



The Use of Rippling to Automate Event-B Invariant Preservation Proofs

Yuhui Lin, Alan Bundy & Gudmund Grov

School of Informatics
University of Edinburgh

NFM 2012



Outline

In this talk we

- show that a proof technique called *rippling* is applicable to Event-B invariant proofs (1st order logic)
- outline a novel approach combining rippling with theory formation to automate lemma discovery

Event-B invariant proofs and rippling

- We have observed that
 - the majority of proofs requiring automation are invariant proofs (e.g. 59% in one case study)
 - Invariant proofs typically follow a pattern where one of the assumptions is embedded in the goal, i.e. $f(x) \vdash f(g(x))$
- Rippling guides
 - the search by writing the goal until the assumption appears as a sub-formulae, e.g.
$$f(x) \vdash f(g(x)) \quad f(g(x)) = h(f(x)) \quad \longrightarrow \quad f(x) \vdash h(f(x))$$
 - Allows rewrite rules in both directions without loss of termination
 - It may automate many interactive Rodin proofs

Lemma discovery in rippling

- The key feature of rippling is the ability to automatically patch failed proofs via *critics* when required lemmas are not present
 - increases robustness and automation
 - due to the strong expectation on the proof
- Suppose a simplified version of our proof is blocked at:

$$T = \text{dom}(R \cup S ; f)$$

- We can then follow a 4 step process which discovers the missing lemma

$$(A \cup B); C = A; C \cup B; C$$

Lemma discovery steps

1. Generate the left hand side: pick terms of blocked goals which are expected to change in the next rewriting step, e.g.

$$R \cup S ; f$$

2. Conjecture right hand side scheme: we know that the right hand side must have the shape

$$F_1(R ; f)(F_2 S f)$$

Where F_n is a 2nd order placeholders

- since skeleton must be preserved
- wave-front must move outwards.,

Lemma discovery steps

3. Instantiate scheme: then feed the scheme

$R \cup S$; $f = F_1(R ; f)(F_2 S f)$ to IsaScheme,

- which is a tool which discovers conjectures based on a given scheme and set of terms
- with counter-examples checks
- with proof attempts

4. Post filter & prove: one of the “sensible” instantiations is $(R \cup S) ; f = (R ; f) \cup (S ; f)$. This can be proven automatically (by Isabelle), but in more complex cases the process recurses or the user must provide a proof



Conclusion and further work

- We have shown
 - that rippling is applicable to Event-B invariant POs
 - a new technique to help discover missing lemmas
- We are currently implementing the automation of the lemma discovery process.