Turing Machine Equivalent Computational Models

Pushdown Automata with 2 Stacks

- A 2-PDA (PDA with 2 stacks) can be shown to be equivalent in computational power to a Turing machine
  - => Show that we can simulate a single-tape, non-deterministic Turing machine using a 2-PDA
  - <= Show that we can simulate a 2-PDA using a Turing machine

Deque Automaton

- A deque automaton (finite automaton with a double-ended queue) can be shown equivalent in computational power to a Turing machine
  - => Show that we can simulate a 2-PDA using a deque automaton
    - By transitivity, this shows we can simulate a Turing machine using a deque automaton
  - <= Show that we can simulate a deque automaton using a Turing machine

Queue Automaton

- A queue automaton can be shown to be equivalent in computational power to a Turing machine
  - => Show that we can simulate a deque automaton using a queue automaton
    - By transitivity, this shows we can simulate a Turing machine using a queue automaton
  - <= Show that we can simulate a queue automaton using … a deque automaton.
    - This one is easy. We just don't exploit the extra capability of the deque.
    - By transitivity, this shows we can simulate a queue automaton using a Turing machine

Observations

- A deque automaton is no more computationally powerful than a queue automaton
- A queue automaton *IS* more computationally powerful than a PDA (with stack)
- A “0-PDA” (no stack)
  - NFA
- A “1-PDA” (one stack)
  - PDA
- A “2-PDA” (two stacks)
  - TM
- A “3-PDA” (three stacks)
  - TM (without proof, extra stacks don’t add computational power at this point)

Other Computational Models

- Many other models of general purpose computation have been proposed
  - Some similar to Turing Machines
  - Some very different
  - Share characteristic of unrestricted access to unlimited memory
  - All such models are equivalent in power

Quantum Computing

- Turing machines can simulate quantum computers.
  - Quantum computers can't solve any problems that can't already be solved using a Turing machine

- Quantum computing *may* allow certain problems to be solved more efficiently than with classical computing
  - There is a misconception that quantum computing can solve problems exponentially faster than classical computing and thereby solve NP problems in polynomial time
    - This is not known to be true, and is generally suspected to be false