Deductive Verification for Weak Memory Programs

Research in Computer Science Project Description

Gaurav Parthasarathy
Supervisor: Alexander J. Summers

April 1, 2017

1 Introduction

In concurrent programs different threads running in parallel can access shared memory locations. The semantics of such programs with respect to these memory accesses are defined by a memory model. Since giving strong guarantees on the behaviour of memory accesses in a concurrent setting is expensive on modern hardware architectures, certain programming languages define relatively weak memory models. This allows compilers to generate efficient code for the underlying hardware.

Reasoning about weak memory programs is challenging and therefore there is a need to develop program logics and verification tools which can help prove such programs correct. Relaxed Separation Logic (RSL) [7], Fenced Separation Logic (FSL) [1], FSL++ [2], and GPS [6] are program logics which have recently been developed for the C11 relaxed memory model which is the memory model defined in the current C++ standard. In the following, we will refer to RSL, FSL, FSL++ collectively as the *RSL logics* (FSL extends RSL and FSL++ extends FSL).

The proofs for weak memory programs using these logics were originally constructed in the interactive theorem prover Coq. While this gives very strong guarantees on the correctness of the proofs, it leads to a lot of manual effort. To alleviate this problem Alexander J. Summers and Peter Müller have shown in [5] how to encode certain features of the RSL logics into Viper [4]. This automates the verification for weak memory programs using these logics. In the bachelor's thesis of Christiane Goltz [3] a frontend tool

which translates an annotated C++ program into the Viper encoding has been developed.

2 Core Tasks

The main goal of this project is to evaluate the RSL logics with the goal of verifying real-world weak memory programs. The hope is that this evaluation will give a deeper insight into the verification of practical weak memory programs using the RSL logics as well as how to extend the logics to be able to verify a larger class of such programs. To assist the evaluation the Viper tool may be used to automate the proofs.

This goal can be divided into the following core tasks:

- Apply the RSL logics to real-world weak memory programs. Identify the strengths as well as the limitations of the logics with respect to these programs.
- Extend the RSL logics to overcome some of their limitations.
- Verify real-world weak memory programs using the RSL logics along with the introduced program logic extensions.

3 Further Work

Possible extensions for this project are

- Evaluate the difference in annotation overhead when verifying a program using the Viper tool compared to constructing the proof using Coq.
- Potentially identify limitations present in the Viper language/encoding and suggest ways to overcome these limitations.
- Define and implement the encoding of the introduced program logic extensions for Viper.
- Prove soundness of the suggested program logic extensions.

References

[1] Marko Doko and Viktor Vafeiadis. A program logic for C11 memory fences. In *Proceedings of the 17th International Conference on Verification, Model Checking, and Abstract Interpretation - Volume 9583*, VMCAI 2016, pages 413–430, New York, NY, USA, 2016. Springer-Verlag New York, Inc.

[2] Marko Doko and Viktor Vafeiadis. Tackling real-life relaxed concurrency with FSL++. In *European Symposium on Programming*, ESOP, 2017. To appear. Draft available from http://plv.mpi-sws.org/fsl/.

[3] Christiane Goltz. A prototype verifier for weak memory reasoning. Bachelor's thesis, ETH Zurich, 2017.

[4] Peter Müller, Malte Schwerhoff, and Alexander J. Summers. Viper: A verification infrastructure for permission-based reasoning. In *Proceedings of the 17th International Conference on Verification, Model Checking, and Abstract Interpretation - Volume 9583*, VMCAI 2016, pages 41–62, New York, NY, USA, 2016. Springer-Verlag New York, Inc.

[5] Alexander J. Summers and Peter Müller. Automating deductive verification for weak-memory programs. Technical Report arXiv:1703.06368, 2017.

[6] Aaron Turon, Viktor Vafeiadis, and Derek Dreyer. GPS: Navigating weak memory with ghosts, protocols, and separation. In *Proceedings of the 2014 ACM International Conference on Object Oriented Programming Systems Languages & Applications*, OOPSLA '14, pages 691–707, New York, NY, USA, 2014. ACM.

[7] Viktor Vafeiadis and Chinmay Narayan. Relaxed separation logic: A program logic for C11 concurrency. In *Proceedings of the 2013 ACM SIGPLAN International Conference on Object Oriented Programming Systems Languages & Applications*, OOPSLA '13, pages 867–884, New York, NY, USA, 2013. ACM.