Advisor	Amin Timany
Students	Mathias Pedersen
Languages	English
Text tools	LaTeX

Project Description

Correctness of computer programs is a highly desirable property, especially for safety critical pieces of software. With processors getting more threads and concurrent programs and algorithms becoming more prevalent, reasoning about correctness of programs must also evolve to support these features.

A strategy for reasoning about the correctness of programs is to define a logic, which one can then use to guarantee certain behaviour of programs. One such logic is *Iris* which is a concurrent separation logic framework, meaning that it can be used to prove safety of concurrent programs that use a heap.

Concurrent programs usually need a way of synchronising. A common technique for this 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 advantages and disadvantages. However, lock-free algorithms have also been proposed, which rely on primitive, atomic 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 has also been implemented in the Coq proof assistant, and this thesis also aims to use this implementation to mechanise all results.

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.