INF-2990 Bachelor's Thesis in Informatics

Verification of correctness of a concurrent hash map with TLA+

Åsmund Aqissiaq Arild Kløvstad - Advisor: Håvard D. Johansen

"The prevalence of programming errors has led to an interest in proving the correctness of programs"

I. Introduction

1) concurrent systems are important
2) the power wall has made concurrency important
3) highly parallelizable tasks (machine learning)
4) concurrent systems are difficult to check
5) why is this hashmap in particular of interest?
   - just kind of nifty tbh
   - probably useful for something, look into papers that cite Shalev

A. Thesis

Shalev et al.'s concurrent hashmap can be implemented as a TLA+ specifcation and shown to be correct.
The purpose of this specification is to increase confidence in model checkers by showing correspondence with the formal proof of Shalev et al.
Additionally, the model will allow us to check other invariants and properties of the protocol like type correctness, liveness and the absence of deadlocks.

B. Method

1) implement the protocol in TLA+
   a) decreasing levels of abstraction
2) check for the invariants proven by Shalev et.al using TLC model checker

C. Outline

maybe an outline here

II. Background

A. Concurrent programs

B. TLA+

1) temporal logic of actions
2) the Specification language
   - syntax?
3) the TLC model checker
   - state machines
   - transition functions
   - invariants

C. Shalev et.al's hashmap

1) hashmaps in general
2) resizing
3) "lock free" (maybe in the concurrency background)
4) buckets
5) split-ordering
6) dumy nodoes

III. The Specification

Low on details for now

1) assumptions and abstraction
2) step size
3) transition function
4) specifying invariants

IV. Results / evaluation

1) what did we prove?
2) is that the same as Shalev et.al claims?
3) was TLA+ useful?

V. Conclusion

1) what was done?
2) what did we prove / find out?
3) further work? maybe its own section