CMPSC 464: Introduction to the Theory of Computation **(Date: 09/14/2022)**

Recitation #3 Solution

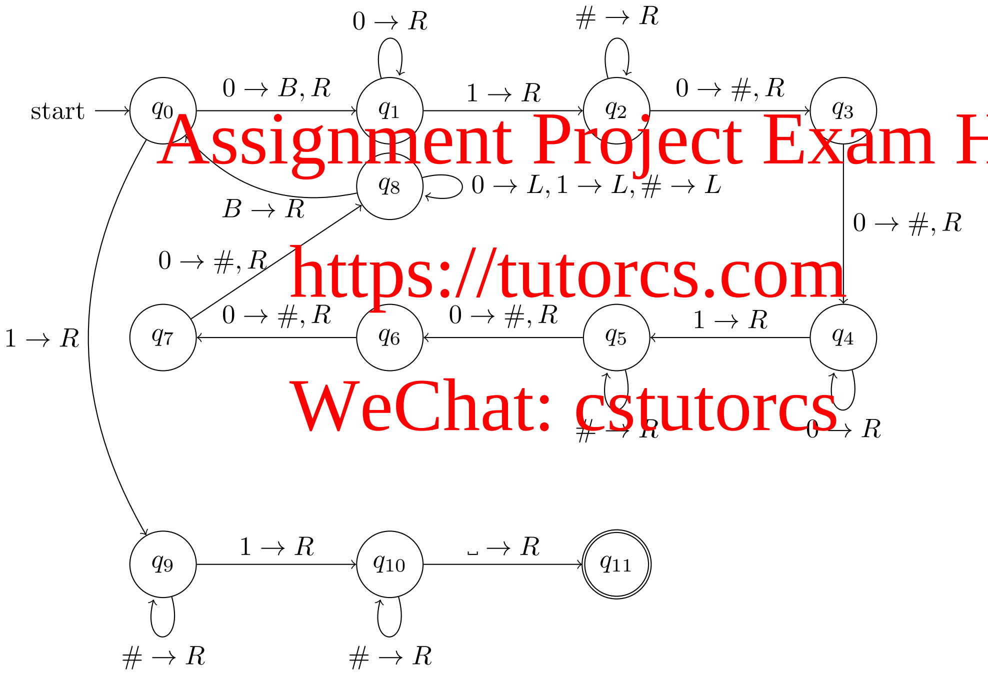
Instructor: Dr. Young Kun Ko *TAs:* Hamed Mahdavi, Levent Toksoz, Neha Sanjay Rathod, Yuzhang Wang

Problem 1

Let $\Sigma = \{0, 1\}$. Give a state diagram of a Turing Machine that accepts the following language.

$$L = \{0^n 1 0^{2n} 1 0^{3n} | n \geq 0\}$$

Solution:
We give the following turing machine $M = \{Q, \Sigma, \Gamma, q_0, F, \delta\}$, where $\Sigma = \{0, 1\}$ and $\Gamma = \{0, 1, \#, B\}$
The transitions $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ are as follows:



Given a. input w, M checks if $w \in L$. It works as follows:

1. Overwrite one 0 (in the first sequence of 0s) by one B and move right.
2. Move right skipping over 0s until the first 1 is encountered.
3. Move right skipping over #s until 0 is encountered.
4. Overwrite two 0s by two #s and move right.
5. Move right skipping 0s until second 1 is encountered.
6. Move right skipping over #s until 0 is encountered.
7. Overwrite three 0s by three #s and move right.
8. Move left skipping over all symbols until B is encountered.

We repeat the above steps until M halts. Finally we can check if $w \in L$ by checking if all 0s on the tape are overwritten by Bs or $\#s$.

Problem 2

2-dimensional Turing machine has the usual finite-state control, but a tape that is a *2-dimensional* grid of cells, infinite in all directions. The input is placed on one row of the grid, with the head at the left end of the input and the control in the start state. Acceptance is by entering the final state. Prove that the languages accepted by a *2-dimensional* Turing machine are the same as those accepted by an ordinary Turing machine.

Solution:
We will show that a language accepted by a 2D TM M_2 can be accepted by a multi-tape Turing machine M_1. Thus it will follow that it can be accepted by a standard single-tape TM.

- Let us number the cells of M_2 by pairs of integers (x, y) where the initial position of $M_2's$ head is numbered (0, 0). This splits the grid into four quadrants.
- M_1 stores the pair of integers on the first "position" tape.
- M_1 will have four one-way infinite "grid" tapes, each representing a quadrant of the 2D grid (e.g. the first tape represents the quadrant where $x \geq 0, y \geq 0$).
- To simulate an access to an element of the grid, M_1 uses the signs of x, y to determine the tape on which the symbol is stored.
- The absolute values of x, y are used to determine the position of the element on the grid tape, using the standard diagonal numbering function:[1]

$$f(a, b) = \frac{(a+b)(a+b+1)}{2} + b$$

- The value of $f(|x|, |y|)$ is calculated on an additional "scratch" tape. The simulation of a single transition of M_2 is then done as follows:
  1. Use the signs of x and y to determine the grid tape which should be used.
  2. Compute $a = f(|x|, |y|)$ using the scratch tape.
  3. Move the appropriate grid tape head to the a^{th} cell. (Note that grid tapes are one-way infinite, and we can use a special marker symbol to mark the leftmost cell.)
  4. Based on the current state of M_2 and the symbol read, determine the next state, overwrite the current cell on the grid tape, and simulate the move of the head of M_2 by increasing or decreasing x or y on the position tape.

[1] There are infinitely many such functions. We just picked one of them.