Abstract Debugging with GobPie

Karoliine Holter[1], Juhan Oskar Hennoste[1], Simmo Saan[1],
Patrick Lam[2], Vesal Vojdani[1]

[1]University of Tartu, [2]University of Waterloo

DEBT, September 2024

Demo

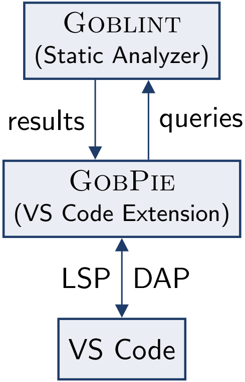
Introduction

Goblint

A program analysis framework based on abstract interpretation, with a focus on multithreaded C-Code.

GobPie

A Visual Studio Code extension designed to provide an intuitive user interface for Goblint.



GobPie

GobPie provides

- Goblint warnings in code editor,
- an abstract debugging feature,

and thus facilitates

- interactive use of Goblint,
- using a formal verification tool in a setting already familiar to developers.

Abstract debugger

- An abstraction of concrete debugger.
- Uses static analysis results — an over-approximation of *all* program behaviors.
- Results are recorded in an Abstract Reachability Graph (ARG) of a program.
- ARG models the control flow relation between the abstract representations of the possible concrete states.
- Navigation of the ARG mimics the step-based program execution in a traditional debugger, with stepping operations functioning similarly.

Theorem (Soundness [1])

Let $c_0 \Rightarrow c_1 \Rightarrow \cdots \Rightarrow c_n$ be a debugging session in the concrete world. Then, there exists a corresponding debugging session $a_0 \rightsquigarrow a_1 \rightsquigarrow \cdots \rightsquigarrow a_n$ in the abstract world such that $c_i \in \gamma(a_i)$ for $0 \leq i \leq n$.

Demo

Features' overview

Covered in demo:

- Stepping operations
- Displaying Variable Values
- Evaluating expressions
- Breakpoints
- At Breakpoints: Displaying multiple program states at once

Additional features:

- Context-sensitivity
- Call Stack displays the longest common suffix of the paths leading to the node
- Stepping with multiple program states at once
- Conditional Breakpoints

Conclusion

The abstract debugger feature within GobPie

- Uses static analysis results with stepping operations similar to conventional debugger,
- Helps in understanding the analyzer warnings
  (e.g., why the analyzer believes that a data race might happen).
- Enables interactively posing questions about interprocedural program behavior and to receive answers valid for *all* program executions, using static analysis information.

Further reading

- Karoliine Holter, Juhan Oskar Hennoste, Patrick Lam, Simmo Saan, Vesal Vojdani
  Abstract Debuggers: Exploring Program Behaviors Using Static Analysis Results
  To appear at Onward! 2024

- https://goblint.in.tum.de
- https://github.com/goblint/analyzer
- https://github.com/goblint/gobpie