Advisor	Amin Timany
Students	Mathias Pedersen
Languages	English
Text tools	LaTeX

Project Description

Safety of computer programs is a highly desirable property, especially for critical pieces of software. With processors getting more threads and concurrent programs and algorithms becoming more prevalent, reasoning about the safety of programs must also evolve to support these features. One strategy for reasoning about the safety of programs is to define a logic, which one can then use to prove correctness of a given program.

A common synchronisation strategy is using *locks* which allows for only a single thread to access protected data at a time. A plethora of different lock mechanisms have been put forward, each with their own positives and negatives. However, lock-free algorithms have also been proposed, which rely on primitive operations to achieve synchronisation.

This thesis will study concurrent data-structures that aim to facilitate synchronisation in concurrent programs. In particular, it will study the lock-free algorithm: The Michael-Scott Queue.

Further, the thesis will use *Iris* to reason about the safety of the Michael-Scott Queue. Iris is a Concurrent Separation Logic Framework, meaning that it can be used to prove safety of concurrent programs that use a heap. Iris has been implemented in the Coq proof assistant, and this thesis also aims to formalise all the work within this implementation.

Provisional Activity Plan

1. Study and understand the Michael-Scott Queue.
2. Learn the Iris implementation in Coq.
3. Implement the Michael-Scott Queue in the language $\lambda_{\text{ref,conc}}$.
4. Formalise and prove a specification for the Michael-Scott Queue.
5. Write a report

Supervision Plan

Time has been allocated for meetings, with possible exceptions for the month of March.