Supporting Alternative SMT Solvers in Viper
Practical Work Description

Lasse F. Wolff Anthony
laanthony@student.ethz.ch
Supervised by Dr. Malte Schwerhoff
Department of Computer Science
ETH Zurich
Switzerland

1 Introduction

Viper [1], or Verification Infrastructure for Permission-based Reasoning, is a tool chain and infrastructure for program verification [1]. It consists of the Viper intermediate language, based on separation logic to encode verification problems, automatic verifiers for the language, and several example front-end tools for languages such as Python or Rust. Viper is developed by the Programming Methodology group at ETH Zurich, Switzerland.

Viper includes two automated verification backends: (1) Silicon [2], a verifier based on symbolic execution (SE), and (2) Carbon [3], a verifier based on verification-condition generation (VCG). Both of these ultimately rely on an SMT solver to discharge resulting proof obligations. Silicon interacts directly with an SMT solver, currently Z3 [4]. Carbon instead uses the intermediate verification language Boogie [5] to abstract over the underlying SMT solver.

Most SMT solvers support the SMT-LIB [6] language as input, but use different combinations of subsolvers, strategies, and heuristics to solve different tasks, which affects their completeness and efficiency. As such, supporting different SMT solvers in Viper may result in improved performance for certain classes of inputs.

2 Project Goal

The goal of this project is to enable backend support for multiple SMT solvers in Viper, with a focus on supporting cvc5 [7], and additionally Yices 2 [8] for the SE-based Silicon verifier. Reaching this goal requires (1) ensuring that the produced encodings are compatible with Z3, cvc5, and ideally Yices 2, and (2) generalizing the codebase to support the different solvers.

3 Tasks

The core tasks to achieve this goal are as follows:

- Identify Z3-specific extensions of SMT-LIB that the Silicon uses

[1] http://viper.ethz.ch
[2] https://github.com/viperproject/silicon
[3] https://github.com/viperproject/carbon
[4] https://github.com/Z3Prover/z3
[5] https://github.com/boogie-org/boogie
[6] http://smtlib.cs.uiowa.edu/
[7] https://cvc5.github.io/
[8] https://yices.csl.sri.com/

Practical work, fall semester 2021.

- Modify Silicon to only produce encodings that follow the SMT-LIB standard (or a variation that Z3, cvc5, and ideally Yices 2 support)
- Compare the runtime of original vs. changed Silicon, to see if any of the Z3-specific SMT-LIB extensions (if any) are crucial for performance
- Introduce support for cvc5 and Yices 2 to the codebase
- Benchmark performance of Viper when using the Z3, cvc5, and Yices 2

The project could be extended to achieve similar goals for the VCG-based verifier, Carbon. This would involve similar core tasks, but the changes would need to be done on Boogie rather than Carbon, since Boogie communicates with the SMT solver.

References

[1] Peter Müller, Malte Schwerhoff, and Alexander J Summers. Viper: A verification infrastructure for permission-based reasoning. In *International conference on verification, model checking, and abstract interpretation*, pages 41–62. Springer, 2016.