A RODIN PLUG-IN FOR CONSTRUCTING REUSABLE SCHEMATIC LEMMAS

P. Stankaitis, A. Iliasov, D. Adjepon-Yamoah, A. Romanovsky

May 23, 2016

Newcastle University

INTRODUCTION

In this presentation we discuss:

- an approach to making proofs more generic and thus less fragile and more reusable
- a tool and verification results.

INTERACTIVE PROOFS

There is a number of circumstances when existing interactive proofs become invalidated:

- a part of the model is changed
- incremental changes that alter the goal, set of hypotheses, identifier names or types

INTERACTIVE PROOFS

There is a large number of essentially identical interactive proofs re-appearing in different projects due specific weaknesses in the underlying automatic provers.

WHY3 PLUGIN

WHY3 PLUGIN

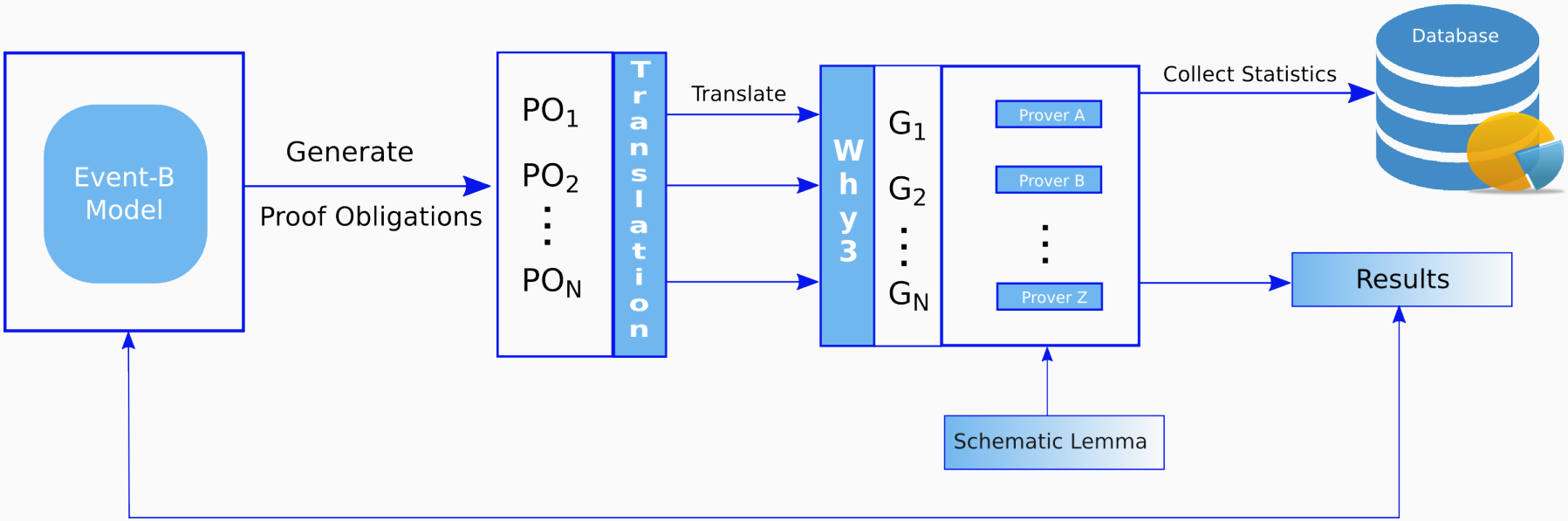

Figure: Verification tool concept

WHY3 PLUGIN

Most of the translation effort goes into the construction and fine-tuning of Why3 support theories.

WHY3 PLUGIN

Model	Proof obligations	open, Tactic[1]	open, Tactic[2]	open, Why3	open, Why3 (+ SL)
Order/Supply Communication	276	24	4	8	4 (+2)
Fisher's Algorithm	82	16	4	1	0 (+1)
Train Control System	133	36	5	32	32 (+0)
B2B Communication prot.	498	63	25	20	8 (+5)
Automated Teller Machine	962	77	28	1	0 (+1)
Total	1951	216	66	62	12

[1]Tactic - rewrite rule + nPP + PP + ML

[2]Tactic - rewrite rule + nPP + PP + ML + SMT

WHY3 PLUGIN

The initial experiments have shown that a minimal axiomatisation support is not sufficient to discharge a sizeable proportion of proof obligations.

SCHEMATIC LEMMAS

SCHEMATIC LEMMAS

A schematic lemma is a provable conjecture that does not reference any model identifiers.

SCHEMATIC LEMMAS

$$\cdots \vdash \mathsf{database} \lhd \{\mathsf{ai} \mapsto \mathsf{a}\} \in \mathsf{Attr\_id} \to \mathsf{Attrs}$$

SCHEMATIC LEMMAS

```
lemma lemma_total_overriding:
  forall f:rel 'a 'b, s:set 'a, t:set 'b, x: 'a, y : 'b.
    mem f (s --> t) /\ mem x s /\ mem y t ->
      mem (f <+ singleton (x, y)) (s --> t)
```

Figure: Schematic Lemma

SCHEMATIC LEMMAS

```
lemma lemma_total_overriding_help0:
  forall f : rel 'a 'b, x : 'a, y : 'b.
    subset  (dom f) (dom (f <+ (singleton (x, y))))

lemma lemma_total_overriding_help1:
  forall f:rel 'a 'b, s:set 'a, t:set 'b, x: 'a, y : 'b.
    mem f (s --> t) /\ mem x s /\ mem y t ->
      mem (f <+ singleton (x, y)) (s +-> t)
```

Figure: Schematic Lemmas

SCHEMATIC LEMMAS

As we have stated previously, it has been one of the goals of this research to establish to what degree schematic lemmas are reusable at least within the same project.

SCHEMATIC LEMMA

- The numbers show how each next lemma (L_1, L_2, ...) affects the overall number of open proof obligations.

Model	open, Why3	open, + L_1	open, + L_2	open, + L_3	open, + L_4	open, + L_5
B2B Communication prot.	20	16	14	12	10	8

SCHEMATIC LEMMAS

```
lemma lemma_natural_increment:
forall x, n : int.
mem x bnatural1 /\ n >= 0 ->
mem (x + n) bnatural1
```

Figure: Schematic Lemma

SCHEMATIC LEMMAS

```
lemma lemma_finite_partial_domain:
  forall f : rel 'a 'b, s : set 'a, t : set 'b.
    finite (dom f) /\ mem f (s +-> t) ->
      finite f
```

Figure: Schematic Lemma

SCHEMATIC LEMMA

There is a fine interplay between the functioning of the schematic lemmas plug-in and the Why3 plug-in filtering mechanism.

SCHEMATIC LEMMA TOOL

SCHEMATIC LEMMA TOOL

- It integrates into the prover perspective and offers an alternative way to conduct an interactive proof.

- The Why3 notation is employed, but the first release will support entering schematic lemma in the Event-B mathematical notation.

SCHEMATIC LEMMA TOOL

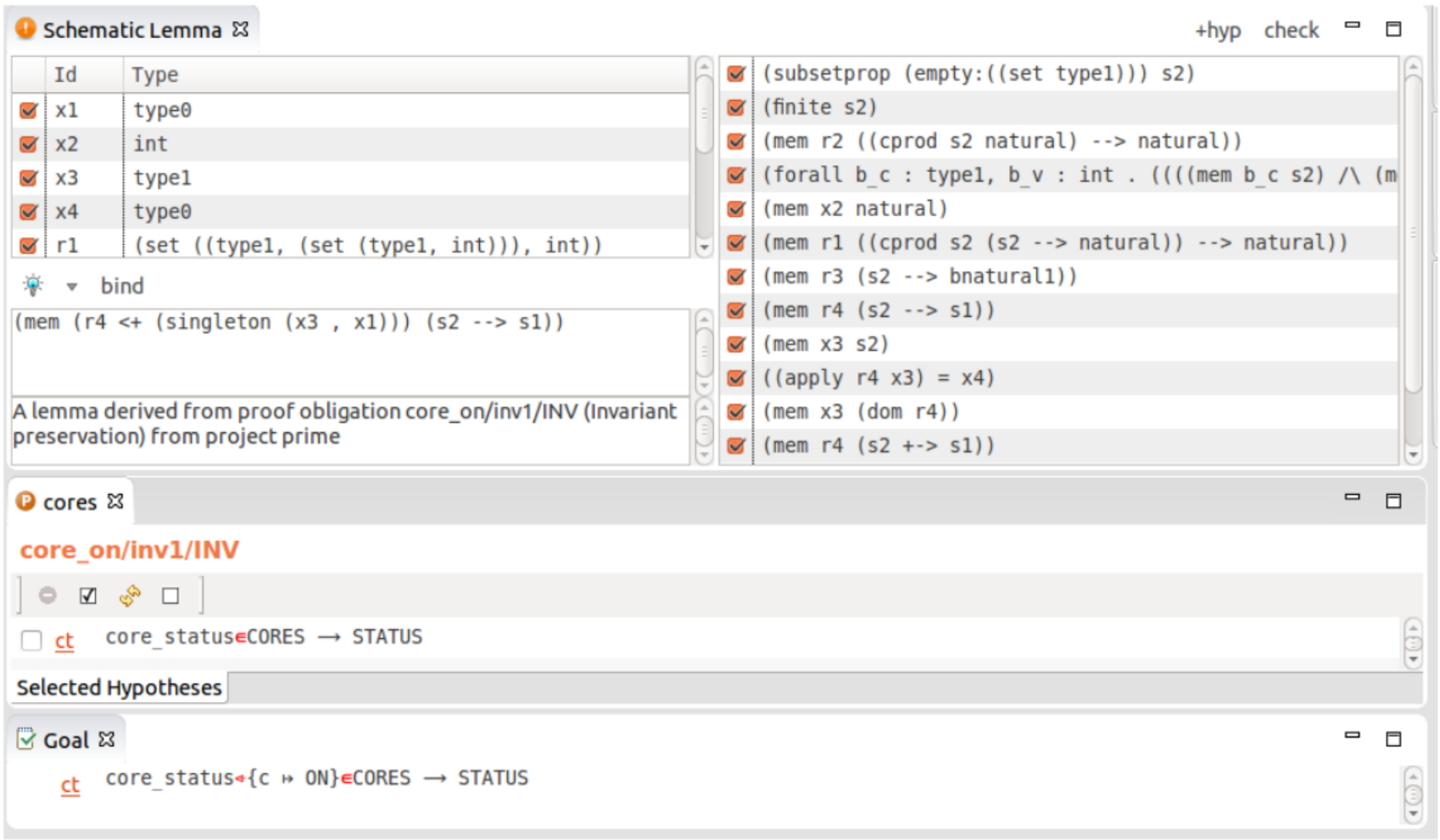

Figure: Schematic Lemma Plugin

SCHEMATIC LEMMA TOOL

The plug-in automatically constructs the first attempt at a schematic lemma through a simple syntactic transformation of a context proof obligation.

SCHEMATIC LEMMA TOOL

From this starting point it is up to the modeller to construct a sensible lemma by changing identifier, hypotheses and goal definitions.

CONCLUSION & FUTURE WORK

FUTURE WORK

- Applying schematic lemma technique to a large number of models.
- Release the first version of the plugin.

CONCLUSIONS

- The idea of generalisation for the purpose of proof reuse has been explored in different settings.
- We also hypothesise that at a certain stage accumulated schematic lemma make automatic proof support so complete that interactive proofs are no longer necessary.

Questions?