CONSTRUCTOR UNIVERSITY BREMEN

Bachelor of Science
Computer Science

Andrei Kozyrev

Equality saturation for solving equalities of relational expressions

Bachelor's Thesis

Scientific supervisor:
professor Anton Podkopaev

Reviewer:

Bremen
2023

Abstract

Modern CPUs are being developed exceptionally fast, and the number of cores is increasing rapidly. This has led to the development of multithreading, which is a technique that allows for the execution of multiple threads on a single CPU. Memory models are a fundamental aspect of multithreading and describe how memory is ordered at runtime in relation to source code. Currently, existing memory models are unsatisfactory and there is a need for new models that can be rigorously proven. In order to achieve this, formal verification using the Coq proof assistant is utilized, which enables automated proof checking and ensures the accuracy of results. Specialists in weak memory are continuosly improving the results in this domain.

One of the big and common problems in weak memory is the proof of equivalence of several memory models. Memory models are represented as expressions over relational language.

This thesis focuses on the automation of proving equalities over relational expressions in Coq. We are utilizing the techniques of equality saturation and E-graph data structure to generate proof of equaivalence for a given pair of terms. By automating these proofs, we can greatly increase the efficiency and accuracy of the proof process in weak memory.

Acknowledgements

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Contents

1 Introduction

Weak memory specializes on research that aims to improve the results in memory modeling. Weak memory focuses on determining the relative strength of different models. This is why one of the common challenges is to show equaivalences between several models. Given two memory models `A` and `B` we might want to check if model `A` is stronger then model `B`. If that condition holds, then the consistency of the model `A` would imply the consistency of the model `B`.

Memory models and propositions about them are usually represented as expressions over relational language. However, proofs of such the propositions are typically massive and very error prone. There were several cases of incorrect result in submited and published papers. Batty et al. [1] suggested an incorrect fix for the semantics of SC calls in C++, which was later documented by Lahav et al. [3]. Moreover Pichon-Pharabod et al. in 2016 suggested an incorrect proof of compilation in their paper [4].

Weak memory proofs are usually written in Coq [2], which is a proof assistant system. Coq helps to grant the correctness of the proof process. Coq provides a language called Calculus of Inductive Constructions (CIC), which is used to write the proofs. In CIC, the proofs are expressed as formal mathematical objects, and the correctness of the proof is ensured by checking that the proof is consistent with the axioms and rules of logic. Coq is the standard of development in the field of weak memory, helping to unify and verify already complex proofs.

Nevertheless, weak memory proofs, even with a use of Coq, tend to be huge and convoluted. This is why in this thesis, we are focusing on automating a specific part of weak memory proofs, namely the proofs of equaivalences between several memory models. As already mentioned, memory models are defined as relations. Let's denote `r` and `r'` the two memory models or relations over a given set. Now we may consider common operations in relational language, e.g. transitive closure (`r*`), reflexive closure (`r?`), or composition (`r ;; r'`).

References

[1] Mark Batty, Alastair F. Donaldson, and John Wickerson. "Overhauling SC atomics in C11 and OpenCL". In: (2016). URL: https://doi.org/10.1145%2F2837614.2837637.

[2] Yves Bertot and Pierre Castéran. *Interactive theorem proving and program development: Coq'Art: the calculus of inductive constructions*. Springer Science & Business Media, 2013.

[3] Ori Lahav et al. "Repairing Sequential Consistency in C/C++11". In: *SIGPLAN Not.* 52.6 (2017), pp. 618–632. ISSN: 0362-1340. URL: https://doi.org/10.1145/3140587.3062352.

[4] Jean Pichon-Pharabod and Peter Sewell. "A Concurrency Semantics for Relaxed Atomics That Permits Optimisation and Avoids Thin-Air Executions". In: *SIGPLAN Not.* 51.1 (2016), pp. 622–633. ISSN: 0362-1340. DOI: 10.1145/2914770.2837616. URL: https://doi.org/10.1145/2914770.2837616.