Report

Rodin's unsoundness bugs

Author(s):
Schmalz, Matthias

Publication Date:
2011

Permanent Link:

https://doi.org/10.3929/ethz-a-006886149 →

Rodin's Unsoundness Bugs

Matthias Schmalz

March 3, 2011

This document provides a commented list of Rodin's unsoundness bugs. We do not consider bugs in the documentation that have never affected the implementation. See [1] for the notation used in this document as well as a detailed discussion of syntax, semantics, and proofs in Event-B.

1 Rodin's Built-in Theorem Prover

Discovery date: 10/10/2007
Discovered by: Thai Son Hoang
Version: 0.8.0
Url: http://sourceforge.net/tracker/?func=detail&aid=1811032&group_id=108850&atid=651669
Description: Some proof obligation does not update properly when the model changes. It may thus happen that all proof obligations are discharged although the (changed) model is inconsistent.

Discovery date: 12/12/2007
Discovered by: Pierre Casteran
Version: 0.8.1
Url: http://sourceforge.net/tracker/?func=detail&aid=1849522&group_id=108850&atid=651669
Description: The prover implements the following unsound rewrite rules:

$$\mathrm{card}(\$R) \leq \mathrm{card}(\$S) \sqsubseteq \$R \subseteq \$S$$

$$\mathrm{card}(\$R) \geq \mathrm{card}(\$S) \sqsubseteq \$S \subseteq \$R$$

$$\mathrm{card}(\$R) < \mathrm{card}(\$S) \sqsubseteq \$R \subset \$S$$

$$\mathrm{card}(\$R) > \mathrm{card}(\$S) \sqsubseteq \$S \subset \$R$$

$$\mathrm{card}(\$R) = \mathrm{card}(\$S) \sqsubseteq \$R = \$S$$

Discovery date: 25/11/2008
Discovered by: Laurent Voisin
Version: 0.9.0
Url: http://sourceforge.net/tracker/?func=detail&aid=2343565&group_id=108850&atid=651669
Description: The intersection of a carrier set with itself is rewritten to the empty set. This is unsound, as carrier sets never denote empty sets.

Discovery date: 10/11/2009
Discovered by: Matthias Schmalz, Laurent Voisin
Version: 1.1
Url: http://sourceforge.net/tracker/?func=detail&aid=2895507&group_id=108850&atid=651669
Description: The prover implements the following rules, which are unsound:

$$\frac{\vdash \$\psi_1 \qquad \$\psi_1, \$\psi_2 \vdash \$\varphi}{\$\psi_1 \Rightarrow \$\psi_2 \vdash \$\varphi}\ \mathsf{MH}$$

$$\frac{\vdash \$R \subseteq \$S \qquad \vdash \mathrm{finite}(\$S)}{\vdash \mathrm{finite}(\$R)}\ \mathsf{FIN\_SUBSETEQ\_R}$$

To understand why these rules are unsound, one needs to know that a formula denotes true, false, or is ill-defined. A sequent denotes either true or false. A sequent is false iff all hypotheses denote true and the goal denotes false. The formula $\psi_1 \Rightarrow \psi_2$

- denotes true iff ψ_1 denotes false or ψ_2 denotes true,
- denotes false iff ψ_1 denotes true and ψ_2 denotes false, and
- is ill-defined otherwise.

Consider MH and suppose that $\$\psi_1$ is ill-defined, $\$\psi_2$ denotes true, and $\$\varphi$ denotes false. Then the consequent denotes false, but all antecedents denote true. Concerning FIN_SUBSETEQ_R suppose that $\$R$ denotes an infinite set and $\$S$ is ill-defined. Then again all antecedents denote true and the consequent denotes false.

The prover also implements the following unsound rewrite rule:

$$\mathrm{card}(\$R \setminus \$S) \sqsubseteq \mathrm{card}(\$R) - \mathrm{card}(\$S) \quad \mathsf{SIMP\_CARD\_SETMINUS.}$$

Roughly spoken, a rewrite rule is sound iff well-definedness of the left-hand side implies that the right-hand side has the same denotation of the left-hand side. Concerning SIMP_CARD_SETMINUS, let $\$R$ and $\$S$ denote the same infinite set. Then the left-hand side is well-defined, but the right-hand side is not and therefore unequal to the left-hand side. (The term $\mathrm{card}(\$R)$ is well-defined only if $\$R$ denotes a finite set.)

According to the bug report, Rodin 1.1 implements more unsound rules. We list them without further comments:

$$\frac{\vdash \neg\$\psi_2 \qquad \neg\$\psi_1, \neg\$\psi_2 \vdash \$\varphi}{\$\psi_1 \Rightarrow \$\psi_2 \vdash \$\varphi}\ \mathsf{HM}$$

$$\frac{\vdash \$r \in \$R \leftrightarrow \$S \qquad \vdash \mathrm{finite}(\$R) \qquad \vdash \mathrm{finite}(\$S)}{\vdash \mathrm{finite}(\$r)}\ \mathsf{FIN\_REL\_R}$$

$$\frac{\vdash \$r \in \$R \nrightarrow \$S \qquad \vdash \mathrm{finite}(\$R)}{\vdash \mathrm{finite}(\$r)}\ \mathsf{FIN\_FUN1\_R}$$

$$\frac{\vdash \$r\sim \in \$R \nrightarrow \$S \qquad \vdash \mathrm{finite}(\$R)}{\vdash \mathrm{finite}(\$r)}\ \mathsf{FIN\_FUN2\_R}$$

$$\frac{\vdash \$r \in \$S_1 \nrightarrow \$S_2 \qquad \vdash \mathrm{finite}(\$R)}{\vdash \mathrm{finite}(\$r[\$R])}\ \mathsf{FIN\_FUN\_IMG\_R}$$

$$\frac{\vdash \$r \in \$R \nrightarrow \$S \qquad \vdash \mathrm{finite}(\$R)}{\vdash \mathrm{finite}(\mathrm{ran}(\$r))}\ \mathsf{FIN\_FUN\_RAN\_R}$$

$$\frac{\vdash \$r\sim \in \$R \nrightarrow \$S \qquad \vdash \mathrm{finite}(\$R)}{\vdash \mathrm{finite}(\mathrm{dom}(\$r))}\ \mathsf{FIN\_FUN\_DOM\_R}$$

$$\frac{\vdash \mathrm{WD}(\$y) \qquad \mathrm{WD}(\$y) \vdash \$\psi_1(\$y) \qquad \mathrm{WD}(\$y), \$\psi_1(\$y), \$\psi_2(\$y) \vdash \$\varphi}{\forall x \cdot \$\psi_1(x) \Rightarrow \$\psi_2(x) \vdash \$\varphi}\ \mathsf{FORALL\_INST\_MP}$$

$$(\$R \cup \$S) \lhd \$r \sqsubseteq \$R \lhd \$r \cup \$S \lhd \$r \quad \mathsf{DISTRI\_DOMSUB\_BUNION\_L}$$

$$(\$R \cap \$S) \lhd \$r \sqsubseteq \$R \lhd \$r \cap \$S \lhd \$r \quad \mathsf{DISTRI\_DOMSUB\_BINTER\_L}$$

$$\$r \rhd (\$R \cup \$S) \sqsubseteq \$r \rhd \$R \cup \$r \rhd \$S \quad \mathsf{DISTRI\_RANSUB\_BUNION\_R}$$

$$\$r \rhd (\$R \cap \$S) \sqsubseteq \$r \rhd \$R \cap \$r \rhd \$S \quad \mathsf{DISTRI\_RANSUB\_BINTER\_R}$$

$$\mathrm{card}(\$R \times \$S) \sqsubseteq \mathrm{card}(\$R) * \mathrm{card}(\$S) \quad \mathsf{SIMP\_CARD\_CPROD}$$

The wiki page (http://wiki.event-b.org/index.php?title=Inference_Rules&oldid=4592) also states that CARD_EMPTY_INTERV and CARD_SUBSETEQ have been implemented unsoundly, but we were unable to retrieve the unsound versions of these rules.

Discovery date: 26/01/2010
Discovered by: Laurent Voisin
Version: 1.2RC1
Url: http://sourceforge.net/tracker/?func=detail&aid=2940139&group_id=108850&atid=651669
Description: The reasoner "org.eventb.core.seqprover.totalDom:0" does not properly report the hypotheses it uses (i.e., hypotheses of the form $f \in S \nrightarrow T$). When such a

hypothesis disappears from the sequent, because the user has changed the model, the proof obligation is still displayed as "proved", although the proof is no longer correct.

Discovery date: 15/02/2010
Discovered by: Matthias Schmalz
Version: 1.2
Url: http://sourceforge.net/tracker/?func=detail&aid=2952087&group_id=108850&atid=651669
Description: Rodin 1.2 accepts unsound proofs of Rodin 1.1.

Discovery date: 05/03/2010
Discovered by: Thai Son Hoang
Version: 1.2
Url: http://sourceforge.net/tracker/?func=detail&aid=2964360&group_id=108850&atid=651669
Description: The prover implements the following rule

$$\frac{\$x \in \mathrm{dom}(\$s) \vdash \$\varphi(\$s(\$x)) \qquad \neg \$x \in \mathrm{dom}(\$s) \vdash \$\varphi(\$r(\$x))}{\vdash \$\varphi(\$r \lhd\!\!\!\!- \$s(\$x))}\ \mathsf{OV\_R}.$$

The rule has a side-condition ensuring that ill-definedness of $\$r$, $\$s$, or $\$x$ implies ill-definedness of $\$\varphi(\$r \lhd\!\!\!\!- \$s(\$x))$. Yet, the rule is unsound: let $\$r = \{0 \mapsto 0, 0 \mapsto 1, 1 \mapsto 0\}$, $\$s = \{0 \mapsto 0\}$, $\$x = 1$, and $\$\varphi(\$x) = (\$x = 1)$. Then the rule becomes

$$\frac{1 \in \mathrm{dom}(\{0 \mapsto 0\}) \vdash \ldots \qquad \cdots \vdash \{0 \mapsto 0, 0 \mapsto 1, 1 \mapsto 0\}(1) = 1}{\vdash (\{0 \mapsto 0, 0 \mapsto 1, 1 \mapsto 0\} \lhd\!\!\!\!- \{0 \mapsto 0\})(1) = 1}\ .$$

The first antecedent denotes true, because its hypothesis denotes false. The second antecedent denotes true, because the goal is ill-defined; note that $\{0 \mapsto 0, 0 \mapsto 1, 1 \mapsto 0\}(1)$ is ill-defined, because $\{0 \mapsto 0, 0 \mapsto 1, 1 \mapsto 0\}$ is not functional (0 has two images). The consequent denotes false, because it can be simplified to $\vdash \{0 \mapsto 0, 1 \mapsto 0\}(1) = 1$.

There is another rule called OV_R and there are another two rules called OV_L, which are also unsound. See http://wiki.event-b.org/index.php?title=Inference_Rules&oldid=5284 for their definitions.

Discovery date: 03/05/2010
Discovered by: Louis Mussat
Version: 1.3
Url: http://sourceforge.net/tracker/?func=detail&aid=3025836&group_id=108850&atid=651669
Description: The term

$$(\lambda x \cdot x \subseteq R \mid (\lambda y \mapsto z \cdot y \in x \land z \subseteq x \mid z))([[0]])$$

is simplified to $(\lambda y \mapsto z \cdot y \in z \land z \subseteq z \mid z)$ instead of $(\lambda y \mapsto z \cdot y \in [[2]] \land z \subseteq [[2]] \mid [[2]])$. Here $[[0]]$ and $[[2]]$ are the bound variables with de Bruijn index 0 and 2, respectively.

Discovery date: 06/07/2010
Discovered by: Matthias Schmalz
Version: 1.3.1
Url: http://sourceforge.net/tracker/?func=detail&aid=3025836&group_id=108850&atid=651669
Description: The term

$$\{x \cdot x \in \mathbb{Z} \mid x \mapsto x\}([[2]])$$

is simplified to [[1]] instead of [[2]]. Here [[1]] and [[2]] are bound variables with de Bruijn index 1 and 2, respectively. The reason is an incorrect fix of the bug from 03/05/2010.

Discovery date: 12/07/2010
Discovered by: Laurent Voisin
Version: 1.3.1
Url: http://sourceforge.net/tracker/?func=detail&aid=3028473&group_id=108850&atid=651669
Description: The reasoner contrHyps incorrectly reports the hypotheses it uses. If the user changes the model, unproved proof obligation may thus be displayed as proved.

Discovery date: 02/11/2010
Discovered by: Nicolas Beauger
Version: 2.0
Url: http://sourceforge.net/tracker/?func=detail&aid=3102302&group_id=108850&atid=651669
Description: PP and ML discharge any sequent of the form

$$\vdash f(t) = t,$$

where f is an arbitrary operator defined by mathematical extensions and t an arbitrary term of appropriate type.

Discovery date: 14/01/2011
Discovered by: Matthias Schmalz
Version: 2.0.1
Url: http://sourceforge.net/tracker/?func=detail&aid=3158594&group_id=108850&atid=651669
Description: "Simplification rewrites" incorrectly rewrites

$$0 \mapsto 0 \in \{x \cdot \exists y \cdot y * y < 0 \land y = 1 \div 0 \mid x\}$$

to

$$1 \div 0 * (1 \div 0) < 0,$$

i.e., the one-point rule is applied although its precondition is violated. That rewrite step can be used to construct a proof of $\bot$.

2 New PP

Technically, New PP is shipped with the Rodin platform and is therefore a built-in theorem prover. Yet, we list the bugs in New PP within a separate section, because they can be avoided by not using New PP.
Discovery date: 20/03/2008
Discovered by: Louis Mussat
Version: 0.8.2
Url: http://sourceforge.net/tracker/?func=detail&aid=1920747&group_id=108850&atid=651669
Description: The sequent

$$\begin{array}{l} x \in R_1, \\ R_1 \subseteq S, \\ R_2 \subseteq S \\ \vdash \\ x \in R_2 \end{array}$$

is discharged. (The set S is a carrier set.)

Discovery date: 15/02/2010
Discovered by: Laurent Voisin
Version: 1.2, 1.3, 1.3.1, 2.0
Url: http://sourceforge.net/tracker/?func=detail&aid=2952091&group_id=108850&atid=651669
Description: New PP discharges

$$\vdash (P = \text{TRUE} \lor R = \text{TRUE}) \Leftrightarrow (P = \text{TRUE} \lor Q = \text{TRUE}).$$

The bug could not be fixed so far.

Discovery date: 10/10/2010
Discovered by: Alexei Iliasov
Version: 2.0
Url: http://sourceforge.net/tracker/?func=detail&aid=3085103&group_id=108850&atid=651669
Description: The sequent

$$\begin{array}{l} y \in \text{BOOL}, \\ x = \text{FALSE}, \\ y = z \\ \vdash \\ x = \text{TRUE} \Leftrightarrow y = z \end{array}$$

is discharged.

Discovery date: 04/11/2010
Discovered by: Hector Ruiz
Version: 2.0
Url: http://sourceforge.net/tracker/?func=detail&aid=3102775&group_id=108850&atid=651669
Description: New PP discharges

$$
\begin{array}{l}
r \in \mathrm{BOOL} \to \{0,1\},\\
r(\mathrm{TRUE}) = 0,\\
r(\mathrm{FALSE}) = 1,\\
\vdash\\
\bot
\end{array}
$$

3 Theory Plug-in (Formerly "Rule-Based Prover")

Discovery date: 05/03/2010
Discovered by: Matthias Schmalz
Version: 1.3, 1.3.1, 2.0
Url: http://sourceforge.net/tracker/?func=detail&aid=2964359&group_id=108850&atid=651669
Description: The rule-based prover allows one to introduce the unsound rewrite rule

$$\neg 1 \div \$x = 1 \div \$x \Rightarrow \neg \$x \neq 0 \sqsubseteq \$x \neq 0 \quad .$$

The rule is unsound, because when $\$x$ denotes 0, the left-hand side denotes true and the right-hand side denotes false. The rule is accepted, because the rule-based prover asks one to prove

$$\mathrm{WD}(\neg 1 \div x = 1 \div x \Rightarrow \neg x \neq 0) \vdash \mathrm{WD}(x \neq 0) \wedge ((\neg 1 \div x = 1 \div x \Rightarrow \neg x \neq 0) \Leftrightarrow (x \neq 0)),$$

which is equivalent to

$$x \neq 0 \vdash \top \Leftrightarrow x \neq 0.$$

A correct proof obligation would have been

$$\mathrm{D}(\neg 1 \div x = 1 \div x \Rightarrow \neg x \neq 0) \vdash \mathrm{D}(x \neq 0) \wedge ((\neg 1 \div x = 1 \div x \Rightarrow \neg x \neq 0) \Leftrightarrow (x \neq 0)),$$

which is equivalent to

$$\vdash \top \Leftrightarrow x \neq 0.$$

Note that $\mathrm{WD}(t)$ entails $\mathrm{D}(t)$, but $\mathrm{D}(t)$ does not entail $\mathrm{WD}(t)$. It is thus in general incorrect to replace D by WD.

It has been claimed that this bug has been fixed in Rodin 2.0. But the theory plug-in accompanying Rodin 2.0 still has this bug.

Discovery date: 10/11/2010
Discovered by: Matthias Schmalz
Version: 2.0
Url: https://sourceforge.net/tracker/index.php?func=detail&aid=3106728&group_id=108850&atid=651669
Description: The theory plug-in allows one to introduce the rule

$$\top \lor \$x = \$x \sqsubseteq \$x = \$x.$$

The rule is unsound, because the left-hand side is always well-defined, but the right-hand is not. The theory plug-in creates the proof obligation

$$\forall x \cdot \top \lor x = x \Leftrightarrow x = x,$$

which is provable. A correct proof obligation would have been

$$\mathrm{D}(\top \lor \$x = \$x) \vdash \mathrm{D}(\$x = \$x) \land (\top \lor \$x = \$x \Leftrightarrow \$x = \$x),$$

which is equivalent to

$$\vdash \mathrm{D}(\$x).$$

Operator variables (such as $\$x$) are not supported by Rodin, but one could replace $\$x$ by a "fresh" (and possibly ill-defined) constant.

References

[1] M. Schmalz. The logic of Event-B. Technical Report 698, ETH Zurich, Switzerland, 2010. http://www.inf.ethz.ch/research/disstechreps/techreports.