Homework Assignment #1
Due: September 19, 2012

1. Let $L = \{0^x 1^x 2^x : x \in \mathbb{N}\}$.

(a) Give a high-level description of a (single-tape) Turing machine that decides L.

(b) Implement your machine and submit it using YUTMFF (see below).

(c) Give a function $T(n)$ such that your machine takes at most $T(n)$ steps on *all* inputs of size n.

For full marks, there should be no Turing machine that solves the problem in worst-case time $o(T(n))$, but you do not have to prove this. (I.e., your machine should be as efficient as possible, ignoring constant factors.)

York University Turing Machine File Format (YUTMFF)

You should use the submit command to submit your solution to part (b) as a text file in York University Turing Machine File Format (YUTMFF), which is described below. 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 $\Sigma = \{c_0, c_1, \ldots, c_{k-1}\}$ where $0 \leq k \leq m-2$ 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. 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.)

Each character in the tape alphabet has a name. The second line of the file contains $m-2$ strings separated by single spaces that give the names of the characters $c_0, c_1, \ldots, c_{m-3}$. We use the name `blank` to represent $c_{m-2} = \sqcup$ and `leftend` to represent $c_{m-1} = \triangleright$.

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 $0 \le i \le n-3$ and $0 \le i' \le n-1$ (inclusive), a and a' are names of 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 $\delta(q_i, a) = (q_i, a, R)$.

Some Java code will be posted on the course web page for reading and simulating a Turing machine in YUTMFF, so that you can test your solution to part (b).