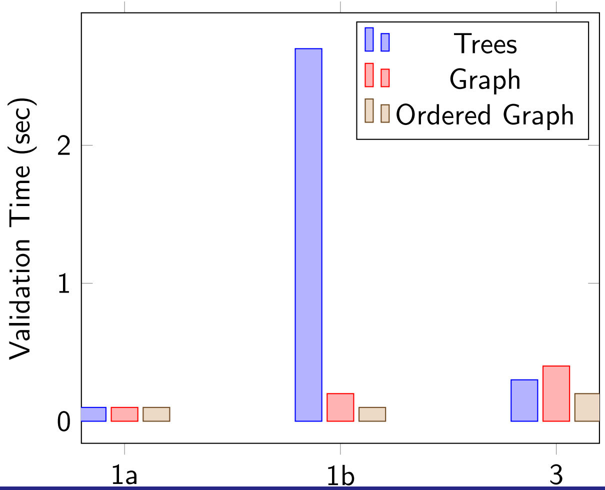
Ordered proof graphs

$$
\begin{tikzcd}
 & T(1,3) & \\
edge(1,2) \arrow[ur] & T(2,3) \arrow[u] \arrow[r] & U(2,3) \arrow[ul] \\
 & edge(2,3) \arrow[u] &
\end{tikzcd}
$$

Ordered Proof

1. $edge(2,3); []$
2. $T(2,3); [1]$
3. $edge(1,2); []$
4. $U(2,3); [2]$
5. $T(1,3); [3,2,4]$

Evaluation



Scenario	Nemo time
1a	59s
1b	0.1s
3	7.8s

1a : transitive closure of chain of length 1000

1b : all facts from smaller transitive closure

3 : 1000 facts from real-world medical ontology

Further results & Open problems

What to find more in the paper:

1. Proofs and more details on the implementation
2. How to validate completeness?

Further results & Open problems

What to find more in the paper:

1. Proofs and more details on the implementation
2. How to validate completeness?

Open questions:

1. Direct integration in a datalog engine
2. Expanding to more features like negation