Reading assignment: Read section 4.2 of Sipser’s book.

Problems:

1. Sipser’s text, 2nd edition Exercise 4.6 (1st edition Exercise 4.7). (Note that infinite binary sequences are not strings since any string has finite length.)


4. Define a queue automaton $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ where $Q$ is the finite set of states, $\Sigma$ is the input alphabet, $\Gamma$ is the queue alphabet, $q_0$ is the start state, $q_{accept}$ and $q_{reject}$ are accept and reject states respectively, and

$$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \rightarrow Q \times \Gamma^*$$

where

- $\delta(q_{reject}, a, B) = (q_{reject}, B)$ for all $a \in \Sigma \cup \{\varepsilon\}$ and $B \in \Gamma$, and
- for all states $q$ either $\delta(q, \varepsilon, B) = (q_{reject}, B)$ for all $B \in \Gamma$ or $\delta(q, a, B) = (q_{reject}, B)$ for all $a \in \Sigma$ and $B \in \Gamma$.

Thus, ignoring moves that immediately lead to rejection, the states can be divided into those on which input symbols are read and those that ignore the input.

A configuration of a queue automaton is an element of $Q \times \Sigma^* \times \Gamma^*$; configuration $(q, y, z)$ represents that the current state is $q$, the remaining input is $y$, the current contents of the queue is $z$ (with the left-most character on the left end of $z$). If $\delta(p, a, A) = (q, B)$ where $A \in \Gamma$, $B \in \Gamma^*$ then its action on configurations is to take $(p, ay, Az)$ to $(q, y, zB)$. The start configuration on input $x \in \Sigma^*$ is $(q, x, \$).

By the condition on the transition function $\delta$, in any configuration there is just one next configuration that does not immediately lead to $q_{reject}$. That is, a queue automaton is like a DPDA except that it has a queue instead of a stack.

Sketch how queue automata are equivalent to Turing machines.
(HINT: to simulate one step of the TM might require going through the entire queue of the queue automaton.)