Proof Hints for Event-B

Thai Son Hoang

Institute of Information Security
Swiss Federal Institute of Technology Zürich (ETH Zürich)

Rodin Workshop, Fontainebleau, France
28th-29th February 2012
Proof obligations are generated from formal models.

- Failed proof attempts required to the models to be fixed.

- How about successful attempts, in particular, interactive proofs?
Interactive Proofs v.s Automatic Proofs

- Maintenance of interactive proofs is difficult.
- Better rate of automatic proofs
  - Better automatic provers (Isabelle, SMT)
  - Better proof profiles.
  - This talk: “Improve” the existing model.

Idea

Expose more proof information in the model: “proof hints”
Existing Proof Hints in Event-B/Rodin Platform

- **Theorems** (add hypothesis)
- **Witnesses** (existential instantiation)
- **Guard selection** (select hypotheses)
Hypotheses Selection (1/2)

**invariants:**

- **inv1:** $x \in \mathbb{N}$
- **inv2:** $x \neq 0 \Rightarrow y \in \mathbb{N}$

**set**

```plaintext
when
  grd1: $x \in \{1, 2\}$
then
  act1: $x := y + 1$
end
```

Selected hypotheses: **inv1** and **grd1**

**inv2** is required, added as a guard theorem.
Selected hypotheses: \textbf{inv1} and \textbf{grd1}

- \textbf{inv2} is required, added as a guard theorem.
Selected hypotheses: **inv1** and **grd1**

**inv2** is required, added as a guard theorem.
Hypotheses Selection (1/2)

invariants:
- inv1: \( x \in \mathbb{N} \)
- inv2: \( x \neq 0 \Rightarrow y \in \mathbb{N} \)

\[
\begin{align*}
\text{set} & \\
\text{when} & \\
\text{grd1}: & \quad x \in \{1, 2\} \\
\text{thm1}: & \quad x \neq 0 \Rightarrow y \in \mathbb{N} \\
\text{then} & \\
\text{act1}: & \quad x := y + 1 \\
\text{end} & \\
\end{align*}
\]

Selected hypotheses: inv1 and grd1

inv2 is required, added as a guard theorem.
Hypotheses Selection (2/2)

Cons for using theorem

- Copy/paste.
- An extra proof obligation (trivially discharged).
Hypotheses Selection (2/2)

set

when
  grd1 : x ∈ \{1, 2\}
  thm1 : x ≠ 0 ⇒ y ∈ \mathbb{N}
then
  act1 : x := y + 1
end

Cons for using theorem

- Copy/paste.
- An extra proof obligation (trivially discharged).
Do Case (1/3)

invariants:

inv1: \( a \leq c \)

inv2: \( a \neq 1 \Rightarrow b = a + 1 \)

inv3: \( a = 1 \Rightarrow b \leq c \)

set

begin

\( a := b - 1 \)

end

\[ a \leq c \]
\[ a \neq 1 \Rightarrow b = a + 1 \]
\[ a = 1 \Rightarrow b \leq c \]

\[ \Downarrow \]

\[ b - 1 \leq c \]

set/inv1/INV

Proof by cases:

- \( a = 1 \)
- \( a \neq 1 \)
Do Case (2/3)

set_a

when
    \( a = 1 \)
then
    \( a := b - 1 \)
end

set_b

when
    \( a \neq 1 \)
then
    \( a := b - 1 \)
end

set

refines set_a, set_b

begin
    \( a := b - 1 \)
end

- Duplication of proof obligations.
- Artificial merging step.
Do Case (3/3)

```
set
begin
  a := b
  case-split
  a = 1 for inv1
end
```
Using information of interactive proofs to “improve” the model.

**Hints** (proof information) help with **proof automation**.

Hints help to **understand model better**.

How far should we go in terms of exposing proof information in the model?

A plug-in (a reasoner) that “interprete” proof hints.