

Developing a user-propagator for graph-based SMT reasoning

Alec Critten

Swansea University/Siemens Mobility

(supervisors: Dr. Monika Seisenberger, Dr. Anton Setzer)

City St. George's, University of London
London, England

11 September 2025

Overview

- 1 Background
- 2 Introducing the user-propagator
- 3 Propagation for graphs
- 4 Conclusion

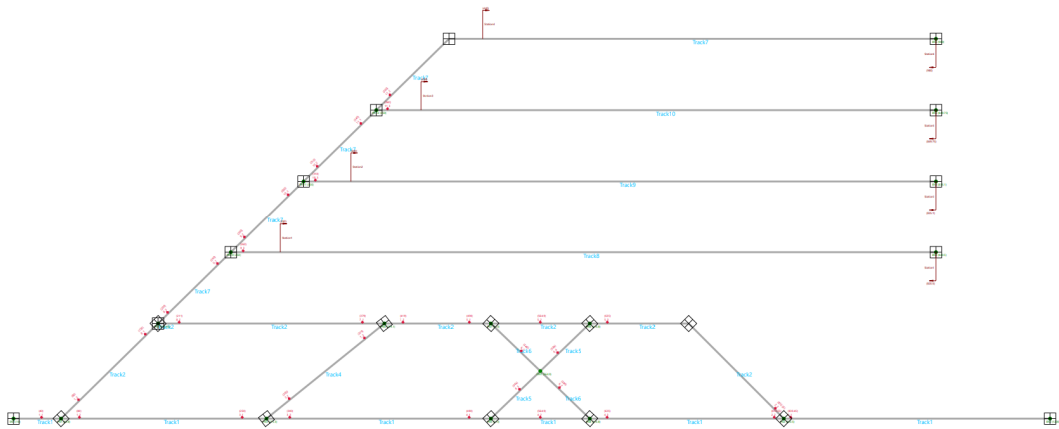
Project Aims

- Improve the **formal verification process** of geographic scheme data for railways
- Develop a custom **graph theory** for the Satisfiability Modulo Theories (SMT) solver Z3
- Integrate graph theory support into Siemens Mobility's workflow for scheme plan verification [1]

Two key definitions

- *Scheme plans* are formal geographic representations of railway systems.
- *Design rules* specify safety-critical constraints which must be verified to hold in the scheme plan.

A scheme plan visualisation



Two example design rules

[BG-04] Designed maximum spacing between adjacent balise groups shall be constrained by:

1. $\leq 1500\text{m}$ between reference balises i.e. with $N\_PIG=0$

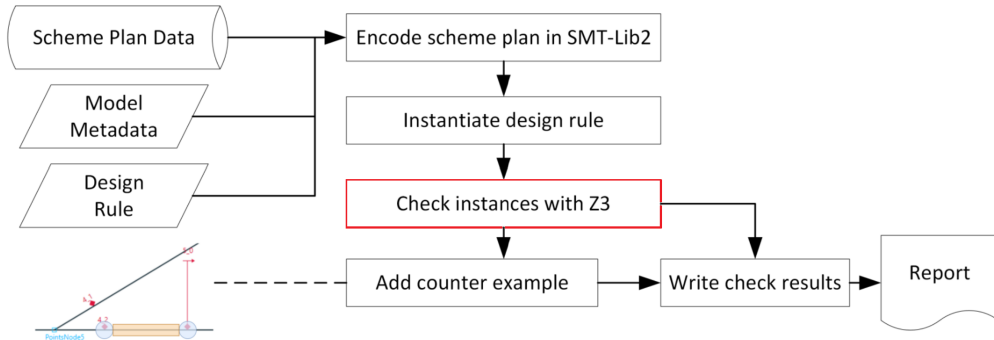
Rationale. This is a rewording of RBC Engineering Rule ER-371 'The distance between consecutive balise groups shall be less than 1500m.'

[BG-05] Designed minimum spacing between adjacent balise groups shall be constrained by:

1. $\geq 12\text{ m}$ between adjacent end balises, one at each end of the two groups

Rationale. This is a design decision based on the maximum spacing permitted between adjacent balises of the same BG (Subset-040 4.1.1.2).

The verification tool-chain [1]



What is SMT?

Satisfiability Modulo Theories (SMT):

- generalises propositional satisfiability (SAT) to first-order logic
- uses decision procedures to check satisfiability with respect to background theories
 - Inbuilt theories: linear arithmetic, bit-vectors, arrays, etc.

SMT's “lazy solving” approach

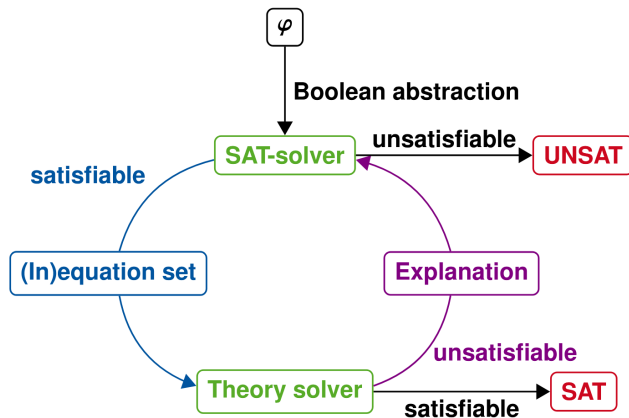
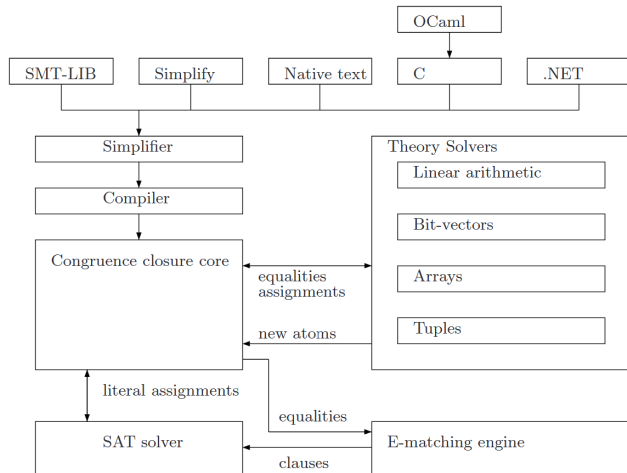


Figure: (Lifted from Erika Ábrahám's lectures at Marktoberdorf 2025)

Why SMT?

- Industry demand as robust verification tool
- Fully automated theorem proving
- A vision for an expressive language w.r.t. scheme plan verification

Z3 is a deeply complicated system... [2]



...only well-known by a few experts...



...but there is a solution!

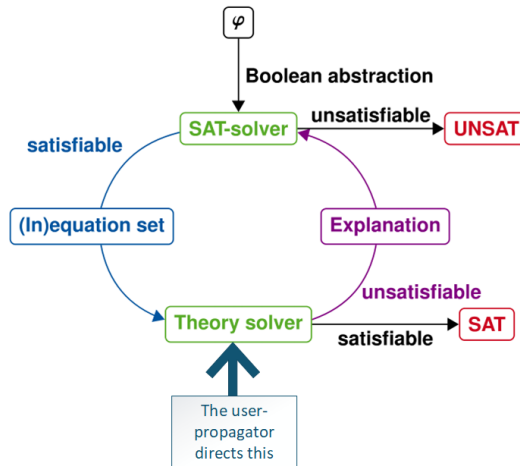
Introducing: user-propagators!

What is a user-propagator?

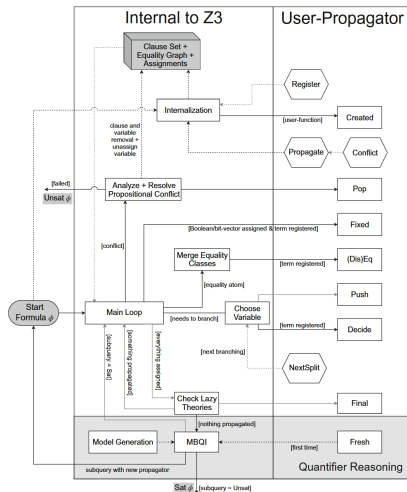
A **user-propagator** (Eisenhofer et. al. 2023) [3]:

- is an **interface** for customised decision procedures
- uses the **EU**F (equality with uninterpreted functions) theory
- operates in an external code-base, exposing **callbacks** to override Z3 behaviour

User-propagators in SMT



User-propagators with Z3 architecture [3]



User-propagator callbacks

Some callbacks of interest:

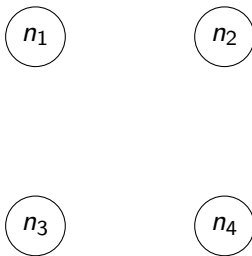
- `Created` - instantiate expressions to be eventually assigned to true/false
- `Push/Pop` - save/revert the state of the SAT solver when it branches on a decision, particularly so we can backtrack
- `Fixed` - when the solver decides an expression is true/false
- `Final` - when the solver has assigned all expressions to true/false

The EUF theory

- Always included in SMT
- Everything is **uninterpreted** - just symbols
- **Congruence closure** to determine satisfiability
- Only $=$ comes preloaded with meaning!
- edge is a **user-function**

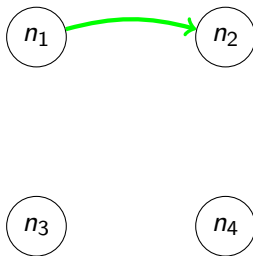
Visualising a graph propagation

A simple propagator enforcing the transitive closure relation over a graph
 $G = \{n_1, n_2, n_3, n_4\}$:



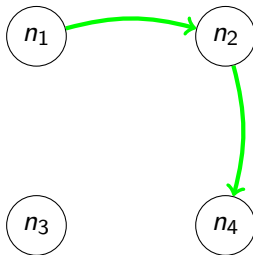
Visualising a graph propagation, cont.

If, for example, we are given that $n_1 \rightarrow n_2 \dots$



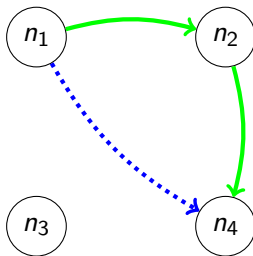
Visualising a graph propagation, cont.

...and $n_2 \rightarrow n_4$...



Visualising a graph propagation, cont.

...then the solver will propagate that $n_1 \rightarrow n_4$:



SMT-LIB input (non-symmetric transitive closure)

```
1  (declare-const n1 Node)
2  (declare-const n2 Node)
3  (declare-const n3 Node)
4  (declare-const n4 Node)
5
6  (assert (edge n1 n2))
7  (assert (edge n2 n3))
8  (assert (edge n3 n4))
9
10 (check-sat)
```

Custom model generation for graphs

Graph model (16 assignments in total):

```
[  
(edge n1 n1) -> false  
(edge n1 n2) -> true  
(edge n1 n3) -> true  
(edge n1 n4) -> true  
(edge n2 n1) -> false  
(edge n2 n2) -> false  
(edge n2 n3) -> true
```

```
(edge n2 n4) -> true  
(edge n3 n1) -> false  
(edge n3 n2) -> false  
(edge n3 n3) -> false  
(edge n3 n4) -> true  
(edge n4 n1) -> false  
(edge n4 n2) -> false  
(edge n4 n3) -> false  
(edge n4 n4) -> false  
]
```

Logging propagation steps

List of unique propagations:

1: (edge n1 n2) = true, derived from input assumption

2: (edge n2 n3) = true, derived from input assumption

3: (edge n1 n3) = true, derived from (ast-vector
 (edge n2 n3)
 (edge n1 n2))

Summary

- Develop **graph theory** for scheme plan verification
- **User-propagators** to customise Z3
- Program **decision procedures** for graphs
- **Groundwork** for expressing design rule constraints in SMT

Swansea Centre for Research in Digital Railways

- Founded 2007 (Railway Verification group), now Center of Expertise in 2025
- Currently 14 staff, 9 PhDs (Computer Science, Engineering, Management)
- Nearly 20 years of research in railways
- Close collaboration with Siemens Mobility
- Other projects/topics: Z3 proof checking, model checking, synthetic test-case generation, ladder-logic verification, timetabling, etc.
- We're hosting the UKRRIN 2026 Student Conference!
 - Submissions open now!
 - More information: <https://ukrrin2026swansea.github.io/>

Discussion Points

What I'm hoping to learn today...

- Unifying framework for SMT theories?
- Formal syntax/semantics for theories
- Process for proposing and developing a new SMT-LIB theory
- Custom decision procedures for theory combinations?

References

-  Banerjee, M., Cai, V., Lakshmanappa, S., Lawrence, A., Roggenbach, M., Seisenberger, M., Werner, T.: A Tool-Chain for the Verification of Geographic Scheme Data. RSSRail 2023. LNCS 14198. Springer (2023).
-  de Moura, L., Bjørner, N.: Z3: An Efficient SMT Solver. TACAS 2008, LNCS 4963. Springer (2008).
-  Bjørner, N., Eisenhofer, C., Kovács, L.: Satisfiability Modulo Custom Theories in Z3. VMCAI 2023. LNCS 13881. Springer (2023).

End of presentation

- Any questions?
- Contact: alec.critten@swansea.ac.uk