

Cs358 Exam

Name: _____ ID: _____

This assignment has 4 questions, for a total of **75** marks.

Question 1: **Formalising capability machines** 25 marks

Capability machines extend assembly instructions with explicit capabilities such that reading and writing on memory is only allowed if a capability is provided.

Take ULC (with Nats, pairs and products) and add a heap from natural numbers (i.e., as the one from assignment 6), make allocation deterministic starting from 0, each new location is at the next number. Extend the language with capabilities and formalise their semantics. You choose how to model them, choose wisely according to their behaviour as described below.

Capabilities are unforgeable and unobservable tokens which the program can create. Every time a memory location is created, it is unprotected. The language must provide primitives for protecting a location given a capability, this should only be possible if the location is unprotected. Reading and writing a memory location is always possible if the location is unprotected. However, if the location is protected with a capability, reading and writing that location is only possible if the same capability is provided at reading and writing time.

Question 2: **Thread-based concurrency**.....20 marks

Take STLC with sums and pairs. Add a new term for spawning a process, call that *fork t*. *fork t* must not reduce *t*, instead it must step to 0 and immediately spawn a thread whose body is *t*. Threads are scheduled nondeterministically for now, though you may want to look at the next exercise to build a more robust solution.

Formalise the statics and dynamic semantics of such a concurrent language and show all changes to the formalisation of the language.

Question 3: **Round-robin scheduling**.....15 marks

Take the language from the previous exercise. Change the scheduling to be round-robin. A thread executes for 10 steps and then control passes to the next thread (if there is one). If a thread terminates before 10 steps, control passes to the next thread. The next thread to be scheduled must be the next one in order of spawning.

Formalise the scheduling process and show changes to the formalisation of the language.

Make the formalisation elegant.

Question 4: **Free theorems**.....15 marks
Consider type $\tau = \forall\alpha.\forall\beta.\alpha \rightarrow \beta$. Is it inhabited? If yes, show a term that inhabits τ and prove that such a term is in the term relation for τ . If not, prove that there is no term in the term relation for τ .