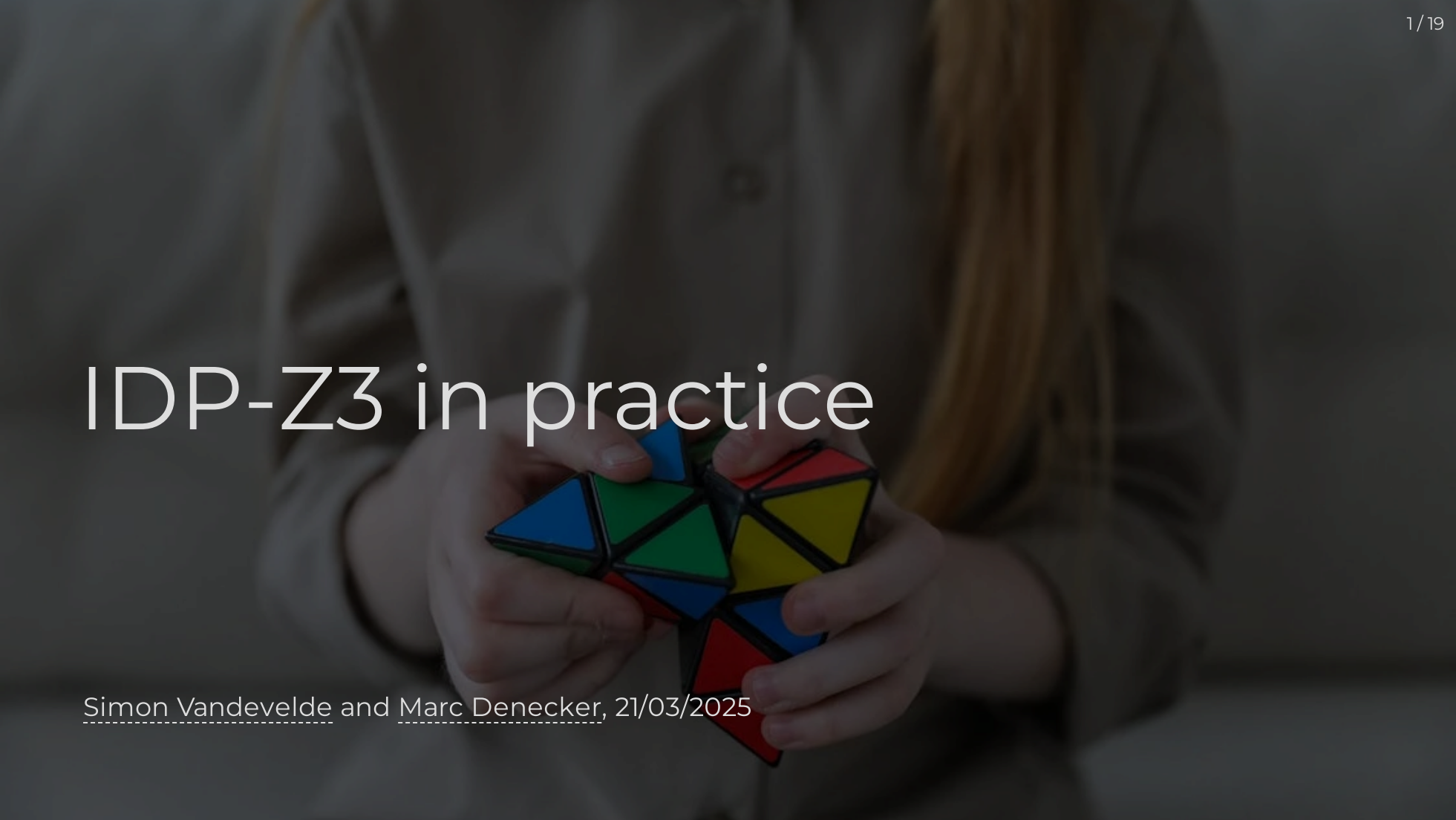


IDP-Z3 in practice

Simon Vandevelde and Marc Denecker, 21/03/2025

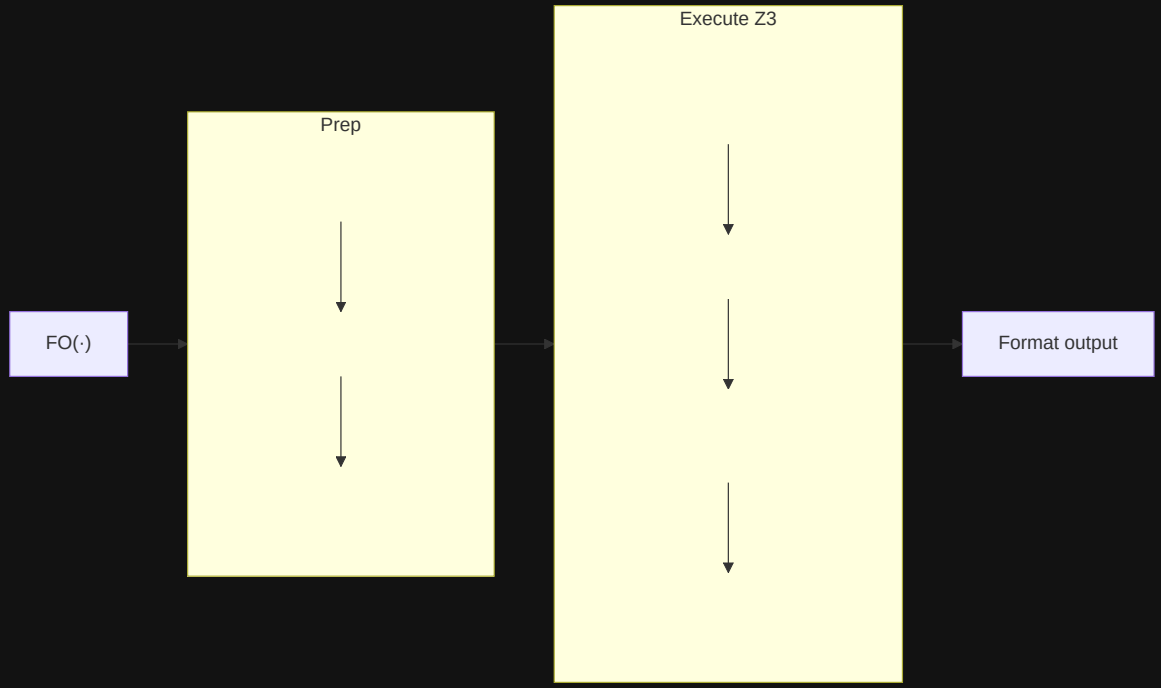
What is IDP?

- Umbrella term for multiple systems
- Each system improves on previous
  - 2008: IDP2 (only model expansion)
  - 2013: IDP3
  - 2019: **IDP-Z3**
  - 2025: ?

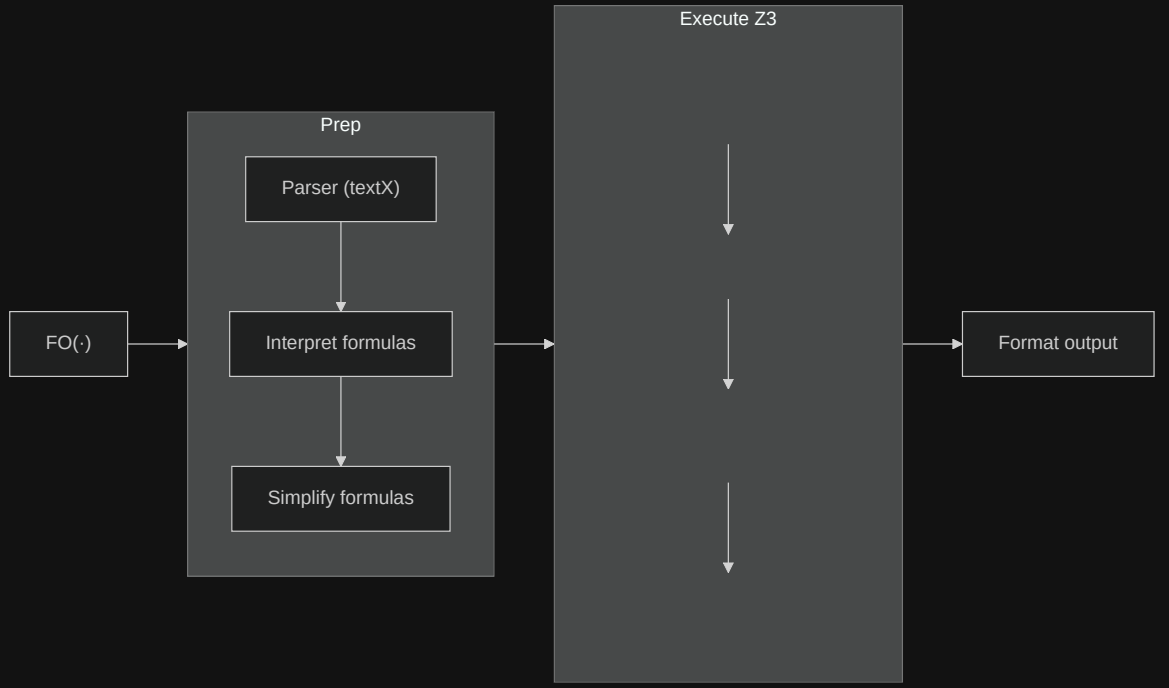
IDP-Z3

- Reasoning engine for FO(·)
- Written in Python
- Open source under LGPLv3
- Uses Z3 SMT solver internally
- Has a rich ecosystem

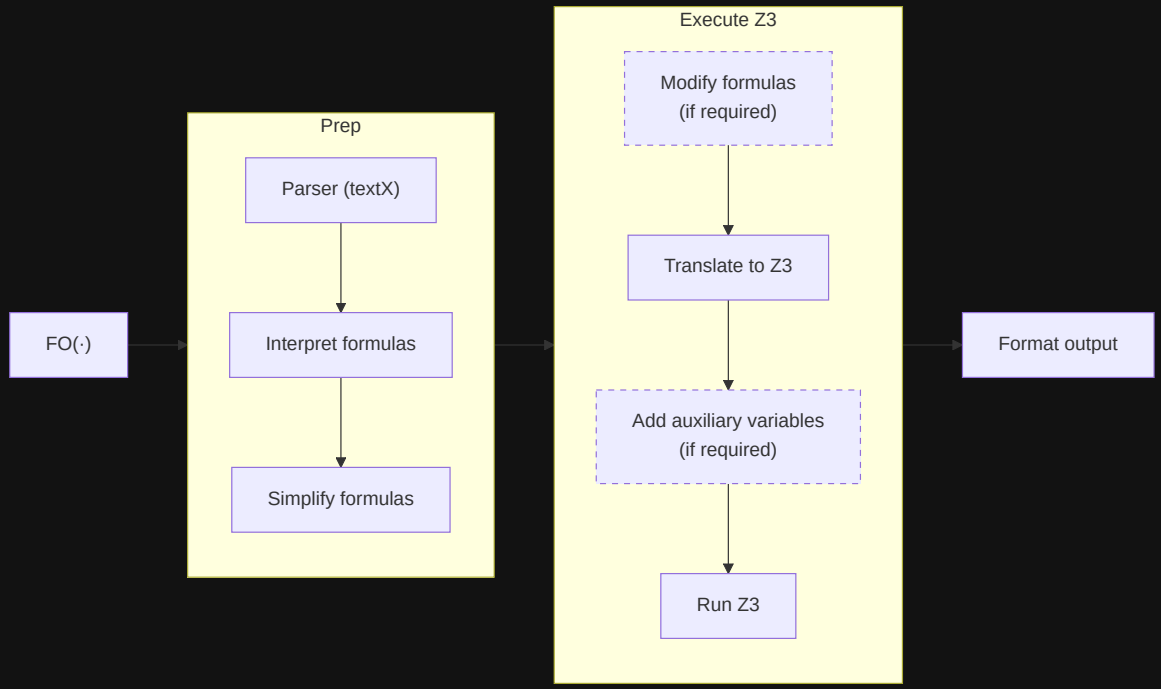
IDP-Z3: main architecture



IDP-Z3: main architecture



IDP-Z3: main architecture



Dotted boxes: inference-specific

Example

Graph coloring with 3 nodes, 2 colors:

$$edge = \{(A, B), (B, C)\}.$$

FO(·) Formula:

$$\forall x, y \in node : edge(x, y) \Rightarrow color(x) \neq color(y).$$

1. Remove quantification:

$$(edge(A, A) \Rightarrow color(A) \neq color(A))$$
$$\wedge (edge(A, B) \Rightarrow color(A) \neq color(B))$$
$$\wedge (edge(A, C) \Rightarrow color(A) \neq color(C))$$
$$\wedge \ldots$$

Example contd.

$(edge(A, A) \Rightarrow color(A) \neq color(A))$
$\wedge(edge(A, B) \Rightarrow color(A) \neq color(B))$
$\wedge(edge(A, C) \Rightarrow color(A) \neq color(C))$
$\wedge \ldots$

2. Interpret using $edge$ interpretion

$(false \Rightarrow color(A) \neq color(A))$
$\wedge(true \Rightarrow color(A) \neq color(B))$
$\wedge(false \Rightarrow color(A) \neq color(8))$
$\wedge \ldots$

3. Simplify

$color(A) \neq color(B) \wedge color(B) \neq color(A) \wedge color(B) \neq color(C) \wedge \ldots$

Example contd. (2)

4. Translate to Z3

```
...
solver.add(Color(A) != Color(B))
solver.add(Color(B) != Color(A))
solver.add(Color(B) != Color(C))
...
```

5. Solve!

```
if solver.sat():
    model = solver.model()
```

Why Z3?

- Z3 supports fragments of FO
- Can reason over infinite domains: $\mathbb{Z}, \mathbb{R}$!
- Fast
- Excellent community
- Open source
- Bindings for many languages!

IDP-Z3 uses Z3's features to solve complex problems

Model-theoretic

Much of what IDP-Z3 does is related to distinguishing **good** from **bad** situations

- Model-check: is my situation correct?
- Model generate: show me a correct situation
- Model expand: given a partial solution, extend it to a full one
- Propagation: what do all feasible situations have in common?
- Explanations: why is a situation not feasible?
- Optimization: find the situation with the most X
- ...

→ all with the same KB! → we can use this to build complex tools

Using IDP-Z3

Most of our use cases:

- Company has knowledge (design, financial, legal, ...)
- Wishes to support staff
- IDP-Z3 is cool, but not user-friendly

$\rightarrow$ Enter: the Interactive Consultant!

Interactive Consultant

Interface for IDP-Z3

- Generic, flexible, interactive
- Main usage loop:
  - Users sets value of their choice
  - IC automatically shows derived knowledge
  - If user is unsure of derivation, they can ask *why* !
- More is possible:
  - Showing a solution
  - Showing an optimal solution w.r.t a term
  - Showing a high-level overview of all remaining solutions

Interactive Consultant: demo

A local handyman is an expert at hanging objects on walls. Sometimes he uses a nail, sometimes he uses glue, sometimes he uses a screw. Over the years, he has gathered the following knowledge:

- Nails support weights up to 25kg, screws support up to 40kg and glue supports only 15kg.
- Nails and screws require drilling holes in the wall, which cannot be done in tile walls.
- Glue is the easiest to use, a nail takes slightly more effort, and a screw is hardest to use.

```
method() = Nail   ⇒ weight() ≤ 25.
method() = Screw  ⇒ weight() ≤ 40.
method() = Glue   ⇒ weight() ≤ 15.
```

```
hole() ⇔ method() = Nail ∨ method() = Screw.
wall() = Tile ⇒ ¬hole().
```

```
{ difficulty() = 1 ← method() = Glue.
difficulty() = 2 ← method() = Nail.
difficulty() = 3 ← method() = Screw.}
```

Link to the IC

IC demo contd.

Now we can answer following questions:

- Which method are possible for hanging a 20kg weight on a brick wall?
- Where can I hang 20kg?
- There is a nail in my wall. How much weight will it support.
- I want to hang a heavy object on my wooden wall, without knowing precisely how heavy it is. Which method will support the most weight?
- Can we hang a 25 kg object on a tile wall?

Example use cases

Component designer:

- International company designs specific component
- Designs are complex
- Lots of engineers, each with own rules-of-thumb
- Lots of costly trial and error

Goal: build a tool to support the engineers in getting more "first-time right" designs

We built a KB capturing the experts' knowledge, which together with IC:

- Succeeds in achieving more first-time right
- Serves as learning tool for newer engineers
- Can challenge assumptions of senior engineers
- Keeps knowledge within the company

Some prior use cases

Intelli-Select:

- Collateral management for loans between large institutions
- Negotiating involves much manual work, takes months
- Using user-friendly interface, companies could enter their own knowledge to KB
- Negotiations in seconds

Notary:

- Registration duties on property purchases in Belgium
- 11 articles of law, quite complex
- Built tool to support notaries in "asking the right questions"
- Relevance inference is crucial!

→ IDP-Z3 has many application domains!

Future research: LLMs

What role can LLMs play for IDP-Z3?

- Text-based interface for IDP-Z3
- Assistant to build a KB
- Maintain KB, validate KB, ...?

Our first foray: VERUS-LM, faithful reasoning for LLMs by using IDP-Z3 as back-end

1. Given problem, create KB
2. Find out relevant inference task
3. Run using IDP-Z3

Beats similar approaches on benchmarks!

https://arxiv.org/abs/2501.14540

Thank you

More info:

- Project: https://idp-z3.be
- Online IDE: https://interactive-consultant.idp-z3.be
- Online tutorial: https://interactive-idp.gitlab.io/
- Slides: https://slides.simonvandevelde.be/BCC25/slides.pdf

 https://simonvandevel.de

 s.vandevelde@kuleuven.be

 @saltfactory@mastodon.social

 Simon Vandevelde