What's new in Rodin 3.9 and the Theory plug-in

Guillaume Verdier[1], Laurent Voisin[2], Idir Ait-Sadoune[3]

[1] UPEC
guillaume.verdier@irit.fr
[2] Systerel
laurent.voisin@systerel.fr
[3] Paris-Saclay University, CentraleSupélec, LMF laboratory
idir.aitsadoune@centralesupelec.fr

1 Introduction

The Rodin platform [1] is an integrated development environment for designing software with Event-B [2]. Thanks to support from the French ANR project Event-B Rodin Plus (EBRP, ANR-19-CE25-0010), Rodin and the Theory plug-in are actively updated with bug fixes and implementation of feature requests. We present the evolution of the Rodin platform since ABZ 2023 and provide some news on the ongoing effort to improve the Theory plug-in [3].

2 Rodin 3.9

A release candidate for Rodin 3.9 was released on April 23rd, 2024 and the final release will be published around mid-June, just before ABZ 2024.

Many new proof rules have been implemented:

- several rules have been added to the auto-rewriter:
  - $\min(A) \in A \equiv \top$
  - $\max(A) \in A \equiv \top$
  - $\text{bool}(B = \text{TRUE}) \equiv B$
  - $E \mapsto E \in \text{id} \equiv \top$
  - $E \mapsto E \in r \setminus \text{id} \equiv \bot$
  - $E \mapsto E \in S \lhd \text{id} \equiv E \in S$
  - $E \mapsto E \in r \setminus (S \lhd \text{id}) \equiv E \mapsto E \in S \lhd r$
- some auto-rewriter rules can also be applied manually:
  - $F \in \{x, y \cdot P(x, y) \mid E(x, y)\} \equiv \exists x, y \cdot P(x, y) \land E(x, y) = F$
  - $E \in \{x \cdot P(x) \mid x\} \equiv P(E)$
- some new manual rules have been added:
  - $f(x) = y \equiv x \mapsto y \in f$
  - rewrite a^n to $a \times a^{n-1}$ with an additional sub-goal $n \neq 0$
  - infer $\text{finite}(\{i \cdot P(i) \mid F(i)\})$ from $\text{finite}(\{i \mid P(i)\})$
- new reasoners have been created on inductive types:
  - $datatype(T1, U1, ...) = datatype(T2, U2, ...) \equiv T1 = T2 \land U1 = U2 \land ...$
  - $cons(a, b, ...) \in datatype(T, ...) \equiv a \in destr1set(T, ...) \land b \in destr2set(T, ...) \land ...$

It is now possible to provide names for new identifiers generated during proofs. For *abstract expression*, one can write *ident* = *expr* instead of just *expr*. For *universal quantification introduction*, *existential quantification elimination* and *datatype distinct case*, a comma-separated list of identifiers can be provided in the proof control input. If the provided identifiers are not fresh, they will be used as a base to generate fresh names.

Among bugs fixed, there have been two major ones.

In Rodin 3.8, feature request #371 introduced hiding of rewritten equality identifiers. However, only selected hypotheses were rewritten: the identifier could still appear in default hypotheses, while its equality had been hidden. Now, the identifier is either deselected or hidden depending on whether it appears in default hypotheses or not.

Yannis Benabbi found a breaking bug in the auto rewriter. In some rare cases with nested comprehension sets, the auto rewriter could "prove" a false goal. The bug has been fixed and the auto rewriter's version incremented.

Besides these bugs, miscellaneous crashes and issues related to exception handling have been fixed. Also, the "Prove automatically" setting is now persisted correctly.

3 Theory plug-in

Before the start of the EBRP project, a release candidate for version 4 had been released in 2017, without a final release. A final release of version 4 with a few more bug fixes was released on December 22, 2020, followed by three more bug-fix releases in 2021 and 2022. The remaining issues in the Theory plug-in are complex ones that require more important changes. A large refactoring has been started and is ongoing.

First, the handling of formula factories and extensions (created to represent constructions of the Theory plug-in, such as inductive data types and operators) has been completely reworked. This should fix many issues related to incompatible formula factories being mixed up. Then, the static-checker has been cleaned up and improved, as well as the representation of elements in the Rodin database. The proof obligation generator is the main remaining part that has to be worked on. It will require quite some work as there are many problems with it, particularly in relation to proof rules.

4 Conclusion

Rodin is under active development and new versions are released yearly. The Theory plug-in, which had many issues, is also being updated. It is now going through an important refactoring to fix most complex underlying issues.

References

[1] Jean-Raymond Abrial, Michael Butler, Stefan Hallerstede, Thai Son Hoang, Farhad Mehta, and Laurent Voisin. *Rodin: an open toolset for modelling and reasoning in Event-B.* International Journal on Software Tools for Technology Transfer, 12(6):447–466, Nov 2010.

[2] Jean-Raymond Abrial. *Modeling in Event-B: System and Software Engineering.* Cambridge University Press, New York, NY, USA, 1st edition, 2010.

[3] Thai Son Hoang, Laurent Voisin, Asieh Salehi, Michael J. Butler, Toby Wilkinson, and Nicolas Beauger. *Theory plug-in for Rodin 3.x.* CoRR, abs/1701.08625, 2017.