Investigating Higher-order Simplification for SMT Solving

work in progress

Hans-Jörg Schurr

26.06.2018

What this is about

The goal of the Matryoshka is to improve the usefulness of automated tools in interactive theorem proving by strengthening their higher-order capabilities.

What this is about

The goal of the Matryoshka is to improve the usefulness of automated tools in interactive theorem proving by strengthening their higher-order capabilities.

The Isabelle *simplifier* provides a basic form of automatization.

Examples

```
fun add :: "nat ⇒ nat ⇒ nat" where
  "add 0 n = n" |
  "add (Suc m) n = Suc(add m n)"

lemma add_02: "add m 0 = m"
 apply(induction m)
   apply(simp)
  apply(simp)
 done
```

What we are looking at

- Isabelle interfaces with automatic systems via the *Sledgehammer* tool
- How does the simplifier compare to Sledgehammer?

What we are looking at

- Isabelle interfaces with automatic systems via the *Sledgehammer* tool
- How does the simplifier compare to Sledgehammer?
- The *Mirabelle* tool is a tool included with the Isabelle distribution for experiments with Sledgehammer
- Goals are considered trivial if they are also solved by `try0`
- We added an additional parameter to run the simplifier only

Numbers

- We ran this on 64 randomly chosen AFP entries
- Configuration: Timeout of 20s, default provers
- Overall 1973 Sledgehammer calls
- Sledgehammer succeeded 1223 (62%) times
- The simplifier solved 344 (17%) problems and 45 (2.3%) exclusively

Numbers with comments

- In an earlier test we identified 24 simp only problems and investigated them further
- 13 were solved when reproducing locally with default Sledgehammer settings
- Many were some form of computation (i.e. in theories about cryptography or functional programming)
- Example problem `Tycon/Lazy_List_Monad.thy:coerce_LNi`[1]:
  - Fails when naively applying Sledgehammer
  - The simplifier uses four rules
  - Sledgehammer succeeds when adding these rules, but took a while
  - Only exporting those rules results in a easy problem

[1] AFP 2018-05-09

The Isabelle simplifier (Nipkow 1989)

- User defined set of rewriting rules
  - automatically from e.g. function definitions
  - Lemmas annotated with `[simp]`
  - When calling the simplifier (also allows restricting the set)
- This should be confluent and terminating
  - In practice both often does not hold
  - but it doesn't matter much for the user
- Workings in HOL: no encoding needed.
- Some extensions: Conditional, local assumptions, *simp progs*

The Isabelle simplifier

Main task of the Isabelle simplifier: Express rewriting in the LCF-Style framework of Isabelle.

The Isabelle simplifier

Main task of the Isabelle simplifier: Express rewriting in the LCF-Style framework of Isabelle.

Utilizes lazy lists to express the possible infinite number of unifiers.

(First-order) rewriting and SMT

- Exported simp annotations have been used in the superposition prover SPASS (Blanchette et al. 2012)
- SMT however is not fundamentally based on rewriting, but ground reasoning + instantiation
- Some current usage:
  1. as pre-processing
  2. as part of the theory reasoner (see next slide)
  3. Z3 supports explicit simplification commands (Moura and Passmore 2013)

Context-dependent Simplification (Reynolds et al. 2017)

Solve some extension of a theory by using qualities entailed by the base theories together with simplification rules.

Context-dependent Simplification (Reynolds et al. 2017)

Solve some extension of a theory by using qualities entailed by the base theories together with simplification rules.

- $\{z = y \times y, y = w + 2, w = 1\}$

Context-dependent Simplification (Reynolds et al. 2017)

Solve some extension of a theory by using qualities entailed by the base theories together with simplification rules.

- $\{z = y \times y, y = w + 2, w = 1\}$
- $M = \{z = x, y = w + 2, w = 1\}$, $X(M) = \{x = y \times y\}$

Context-dependent Simplification (Reynolds et al. 2017)

Solve some extension of a theory by using qualities entailed by the base theories together with simplification rules.

- $\{z = y \times y, y = w + 2, w = 1\}$
- $M = \{z = x, y = w + 2, w = 1\}$, $X(M) = \{x = y \times y\}$
- $M \models_A y = 3$ and we get $\sigma = \{y \mapsto 3\}$

Context-dependent Simplification (Reynolds et al. 2017)

Solve some extension of a theory by using qualities entailed by the base theories together with simplification rules.

- $\{z = y \times y, y = w + 2, w = 1\}$
- $M = \{z = x, y = w + 2, w = 1\}$, $X(M) = \{x = y \times y\}$
- $M \models_A y = 3$ and we get $\sigma = \{y \mapsto 3\}$
- $(y \times y)\sigma \downarrow = (3 \times 3) \downarrow = 9$

Then apply the solver for the base theory.

Onward

Some ideas:

- Naive: Use some rules as pre-processing and get some potentially helpful new rules
- Handling quantifiers in SMT is still an area of active research
- Even more important as we move to HOL

Onward

Some ideas:

- Naive: Use some rules as pre-processing and get some potentially helpful new rules
- Handling quantifiers in SMT is still an area of active research
- Even more important as we move to HOL

Can we separate some quantified input formulas into a *simpset* and use this for some form of rewriting instead of instantiation?

References I

Nipkow, Tobias (1989). "Equational reasoning in Isabelle". In: *Science of Computer Programming* 12.2, pp. 123–149. ISSN: 0167-6423. DOI: 10.1016/0167-6423(89)90038-5.

Blanchette, Jasmin Christian et al. (2012). "More SPASS with Isabelle". In: *Interactive Theorem Proving*. Ed. by Lennart Beringer and Amy Felty. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 345–360. ISBN: 978-3-642-32347-8.

Moura, Leonardo de and Grant Olney Passmore (2013). "The Strategy Challenge in SMT Solving". In: *Automated Reasoning and Mathematics: Essays in Memory of William W. McCune*. Ed. by Maria Paola Bonacina and Mark E. Stickel. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 15–44. ISBN: 978-3-642-36675-8. DOI: 10.1007/978-3-642-36675-8_2.

References II

Reynolds, Andrew et al. (2017). "Designing Theory Solvers with Extensions". In: *Frontiers of Combining Systems*. Ed. by Clare Dixon and Marcelo Finger. Cham: Springer International Publishing, pp. 22–40. ISBN: 978-3-319-66167-4.