

FoREnSiC – An Automatic Debugging Environment for C Programs

Roderick Bloem¹ Rolf Drechsler² Görschwin Fey² Alexander Finder²
Georg Hofferek¹ Robert Könighofer¹ Jaan Raik³ Urmas Repinski³ André Sülflow²
forensic@lists.iaik.tugraz.at

¹ Graz University of Technology

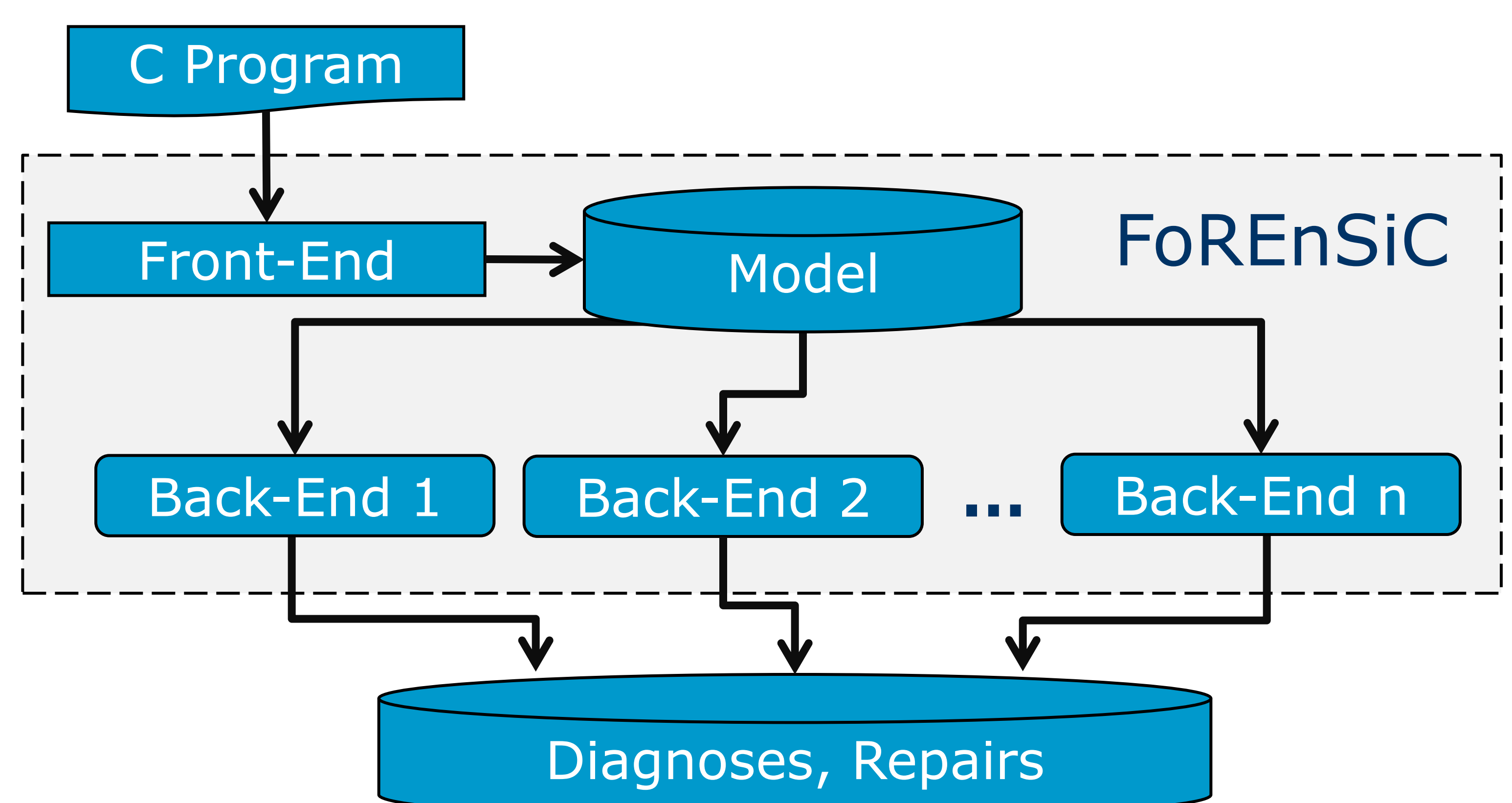
²University of Bremen

³Tallinn University of Technology

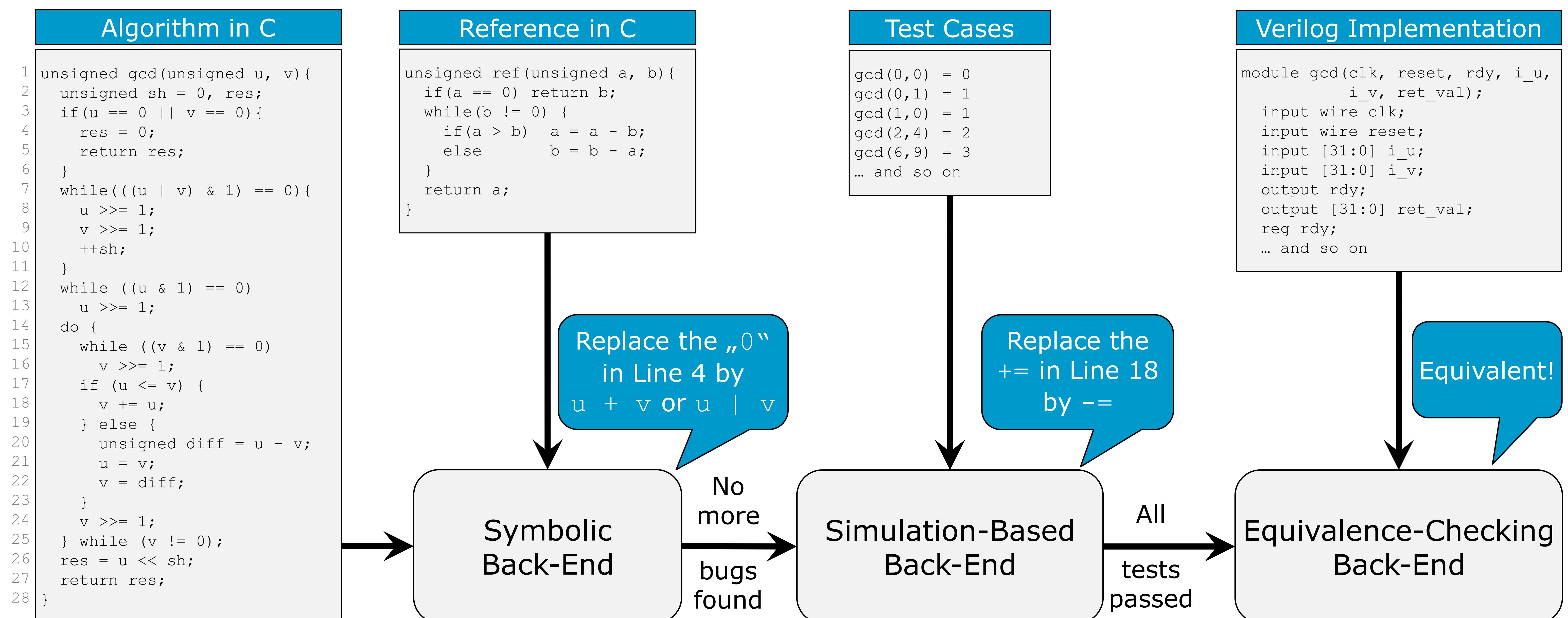
Abstract

FoREnSiC is short for **F**ormal **R**epair **E**nvironment for **S**imple **C** and represents an extensible environment for automatic error detection, localization, and correction in C programs. It implements different debugging methods in a unified way. Currently, a scalable simulation-based back-end, a back-end using symbolic execution, and a formal back-end to verify equivalence between a C program and a hardware design are available. FoREnSiC is designed as an extensible framework. Its infrastructure includes a powerful front-end and interfaces to logic problem solvers and can be reused for implementing new program analysis and debugging methods.

Architecture



Example



Symbolic Back-End

- Symbolic execution and SMT-solving
- Model-based diagnosis
- Template-based repair

Simulation-Based Back-End

- Simulation-based verification and error localization
- Mutation-based repair

Equivalence-Checking Back-End

- Simulation-based cutpoint detection
- SAT-based verification
- Cutpoint-based diagnosis



Bibliography

- <http://www.informatik.uni-bremen.de/agra/eng/forensic.php>
- A. Sülflow, U. Kühne, G. Fey, D. Große, and R. Drechsler. WolFram – a word level framework for formal verification. In *International Workshop on Rapid System Prototyping*, pages 11–17, 2009.
- R. Könighofer and R. Bloem. Automated error localization and correction for imperative programs. In *International Conference on Formal Methods in Computer Aided Design*, pages 91 – 100, 2011.