

# CS 3530: Assignment 6e

Fall 2014

## Problems

### Problem 3.14 (20 points)

#### Problem

A *queue automaton* is like a push-down automaton except that the stack is replaced by a queue. A *queue* is a tape allowing symbols to be written only on the left-hand end and read only at the right-hand end. Each write operation (we'll call it a *push*) adds a symbol to the left-hand end of the queue and each read operation (we'll call it a *pull*) reads and removes a symbol at the right-hand end. As with a PDA, the input is placed on a separate read-only input tape, and the head on the input tape can move only from left to right. The input tape contains a cell with a blank symbol following the input, so that the end of the input can be detected. A queue automaton accepts its input by entering a special accept state at any time. Show that a language can be recognized by a deterministic queue automaton iff the language is Turing-recognizable.

#### Solution

2 parts:

1. Show how TM can generate a DQA that recognizes the TM language
2. Show how a DQA can generate a TM that recognizes the DQA language

TMs and DQAs can do equivalent actions:

DQA push: add a symbol to the left end

DQA pull: read and remove symbol from the right end

TM push: add a symbol to the right end of the tape

TM pull: read a symbol from the left end and mark it

It seems that both TMs and DQAs are very much the same. The strings can be processed the same ways as long as there is a marker in the queue that shows the front of the string. To access a particular symbol in the queue for TMs you would just scan it with the head and mark it. To do the same for a DQA you can pull symbols off and push them until you get the right one then you pull it without pushing. Instead of moving the head to the start in the TM. You would pull and push the symbols until you get to the initial marker.

example:

the language is in the form of  $abaabbbaabbb\dots$  for any length.

The TM would mark an a then mark a b that followed its group of a's.

The DQA would pull an a without pushing it, pull and push any a's that followed the initial a and pull its first b (To make sure groups of a's and b's to get mixed up you can use additional marker symbols)