582206 Models of Computation (Autumn 2011)
Homework 11 (29 November–2 December)

1. The two Turing machines for this problem are given on the next page. Although it is not important
for solving the problem, you may wish to know that both the Turing machines recognize the language
\( \{ \, w w \mid w \in \{ 0, 1 \}^* \, \} \).

(a) Show the computation (that is, the sequence of configurations, as at bottom of page 146 in the
textbook) for the deterministic Turing machine given on next page on input 001001.

(b) Show one accepting and one rejecting computation for the nondeterministic Turing machine given
on next page on input 001001.

Hint: the nondeterminism is in states \( q_1 \), which has two transitions for symbol 0, and \( q_6 \), which
has two transitions for symbol 1. The idea for an accepting computation is to loop in state \( q_1 \) or \( q_6 \),
depending on the first symbol in the input, for the first half of the input, and then progress to state
\( q_2 \) just after the midpoint of the input string is passed.

2. Give a state diagram for a Turing machine that recognizes the language \( \{ a^i b^j c^i d^j \mid i, j \in \mathbb{N} \} \).

3. Give a state diagram for a two-tape Turing machine that recognizes the language \( \{ a^n b^n c^n \mid n \in \mathbb{N} \} \).

One suitable way of representing the transition \( \delta(r, a_1, a_2) = (s, b_1, b_2, D_1, D_2) \) is

```
\begin{array}{c}
a_1 \rightarrow b_1, D_1 \\
a_2 \rightarrow b_2, D_2 \end{array}
```

4. Give a state diagram for a nondeterministic Turing machine that recognizes the language
\( \{ \# w_1 \# w_2 \# \ldots \# w_n \# \mid w_i \in \{ 0, 1 \}^* \text{ for all } i \text{ and } w_i = w_j \text{ for some } i \neq j \} \)
over the alphabet \( \{ 0, 1, \# \} \).

5. [Sipser Problem 3.9] Let a \( k \)-PDA be a pushdown automaton that has \( k \) stacks. Thus a 0-PDA is an NFA,
and a 1-PDA is a conventional PDA.

(a) Show that 2-PDAs are more powerful than 1-PDAs.

(b) Show that 3-PDAs are not more powerful than 2-PDAs.

Hint: Simulate the tape of a Turing machine using two stacks. Give the basic idea of the simulation using
high-level pseudocode without getting too deep into details.

6. [Sipser Exercise 3.14] A queue automaton is like a push-down automaton, except that a queue replaces
the stack. Two kinds of operations can be performed on the queue:

- \textsc{Enqueue}(a) inserts the symbol \( a \) to the end of the queue
- \textsc{Dequeue} reads and deletes the first symbol of the queue.

As with a PDA, the input can be read one symbol at a time. We assume that the end of the input is marked
with the symbol \( \_ \) that may not appear anywhere else in the input. The queue automaton accepts by
entering a special accept state (like a Turing machine).

Show that any Turing-recognizable language can be recognized by a queue automaton. A sufficient
solution to this problem is to explain, using a suitable level of pseudocode, how a Turing machine can be
simulated using a queue.

Turing machines for Problem 1 on the next page!
Deterministic Turing machine for Problem 1(a).

Nondeterministic Turing machine for Problem 1(b).