A Sound SMT Encoding for TLAPS

Rosalie Defourné, Univ. Lorraine, CNRS, Inria, LORIA
rosalie.defourne@inria.fr

TLA$^+$ Workshop
April 22, 2023

Summary

TLAPS is TLA$^+$'s prover

It discharges proof obligations (POs) to external solvers (Isabelle/TLA$^+$, Zenon, CVC4, Z3, veriT, PTL)

- The current SMT encoding of TLAPS is very efficient, but complex and unreliable
- I made a safer version by removing all optimizations
  The plan was to reimplement the same optimizations, but...
- I accidentally optimized it with SMT triggers instead
- It performs as well as the original

Contents

TLA$^+$ Proofs

$$PartialFcns(A, B) \triangleq \text{UNION } \{[X \longrightarrow B] : X \in \text{SUBSET } A\}$$

$$\begin{array}{lll}
\multicolumn{3}{l}{\text{THEOREM } MyThm \triangleq} \\
\quad \text{ASSUME} & \text{NEW } A, \ \text{NEW } B, \ \text{NEW } f & \\
\quad \text{PROVE} & f \in PartialFcns(A, B) \Leftrightarrow & \land f \in [\text{DOMAIN } f \rightarrow B] \\
 & & \land \text{DOMAIN } f \subseteq A \\
\quad \text{OBVIOUS} & &
\end{array}$$

TLA$^+$ Proofs

$$PartialFcns(A, B) \triangleq \text{UNION } \{[X \longrightarrow B] : X \in \text{SUBSET } A\}$$

THEOREM $MyThm \triangleq$

ASSUME NEW A, NEW B, NEW f

PROVE $f \in PartialFcns(A, B) \Leftrightarrow \land f \in [\text{DOMAIN } f \rightarrow B]$
$\qquad\qquad\qquad\qquad\qquad\qquad\quad \land \text{DOMAIN } f \subseteq A$

BY DEF *PartialFcns*

TLA$^+$ Proofs

$$PartialFcns(A, B) \triangleq \text{ UNION } \{[X \longrightarrow B] : X \in \text{SUBSET } A\}$$

THEOREM $MyThm \triangleq$

ASSUME NEW A, NEW B, NEW f

PROVE $f \in PartialFcns(A, B) \Leftrightarrow \land f \in [\text{DOMAIN } f \rightarrow B]$
$\qquad\qquad\qquad\qquad\qquad\qquad \land \text{DOMAIN } f \subseteq A$

<1>1. $f \in PartialFcns(A, B) \Rightarrow f \in [\text{DOMAIN } f \rightarrow B] \land \text{DOMAIN } f \subseteq A$

BY DEF $PartialFcns$

<1>2. $f \in [\text{DOMAIN } f \rightarrow B] \land \text{DOMAIN } f \subseteq A \Rightarrow f \in PartialFcns(A, B)$

BY DEF $PartialFcns$

<1>. QED

BY <1>1, <1>2

TLA$^+$ Proofs

$$PartialFcns(A, B) \triangleq \text{UNION } \{[X \longrightarrow B] : X \in \text{SUBSET } A\}$$

THEOREM $MyThm \triangleq$

ASSUME NEW A, NEW B, NEW f

PROVE $f \in PartialFcns(A, B) \Leftrightarrow \land f \in [\text{DOMAIN } f \to B]$
$\qquad\qquad\qquad\qquad\qquad\qquad \land \text{DOMAIN } f \subseteq A$

$<1>1.\ f \in PartialFcns(A, B) \Rightarrow f \in [\text{DOMAIN } f \to B] \land \text{DOMAIN } f \subseteq A$

BY DEF $PartialFcns$

$<1>2.\ f \in [\text{DOMAIN } f \to B] \land \text{DOMAIN } f \subseteq A \Rightarrow f \in PartialFcns(A, B)$

BY DEF $PartialFcns$

$<1>.$ QED

BY $<1>1, <1>2$

The Axiomatic Approach to Encoding TLA^+

TLA^+ is unsorted FOL + axiomatic ZFC; We could technically

1. Translate expressions directly using one sort `idv`
2. Specify all builtin operators with axioms

Some axioms (overloading of $f[x]$ for functions and tuples):

$$\forall a, b, f, x : f \in [a \to b] \land x \in a \Rightarrow f[x] \in b$$
$$\forall a, b, t : t \in a \times b \Rightarrow t[1] \in a \land t[2] \in b$$

The Axiomatic Approach to Encoding TLA^+

TLA^+ is unsorted FOL + axiomatic ZFC; We could technically

1. Translate expressions directly using one sort `idv`
2. Specify all builtin operators with axioms

Some axioms (overloading of $f[x]$ for functions and tuples):

$$\forall a, b, f, x : f \in [a \rightarrow b] \land x \in a \Rightarrow f[x] \in b$$
$$\forall a, b, t : t \in a \times b \Rightarrow t[1] \in a \land t[2] \in b$$

However, this might be considered inefficient for SMT
Having *one sorted domain* and *many symbols* makes first-order instantiation even harder than it already is

The Rewriting Approach to Encoding TLA$^+$

The natural idea then is to

1. Perform type synthesis to categorize expressions into sorts
2. Eliminate primitive operators through rewriting

Illustration:

$$[n \in Nat \setminus \{0\} \mapsto n - 1] \in \text{UNION } \{[X \to Nat] : X \in \text{SUBSET } Nat\}$$

$$\downarrow$$

$$\begin{aligned} \exists X^{\texttt{idv}} : \; & \land \forall z^{\texttt{int}} : \texttt{cast}_{\texttt{int}}(z) \in X \Rightarrow 1 \leq_{\texttt{int}} z \\ & \land \forall y^{\texttt{idv}} : y \in X \Leftrightarrow y \in Int \land \texttt{cast}_{\texttt{int}}(0) \leq y \land y \neq \texttt{cast}_{\texttt{int}}(0) \end{aligned}$$

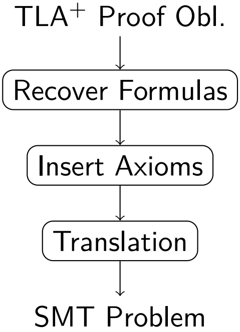
$+$ 3 axioms (one for $\leq$, two for $\texttt{cast}_{\texttt{int}}$)

Issues with the Rewriting-Based Encoding

- Type synthesis is undecidable
- Rewriting may not terminate or leave TLA$^+$ primitives
- **The encoding is too hard to maintain**
    - Mistakes in implementation of rewriting rules happen
    - But the encoded POs are unrecognizable;
      We cannot use the result files for debugging
    - The implementation is just too complex;
      Preprocessing POs practically requires solving them

It *should* be possible to turn off rewriting

Overview of the New Encoding



Back to the Axiomatic approach:

- Minimal typing: `idv` and `bool`
- No rewriting, preserve POs' structure
- Axiomatize everything

The axiomatization is the heart of the encoding

The Axioms of TLA$^+$

Every operator has a set of axioms attached to it

We simply add the axioms for the operators occurring in the PO
(Repeating this recursively if axioms include other operators)

The SMT theory includes 84 axioms:

- 17 for set theory
- 11 for functions
- 14 for arithmetic (handled by SMT's internal arithmetic)
- 25 for sequences
- The remaining 17 for choice, tuples, records and strings
- (reals and bags not supported yet)

Axioms for TLA$^+$ Functions

(Second-order notations used for convenience)

$$\forall F^{\mathtt{idv}\rightarrow\mathtt{idv}}, a^{\mathtt{idv}} : \mathtt{isafcn}([y \in a \mapsto F(y)])$$

$$\forall F^{\mathtt{idv}\rightarrow\mathtt{idv}}, a^{\mathtt{idv}} : \mathrm{DOMAIN}\ [y \in a \mapsto F(y)] = a$$

$$\forall F^{\mathtt{idv}\rightarrow\mathtt{idv}}, a^{\mathtt{idv}}, x^{\mathtt{idv}} : x \in a \Rightarrow [y \in a \mapsto F(y)][x] = F(x)$$

$$\forall a^{\mathtt{idv}}, b^{\mathtt{idv}}, f^{\mathtt{idv}} : f \in [a \rightarrow b] \Rightarrow \mathtt{isafcn}(f) \land \mathrm{DOMAIN}\ f = a$$

$$\forall a^{\mathtt{idv}}, b^{\mathtt{idv}}, f^{\mathtt{idv}}, x^{\mathtt{idv}} : f \in [a \rightarrow b] \land x \in a \Rightarrow f[x]$$

$$\begin{aligned}\forall a^{\mathtt{idv}}, b^{\mathtt{idv}}, f^{\mathtt{idv}} :\ & \land \mathtt{isafcn}(f) \land \mathrm{DOMAIN}\ f = a \\ & \land (\forall x^{\mathtt{idv}} : x \in a \Rightarrow f[x] \in b) \\ & \Rightarrow f \in [a \rightarrow b]\end{aligned}$$

E-matching Patterns (Triggers)

SMT combines SAT-solving with first-order instantiation

Approaches: enumerative, model-based, heuristic-based

$$\forall a^{\mathtt{idv}}, b^{\mathtt{idv}}, f^{\mathtt{idv}}, x^{\mathtt{idv}} : \{f \in [a \to b], x \in a\}$$
$$f \in [a \to b] \land x \in a \Rightarrow f[x] \in b$$

Example: in the ground problem

$$T = [S \to \emptyset], \qquad g \in T, \qquad z \in S$$

The match $\{a \mapsto S, b \mapsto \emptyset, f \mapsto g, x \mapsto z\}$ results in

$$g \in [S \to \emptyset] \land z \in S \Rightarrow g[z] \in \emptyset$$

The Flexibility of Triggers

Many options:

$$\forall a^{\mathtt{idv}}, b^{\mathtt{idv}}, f^{\mathtt{idv}}, x^{\mathtt{idv}} : \begin{array}{l} \{f \in [a \to b], x \in a\} \\ \{f \in [a \to b], f[x]\} \\ \{[a \to b], f[x] \in b\} \\ \{[a \to b], f[x], x \in a\} \end{array}$$

$$f \in [a \to b] \land x \in a \Rightarrow f[x] \in b$$

Challenges:

- Explosion or even Non-termination (matching loops)
- Incompleteness (*crucial* for large proof obligations)

How to Find Good Triggers: An Example

We will try to find a good axiomatization with triggers for the PO:

$$\begin{array}{rl} \text{ASSUME} & S \cap \mathit{Int} \subseteq \emptyset, \\ & 1 \in S \\ \text{PROVE} & \text{FALSE} \end{array}$$

To ensure the theory scales to larger POs, the SMT problem should be provable **only** with instances derived from triggers

We will focus on instantiation alone; assume propositional reasoning is easy

First Attempt

$$\begin{array}{ll} \text{ASSUME} & S \cap Int \subseteq \emptyset, \\ & 1 \in S \\ \text{PROVE} & \text{FALSE} \end{array}$$

First attempt: Use triggers to implement rewriting rules

$$\forall a^{\texttt{idv}}, b^{\texttt{idv}} : \{a \subseteq b\} \qquad \text{(Subseteq)}$$
$$a \subseteq b \Leftrightarrow (\forall x^{\texttt{idv}} : x \in a \Rightarrow x \in b)$$
$$\forall a^{\texttt{idv}}, b^{\texttt{idv}}, x^{\texttt{idv}} : \{x \in a \cap b\} \qquad \text{(Cap)}$$
$$x \in a \cap b \Leftrightarrow x \in a \wedge x \in b$$
$$\forall x^{\texttt{idv}} : \{x \in \emptyset\} \qquad \text{(Empty)}$$
$$\neg(x \in \emptyset)$$

First Attempt—Test

$\forall a^{\mathtt{idv}}, b^{\mathtt{idv}} : \{a \subseteq b\}$	$a \subseteq b \Leftrightarrow (\forall x^{\mathtt{idv}} : x \in a \Rightarrow x \in b)$
$\forall a^{\mathtt{idv}}, b^{\mathtt{idv}}, x^{\mathtt{idv}} : \{x \in a \cap b\}$	$x \in a \cap b \Leftrightarrow x \in a \wedge x \in b$
$\forall x^{\mathtt{idv}} : \{x \in \emptyset\}$	$\neg(x \in \emptyset)$

$S \cap Int \subseteq \emptyset$

$1 \in S$

$1 \in Int$

First Attempt—Test

$\forall a^{\texttt{idv}}, b^{\texttt{idv}} : \{a \subseteq b\}$	$a \subseteq b \Leftrightarrow (\forall x^{\texttt{idv}} : x \in a \Rightarrow x \in b)$
$\forall a^{\texttt{idv}}, b^{\texttt{idv}}, x^{\texttt{idv}} : \{x \in a \cap b\}$	$x \in a \cap b \Leftrightarrow x \in a \wedge x \in b$
$\forall x^{\texttt{idv}} : \{x \in \emptyset\}$	$\neg(x \in \emptyset)$

$S \cap Int \subseteq \emptyset$

$1 \in S$

$1 \in Int$

$S \cap Int \subseteq \emptyset \Leftrightarrow (\forall x^{\texttt{idv}} : x \in S \cap Int \Rightarrow x \in \emptyset)$ not in prenex form

First Attempt—Test

$\forall a^{\texttt{idv}}, b^{\texttt{idv}} : \{a \subseteq b\}$ $\quad a \subseteq b \Leftrightarrow (\forall x^{\texttt{idv}} : x \in a \Rightarrow x \in b)$

$\forall a^{\texttt{idv}}, b^{\texttt{idv}}, x^{\texttt{idv}} : \{x \in a \cap b\}$ $\quad x \in a \cap b \Leftrightarrow x \in a \wedge x \in b$

$\forall x^{\texttt{idv}} : \{x \in \emptyset\}$ $\quad \neg(x \in \emptyset)$

$S \cap Int \subseteq \emptyset$

$1 \in S$

$1 \in Int$

$\forall x^{\texttt{idv}} : S \cap Int \subseteq \emptyset \Rightarrow (x \in S \cap Int \Rightarrow x \in \emptyset)$ $\quad$ no trigger $\longrightarrow$ stuck

$\exists x^{\texttt{idv}} : (x \in S \cap Int \Rightarrow x \in \emptyset) \wedge S \cap Int \subseteq \emptyset$ $\quad$ (unused)

Second Attempt

$$\begin{aligned} \text{ASSUME} \quad & S \cap Int \subseteq \emptyset, \\ & 1 \in S \\ \text{PROVE} \quad & \text{FALSE} \end{aligned}$$

The nested quantifier $\forall x^{\mathtt{idv}}$ came from the axiom (Subseteq)

Second attempt: Reformulate axioms—no nested $\forall$

$$\forall a^{\mathtt{idv}}, b^{\mathtt{idv}} : \{a \subseteq b\} \qquad \text{(SubseteqIntro)}$$

$$(\forall x^{\mathtt{idv}} : x \in a \Rightarrow x \in b) \Rightarrow a \subseteq b$$

$$\forall a^{\mathtt{idv}}, b^{\mathtt{idv}}, {\color{blue}x^{\mathtt{idv}}} : \{a \subseteq b, {\color{blue}x \in a}\} \qquad \text{(SubseteqElim)}$$

$$a \subseteq b \land x \in a \Rightarrow x \in b$$

Second Attempt—Test

$\forall a^{\texttt{idv}}, b^{\texttt{idv}} : \{a \subseteq b\}$	$(\forall x^{\texttt{idv}} : x \in a \Rightarrow x \in b) \Rightarrow a \subseteq b$
$\forall a^{\texttt{idv}}, b^{\texttt{idv}}, x^{\texttt{idv}} : \{a \subseteq b, x \in a\}$	$a \subseteq b \land x \in a \Rightarrow x \in b$
$\forall a^{\texttt{idv}}, b^{\texttt{idv}}, x^{\texttt{idv}} : \{x \in a \cap b\}$	$x \in a \cap b \Leftrightarrow x \in a \land x \in b$
$\forall x^{\texttt{idv}} : \{x \in \emptyset\}$	$\neg(x \in \emptyset)$

$S \cap Int \subseteq \emptyset$

$1 \in S$

$1 \in Int$

We need to know the term $1 \in S \cap Int$, we are already stuck!

Third Attempt

$$\begin{aligned} \text{ASSUME} \quad & S \cap \mathit{Int} \subseteq \emptyset, \\ & 1 \in S \\ \text{PROVE} \quad & \text{FALSE} \end{aligned}$$

We need to generate $1 \in S \cap \mathit{Int}$ from the information available

Third attempt: Provide more triggers to (Cap)

$$\begin{aligned} \forall a^{\mathtt{idv}}, b^{\mathtt{idv}}, x^{\mathtt{idv}} : \; & \{x \in a \cap b\} \\ & \{x \in a, x \in b\} \\ x \in a \cap b & \Leftrightarrow x \in a \land x \in b \end{aligned} \qquad \text{(Cap)}$$

Third Attempt—Test

$\forall a^{\mathtt{idv}}, b^{\mathtt{idv}}, x^{\mathtt{idv}} : \{a \subseteq b, x \in a\} \quad a \subseteq b \land x \in a \Rightarrow x \in b$

$\forall a^{\mathtt{idv}}, b^{\mathtt{idv}}, x^{\mathtt{idv}} : \{x \in a, x \in b\} \quad x \in a \cap b \Leftrightarrow x \in a \land x \in b$

$\forall x^{\mathtt{idv}} : \{x \in \emptyset\} \quad \neg(x \in \emptyset)$

$S \cap Int \subseteq \emptyset$

$1 \in S$

$1 \in Int$

Third Attempt—Test

$$\forall a^{\mathtt{idv}}, b^{\mathtt{idv}}, x^{\mathtt{idv}} : \{a \subseteq b, x \in a\} \quad a \subseteq b \land x \in a \Rightarrow x \in b$$

$$\forall a^{\mathtt{idv}}, b^{\mathtt{idv}}, x^{\mathtt{idv}} : \{x \in a, x \in b\} \quad x \in a \cap b \Leftrightarrow x \in a \land x \in b$$

$$\forall x^{\mathtt{idv}} : \{x \in \emptyset\} \quad \neg(x \in \emptyset)$$

$$S \cap Int \subseteq \emptyset$$

$$1 \in S$$

$$1 \in Int$$

$$1 \in S \cap Int \Leftrightarrow 1 \in S \land 1 \in Int$$

Third Attempt—Test

$\forall a^{\mathtt{idv}}, b^{\mathtt{idv}}, x^{\mathtt{idv}} : \{a \subseteq b, x \in a\} \quad a \subseteq b \land x \in a \Rightarrow x \in b$

$\forall a^{\mathtt{idv}}, b^{\mathtt{idv}}, x^{\mathtt{idv}} : \{x \in a, x \in b\} \quad x \in a \cap b \Leftrightarrow x \in a \land x \in b$

$\forall x^{\mathtt{idv}} : \{x \in \emptyset\} \quad \neg(x \in \emptyset)$

$S \cap Int \subseteq \emptyset$

$1 \in S$

$1 \in Int$

$1 \in S \cap Int \Leftrightarrow 1 \in S \land 1 \in Int$

$1 \in S \cap (S \cap Int) \Leftrightarrow 1 \in S \land 1 \in S \cap Int$

Third Attempt—Test

$\forall a^{\mathtt{idv}}, b^{\mathtt{idv}}, x^{\mathtt{idv}} : \{a \subseteq b, x \in a\} \quad a \subseteq b \land x \in a \Rightarrow x \in b$

$\forall a^{\mathtt{idv}}, b^{\mathtt{idv}}, x^{\mathtt{idv}} : \{x \in a, x \in b\} \quad x \in a \cap b \Leftrightarrow x \in a \land x \in b$

$\forall x^{\mathtt{idv}} : \{x \in \emptyset\} \quad \neg(x \in \emptyset)$

$S \cap Int \subseteq \emptyset$

$1 \in S$

$1 \in Int$

$1 \in S \cap Int \Leftrightarrow 1 \in S \land 1 \in Int$

$1 \in S \cap (S \cap Int) \Leftrightarrow 1 \in S \land 1 \in S \cap Int$

$1 \in S \cap (S \cap (S \cap Int)) \Leftrightarrow 1 \in S \land 1 \in S \cap (S \cap Int)$

and so on...

Fourth Attempt

$$\begin{aligned} \text{ASSUME} \quad & S \cap Int \subseteq \emptyset, \\ & 1 \in S \\ \text{PROVE} \quad & \text{FALSE} \end{aligned}$$

We must prevent matching loops;
A solution is to never create new sets

Final attempt: Use $a \cap b$ as a guard in the trigger

$$\begin{aligned} \forall a^{\texttt{idv}}, b^{\texttt{idv}}, x^{\texttt{idv}} : \ & \{x \in a \cap b\} \\ & \{x \in a, a \cap b\} \\ & \{x \in b, a \cap b\} \\ & x \in a \cap b \Leftrightarrow x \in a \wedge x \in b \end{aligned} \qquad \text{(Cap)}$$

Fourth Attempt—Test

$\forall a^{\mathtt{idv}}, b^{\mathtt{idv}}, x^{\mathtt{idv}} : \{a \subseteq b, x \in a\} \quad a \subseteq b \land x \in a \Rightarrow x \in b$

$\forall a^{\mathtt{idv}}, b^{\mathtt{idv}}, x^{\mathtt{idv}} : \{x \in a, a \cap b\} \quad x \in a \cap b \Leftrightarrow x \in a \land x \in b$

$\forall x^{\mathtt{idv}} : \{x \in \emptyset\} \quad \neg(x \in \emptyset)$

$S \cap Int \subseteq \emptyset$

$1 \in S$

$1 \in Int$

Fourth Attempt—Test

$$\forall a^{\mathtt{idv}}, b^{\mathtt{idv}}, x^{\mathtt{idv}} : \{a \subseteq b, x \in a\} \quad a \subseteq b \land x \in a \Rightarrow x \in b$$

$$\forall a^{\mathtt{idv}}, b^{\mathtt{idv}}, x^{\mathtt{idv}} : \{x \in a, a \cap b\} \quad x \in a \cap b \Leftrightarrow x \in a \land x \in b$$

$$\forall x^{\mathtt{idv}} : \{x \in \emptyset\} \quad \neg(x \in \emptyset)$$

$$S \cap Int \subseteq \emptyset$$

$$1 \in S$$

$$1 \in Int$$

$$1 \in S \cap Int \Leftrightarrow 1 \in S \land 1 \in Int$$

Fourth Attempt—Test

$$\forall a^{\mathtt{idv}}, b^{\mathtt{idv}}, x^{\mathtt{idv}} : \{a \subseteq b, x \in a\} \quad a \subseteq b \wedge x \in a \Rightarrow x \in b$$

$$\forall a^{\mathtt{idv}}, b^{\mathtt{idv}}, x^{\mathtt{idv}} : \{x \in a, a \cap b\} \quad x \in a \cap b \Leftrightarrow x \in a \wedge x \in b$$

$$\forall x^{\mathtt{idv}} : \{x \in \emptyset\} \quad \neg(x \in \emptyset)$$

$$S \cap Int \subseteq \emptyset$$

$$1 \in S$$

$$1 \in Int$$

$$1 \in S \cap Int \Leftrightarrow 1 \in S \wedge 1 \in Int$$

$$S \cap Int \subseteq \emptyset \wedge 1 \in S \cap Int \Rightarrow 1 \in \emptyset$$

Fourth Attempt—Test

$\forall a^{\mathtt{idv}}, b^{\mathtt{idv}}, x^{\mathtt{idv}} : \{a \subseteq b, x \in a\}$ $\quad a \subseteq b \wedge x \in a \Rightarrow x \in b$

$\forall a^{\mathtt{idv}}, b^{\mathtt{idv}}, x^{\mathtt{idv}} : \{x \in a, a \cap b\}$ $\quad x \in a \cap b \Leftrightarrow x \in a \wedge x \in b$

$\forall x^{\mathtt{idv}} : \{x \in \emptyset\}$ $\quad \neg(x \in \emptyset)$

$S \cap Int \subseteq \emptyset$

$1 \in S$

$1 \in Int$

$1 \in S \cap Int \Leftrightarrow 1 \in S \wedge 1 \in Int$

$S \cap Int \subseteq \emptyset \wedge 1 \in S \cap Int \Rightarrow 1 \in \emptyset$

$\neg(1 \in \emptyset)$ Contradiction; Done!

General Strategy for Selecting Triggers

- Put all $\forall$ at the top, Always provide triggers
  $\longrightarrow$ SMT will never introduce a new $\forall$ in the problem
- Include triggers for many situations. . .
  $\longrightarrow$ Reach towards *completeness* (in a loose, pragmatic sense)
- . . . but reject triggers that introduce new sets and functions
  $\longrightarrow$ Ensure *termination*

Encoding Integer Arithmetic

Isomorphishm between the TLA$^+$ set *Int* and the SMT sort `int`

The idea:

$$\mathtt{cast}_{\mathtt{int}} : \mathtt{int} \rightarrow \mathtt{idv}$$

$$\forall x^{\mathtt{idv}} : x \in \mathit{Int} \Leftrightarrow (\exists n^{\mathtt{int}} : x = \mathtt{cast}_{\mathtt{int}}(n)) \quad \text{(Surj)}$$

$$\forall m^{\mathtt{int}}, n^{\mathtt{int}} : \mathtt{cast}_{\mathtt{int}}(m) = \mathtt{cast}_{\mathtt{int}}(n) \Rightarrow m = n \quad \text{(Inj)}$$

$$\forall m^{\mathtt{int}}, n^{\mathtt{int}} : \mathtt{cast}_{\mathtt{int}}(m) + \mathtt{cast}_{\mathtt{int}}(n) = \mathtt{cast}_{\mathtt{int}}(m +_{\mathtt{int}} n) \quad \text{(Plus)}$$

Encoding Integer Arithmetic

Isomorphishm between the TLA$^+$ set *Int* and the SMT sort `int`

The implementation:

$$\mathtt{cast}_{\mathtt{int}} : \mathtt{int} \to \mathtt{idv}$$

$$\mathtt{proj}_{\mathtt{int}} : \mathtt{idv} \to \mathtt{int}$$

$$\forall x^{\mathtt{idv}} : x \in \mathit{Int} \Rightarrow x = \mathtt{cast}_{\mathtt{int}}(\mathtt{proj}_{\mathtt{int}}(x)) \qquad (\mathrm{Surj}_1)$$

$$\forall n^{\mathtt{int}} : \mathtt{cast}_{\mathtt{int}}(n) \in \mathit{Int} \qquad (\mathrm{Surj}_2)$$

$$\forall n^{\mathtt{int}} : \mathtt{proj}_{\mathtt{int}}(\mathtt{cast}_{\mathtt{int}}(n)) = n \qquad (\mathrm{Inj})$$

$$\forall m^{\mathtt{int}}, n^{\mathtt{int}} : \mathtt{cast}_{\mathtt{int}}(m) + \mathtt{cast}_{\mathtt{int}}(n) = \mathtt{cast}_{\mathtt{int}}(m +_{\mathtt{int}} n) \qquad (\mathrm{Plus})$$

Results

Specifications	Size (# obls)	Encoding Used	
		Original	New
TLA$^+$ Examples	1371	1142 35	1265 158
TLAPS Examples	666	583 16	589 22
Deconstructed Bakery	777	652 14	754 116
Total	2814	2377 65	2608 296

Observations

The performance gap comes from two specs only
Overall, the two encodings have similar performances
But the old encoding is the only one to fail on some occasions

The new SMT encoding works great on proofs with no quantifiers
Type invariants can usually be proved in a few lines

Why Do Heuristics/Triggers Work?

TLA$^+$ is very expressive, but most specifications and POs:

1. Follow the same typing conventions
2. Only use a few levels of sets
3. Do not require reasoning on unnamed sets or functions

Triggers do the same work than type synthesis / rewriting, but they "implement" it in the solver

Directions

The 86 axioms are documented but not verified...

$\longrightarrow$ Verification (Isabelle/TLA$^+$); Proof reconstruction (veriT)

Other questions:

- Are there completeness results for some fragments of TLA$^+$?
- Can we combine type synthesis and/or rewriting with the encoding based on triggers?
  (Not so clear: rewriting may erase matching terms)

Thank you!