

CS4223 Assignment 2

Julius Putra Tanu Setiaji (A0149787E), Kyle Timothy Ng Chu (A...)

15 November 2019

1 Choice of Programming Language

Initially, we tried to implement the trace simulator using Python. However, after a while, we realised that it gets unwieldy, unreadable, and difficult to extend given the complexity of implementing a trace simulator. Particularly, while Python has enumerations, they are not as powerful as Algebraic Data Types (ADT) in other languages, especially with regards to associated values. Python's lack of pattern matching also makes implementing a finite state machine (FSM) a hassle. While Python's dynamic typing allows more rapid prototyping, a better compile-time typecheck would be really useful to prevent bugs.

Haskell seems to fit the bill really well here. However, since we are essentially simulating a mutable, eagerly-evaluating von Neumann architecture machine, Haskell's characteristic of being immutable and lazy can become an impediment. We need a language that has some of Haskell's features such as being strongly and statically typed, pattern matching and ADT, while still allowing us to have mutability. For these reasons as well as familiarity, we chose to implement our tracing simulator using **Scala**.

2 Implementation

Our code can be divided into 3 different parts: specific code for common code among all the protocols, the MESI Protocol trace simulator, and specific code for the Dragon Protocol trace simulator.

2.1 Common Code

2.2 MESI Protocol

2.3 Dragon Protocol

3 Quantitative Analysis

4 Advanced Task: Optimisation to Basic MESI Protocol