# Homework Assignment #1
## Due: September 28, 2020 at 5:00 p.m.

1. Let $L = \{x_1 \# x_2 \# \cdots \# x_k : k \geq 2$ and $x_i \in \{0,1\}^*$ and $\exists i > 1$ such that $x_1 = x_i\}$. You must design a Turing machine that decides this language. Given any string in $\{0, 1, \#\}^*$, your machine should accept if the string is in $L$ and reject if the string is not in $L$.

   (a) Give a high-level description of your Turing machine. See Examples 3.11 and 3.12 in the textbook for the kind of description that you should provide.

   (b) Give a complete description of your Turing machine in YUTMFF as a text file (see below).
   There is a Java programme posted on the course website that allows you to test your solution before you submit it.

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


2. A no-writing Turing machine is one that never changes any character written on its tape. (Thus, in the transition function of such a machine, if $\delta(q, a) = (q', a', d)$ then $a = a'$.) Let $L$ be a language. Assume there is a no-writing Turing machine $M$ that decides $L$.

   (a) Prove that there exists a constant $k$ such that, for every input string $x$, $M$ never visits the same square of the tape more than $k$ times. Hint: Think about the sequence of states $M$ is in when it visits that square.

   (b) Show that you can construct a no-writing Turing machine $M'$ that decides $L$ without ever moving beyond the first $n + 2$ squares of the tape on any input of length $n$.

   (c) Prove that $L \in TIME(n)$.

## 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 YUTMFF use the following conventions, as described in Chapter 3 of the textbook.

- They use a single 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 YUTMFF, 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 $0 \leq i \leq n - 3$ and $0 \leq i' \leq n - 1$ (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.

## Submission Instructions

Submit your solution to 1(b) as a text file using the submit command. Submit your solutions to all other parts using the crowdmark link on the eclass site for the course.