York University CSE2001 February 16, 2011

Homework Assignment #5
Due: March 2, 2011 at 2:30 p.m.

1. Give a regular expression that describes the set of all odd-length binary strings that contain an even number of 0's.

2. The York University Turing Machine Description Format (YUTMDF) is a way to write down a description of a Turing machine in a text file. The Turing machines described in YUTMDF use the following conventions, as described in the lectures.

- They use a 1-way infinite tape.
- The tape alphabet has two different special symbols, $\triangleright$ and $\sqcup$ that are not part of the input alphabet.
- Initially, if the input string is w, the tape contains $\triangleright w$ at the left end of the tape, and the rest of the tape contains only $\sqcup$ symbols. The head of the Turing machine is initially positioned at the first character of the input string w (i.e., at the tape's second square).
- Whenever the Turing machine sees the $\triangleright$ symbol, it must leave it unchanged and move right (but it can change state).

We also make some naming conventions. We assume that the state set of the Turing machine is $Q = \{q_0, q_1, \ldots, q_{n-1}\}$ where $n \geq 3$ and the tape alphabet of the Turing machine is $\Gamma = \{c_0, c_1, \ldots, c_{m-1}\}$ where $m \geq 3$. We also assume that q_0 is the initial state, q_{n-2} is the accepting state and q_{n-1} is the rejecting state. We assume that the input alphabet is $\Sigma = \{c_0, c_1, \ldots, c_k\}$ where $0 \leq k \leq m-3$ and $c_{m-2} = \sqcup$ and $c_{m-1} = \triangleright$.

We now explain how to describe, using YUTMDF, a Turing machine that follows the conventions described above. Since YUTMDF is a text file, we use an underscore character _ to represent $\sqcup$ and > to represent $\triangleright$. The first line of the file contains the three integers n, m, and k, separated by single spaces. (Recall that these are the sizes of the state set, tape alphabet and input alphabet, respectively.) The second line of the file contains a string $c_0 c_1 \ldots c_{m-3}$ of length $m-2$ which gives each character of the tape alphabet in order (except $c_{m-2} =$ _ and $c_{m-1} =$ >). All characters in this string should be distinct (and different from > and _). The third line contains a non-negative integer T.

Following this, there are T lines. Each of these remaining lines of the description contains five items i, a, i', a', d separated by single spaces, where i and i' are integers with $0 \leq i \leq n-3$ and $0 \leq i' \leq n-1$ (inclusive), a and a' are characters in the tape alphabet and d is a single character that is either L or R. This line indicates that $\delta(q_i, a) = (q_{i'}, a', d)$. No two lines should have the same i and a. Note that no transitions are given for situations when the machine is in state q_{n-2} or q_{n-1} since those are the accepting and rejecting states. If no transition is given to describe $\delta(q_i, a)$ for a non-halting state q_i, then it is assumed that $\delta(q_i, a) = (q_i, a, R)$.

Now for the question that you have to solve. Design a Turing machine that decides the language $L = \{w \in \{0,1\}^* : w \text{ contains at most twice as many 0's as 1's}\}$. For example, 00100110 is in L because it contains 5 0's and 3 a's and $5 \leq 2 \cdot 3$. You should write your Turing machine description in a text file using YUTMDF and submit it using the submit command. Detailed instructions on how to do this will be posted on the course web page.

I will post a Java programme on the course web page which takes as input a Turing machine (in YUTMDF) and a string and runs the Turing machine on the string. You can use that to test your machine. I will also post an example of a Turing machine specified in YUTMDF.

Bonus Question: Busy Beavers
The bonus question below is separate from the rest of Assignment 5.
The bonus question is due Monday, March 7 at 11:59 p.m.

3. In this question, we shall consider the basic Turing machines as defined on the opposite side of this page. We shall restrict attention to machines that use the tape alphabet $\Gamma = \{\triangleright, \sqcup, 0, 1\}$. The goal is to design a Turing machine which, on input ε, takes as many steps as possible before halting. Your machine must have exactly seven states, including the initial state, the accepting state and the rejecting state.

This bonus question is worth up to 4 percent to be added to your final grade. Students who submit a correct Turing machine will get between 1 and 4 bonus points. If two students submit machines that take the same number of steps, each student will get 1 point. Excluding these students, the higher the number of steps your machine takes, the more bonus points you will earn. To earn all 4 points, you will have to make your machine take a very large number of steps.

If your machine takes more than 20,000,000 steps when given the empty string as input, I may not be able to test it by running it, so you will have to provide, in writing, a written proof that it does indeed halt (and that it takes lots of steps) in order to receive your bonus points.

To submit your solution, write your Turing machine in YUTMDF text file and use the submit command. (Details of how to do this will be posted on the course web page.) If your submission is syntactically incorrect or does not follow the specifications of this question (for example if it has too many states) or runs forever on the empty input string, you will receive 0 points.

For this bonus question, you may consult outside sources. A good place to start is the Computer Recreations column by A. K. Dewdney in the August, 1984 issue of *Scientific American* (pp. 19–23). (Note that the Turing machines discussed there use two-way infinite tapes, so you cannot use them directly to answer this question, but you might get some ideas about how to design your own busy beaver.)