Nathan Villagaray-Carski

6.945 Final Project Proposal

NOTE: I plan on doing this as a solo project if possible, so keep that in mind when evaluating the scope of the project.

Overview:

I would like to implement a generic Turing Machine simulator in scheme. The basic idea is to allow the user of the program to specify the state machine backing the TM and the starting tape for the TM and then run it and return the result (if any). The user would be able to choose between a standard TM, a non-deterministic TM, a read-only TM, a probabilistic TM, circular-tape TM, and possibly others such as a multi-tape or multi-track TM. It will also be possible for the user to mix and match combinations of these machines, run something such as a probabilistic read-only multi-tape TM, reusing components from their respective machines. Most of the challenge would come from creating a generic enough base that it would be possible to write small components to alter the behavior of the simulator and allow for all these different types of machines.

Components:

- Base TM framework – reads in the state machine and starting tape, begins running the machine using the specified modules (see below), and stops the machine when the stop symbol is encountered
- Modules (rough estimate of necessary modules)
  - Read Tape – Handles reading from the tape. This would have to be modified for multi-tape and multi-track TMs
  - Write Tape – Handles writing to the tape. This would have to be modified for multi-tape and multi-track TMs, and read-only TMs
  - State Transition – Handles transitions in the state machine backing the TM. This would have to be modified for the NTM and probabilistic TM
  - Tape Movement – Handles moving the “head” on the tape. This would have to be modified for the circular-tape TM

Plan:

I should be able to get the Base TM framework and the modules for the standard TM up and running pretty quickly, and work out any oversights with the base framework while creating the other modules.

Tentative Milestones:

Week 1/2 – Base framework and standard TM working

Week 3/4 – All other modules (non-deterministic TM, read-only TM, probabilistic TM, etc.) working

Week 5/6 – Finish up documentation