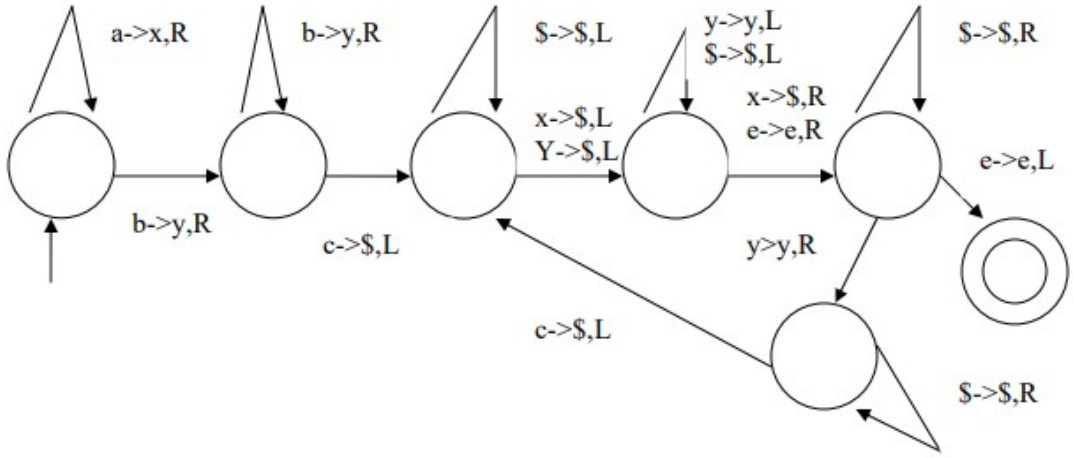
Assignment 4 SOLUTION- CS 301 - Theory of Automata – Fall 2018

Due: December 5, 2018 at the start of class. Late submission until December 7 (12 PM) is allowed with 25% penalty.

You MUST submit the following questions: 1, 2, 5, 7.

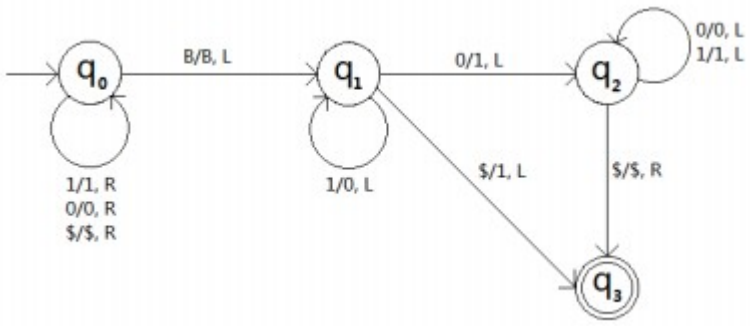
1. [15] Design a deterministic one-tape Turing machine, with input alphabet {a, b, c}, that accepts the language
   $L = \{ a^i b^j c^k \mid i>=1, j>=1, \text{and } k = \max\{i, j\} \}$.
   Your solution should
   - Describe the overall algorithm,
   - Include a transition diagram of the machine, and
   - Show the sequence of IDs (states and tape contents) on the input abbcc.

Answer: NOTE: Only Transition Diagram GIVEN in the solution. 'e' has been used to show space. In one transition, Y is written accidentally, it's the same as y.



2. [15]

Design a Turing machine that takes as input a number N and adds 1 to it in binary. To be precise, the tape initially contains a $ followed by N in binary. The tape head is initially scanning the $ in state q0. Your TM should halt with N + 1, in binary, on its tape, scanning the leftmost symbol of N + 1, in state qf .



3. [10] Describe a Turing machine that accepts exactly those strings of 0's, 1's, and 2's that have the same number of each character (so 010201221 would be accepted but 01122 would be rejected).

SOLUTION

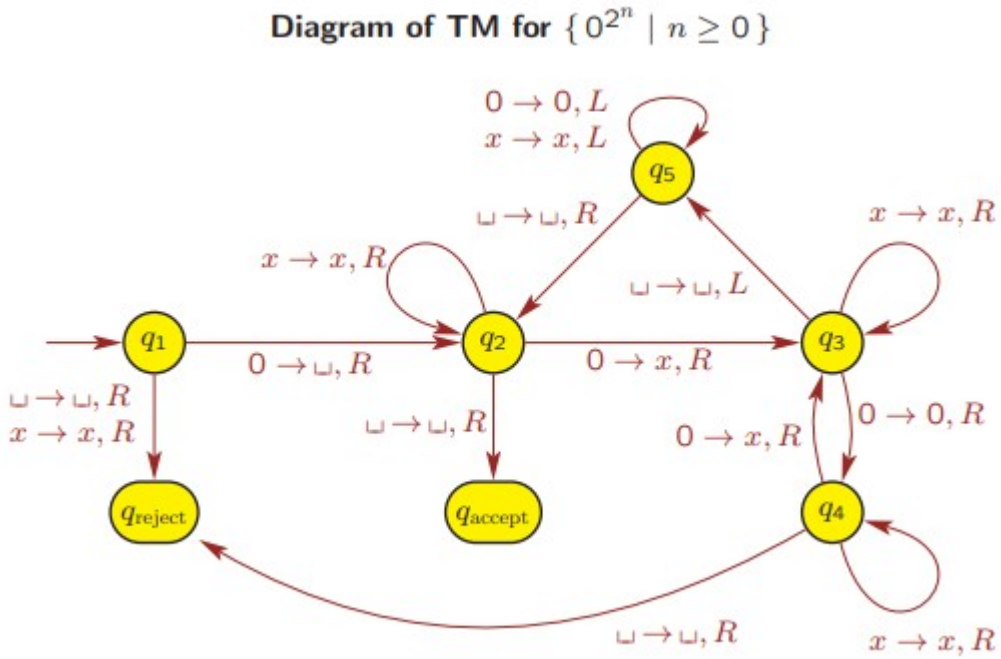
The machine will work by finding a 0, a 1, and a 2 and replacing them with X's. It will repeat the process until all characters in the input have been replaced (accept) or if it can't find a 0, 1, or 2 (reject).

1. Move the read/write head to the leftmost non-blank space.
2. Check whether there is a 0, 1, or 2 before the next blank to the right. If not, then accept (all characters were paired off).
3. Move the head to the leftmost non-blank space and then move right to find the first 0 and replace it with an X. Reject if no 0 is found before the next blank to the right.
4. Move the head to the leftmost non-blank space and then move right to find the first 1 and replace it with an X. Reject if no 1 is found before the next blank to the right.
5. Move the head to the leftmost non-blank space and then move right to find the first 2 and replace it with an X. Reject if no 2 is found before the next blank to the right.
6. Go back to the first step.

NOTE: Only description needed, TM diagram not required.

4. [20] Describe and construct Turing machine that recognizes the following languages:
   a. $A = \{ 0^{2^n} \mid n \geq 0 \}$, which consists of strings of 0s whose length is a power of 2.
   b. B = { $a^i b^j c^k$ | i × j = k and i, j, k ≥ 1 }.

SOLUTION 4a



SOLUTION 4b

Example: Turing machine M_3 to decide language

$$C = \{ a^i b^j c^k \mid i \times j = k \text{ and } i, j, k \geq 1 \}.$$

Idea: If i collections of j things each, then $i \times j$ things total.

M_3 = "On input string w:

1. Scan the input from left to right to make sure that it is a member of $L(a^*b^*c^*)$, and *reject* if it isn't.
2. Return the head to the left-hand end of the tape
3. Cross off an a and scan to the right until a b occurs. Shuttle between the b's and the c's, crossing off each until all b's are gone. If all c's have been crossed off and some b's remain, *reject*.
4. Restore the crossed off b's and repeat stage 3 if there is another a to cross off. If all a's are crossed off, check whether all c's also are crossed off. If yes, *accept*; otherwise, *reject*."

5. [15] Suppose we are given three strings containing {a,b,c}* as input, we have to determine the index of the string which has maximum length.
   Example: if String1=abc String2=cccc and String3=cbabbb are the input, the output= 3
   a. Write an algorithm, in plain English, to solve this problem using a TM.
   b. Create a single tape deterministic Turing machine for your algorithm. Initially assume that all three input strings are separated by the delimiter #. Also, the input starts with # symbol. So when given the input on tape as:
   #abc#cccc#cbabbb#, the tape when TM halts should be: #abc#cccc#cbabbb#3

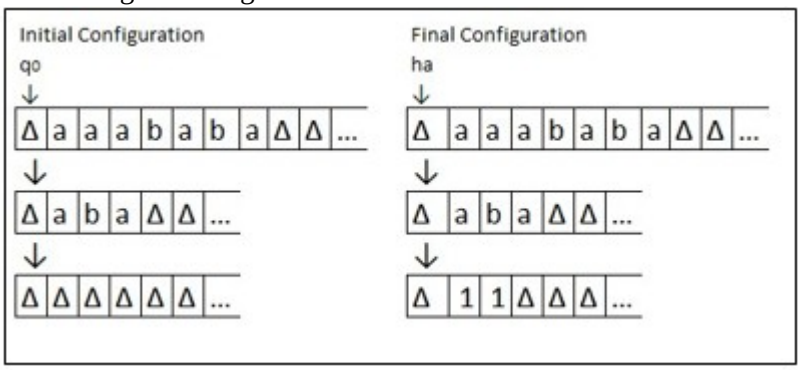
SOLUTION

Idea: Compare String 1 and 2, whichever string is larger in this comparison compare it with 3, which ever string is larger in this comparison write its index on end of tape. If in any comparison both strings are of equal length, one will be selected randomly.
Algorithm:
Step 1: Look for 1st uncrossed symbol of String 1,
If found cross it go to step 2
If not found, go to step 3 // string 2 is larger or equal than string 1
Step 2: Look for 1st uncrossed symbol in String 2,
If found cross it and go to step1
If not found, go to step 4 // string 1 is larger or equal than string 2
Step 3: // string 2 is larger or equal than string 1.
Uncross all the symbols of String 2//restore string 2 compare with String 3
Step 3b: Look for 1st uncrossed symbol in String 2,
If found cross it and go to step3c
If not found; go to end of tape and write 3, halt. // string 3 is greater of equal to string 2 and 1
Step 3c: Look for 1st uncrossed symbol in String 3,
If found cross it and go to step3b
If not found go to end of tape and write 2, halt. // string 2 is greater of equal to string 3 and 1
Step 4: // string 1 is larger or equal than string 2.
Uncross all the symbols of String 1 //restore string 1 compare with String 3
Step 4b: Look for 1st uncrossed symbol in String 1,
If found cross it and go to step4c
If not found; go to end of tape and write 3, halt. // string 3 is greater of equal to string 1 and 2
Step 4c: Look for 1st uncrossed symbol in String 3,
If found cross it and go to step4b
If not found go to end of tape and write 1, halt. // string 1 is greater of equal to string 3 and 2

6. [10]A common operation in Turing-machine programs involves “shifting over”. We need to move the contents of each of the cells to the right (or left) of the current head position one cell right (or left), and then find our way back to the current head position. Construct a Turing Machine for shifting right
   For example: if original tape: 0111001 → after shifting right tape should read $0111001

7. [20] Given 2 Strings X and Y such that X,Y ϵ $\{a,b\}^+$ Construct a Turing Machine that outputs the number of times Y appears in X.
   Your Turing machine will have 3 tapes. Tape 1 will have X as input, Tape 2 will have Y as input, and on Tape 3 you have to write the output. As your output is the number of times Y appears in X, it should be in unary form.

   Following is an example of initial and final configuration of such Turing machine if X= **aaababab** and Y=**abab.**is given in figure below



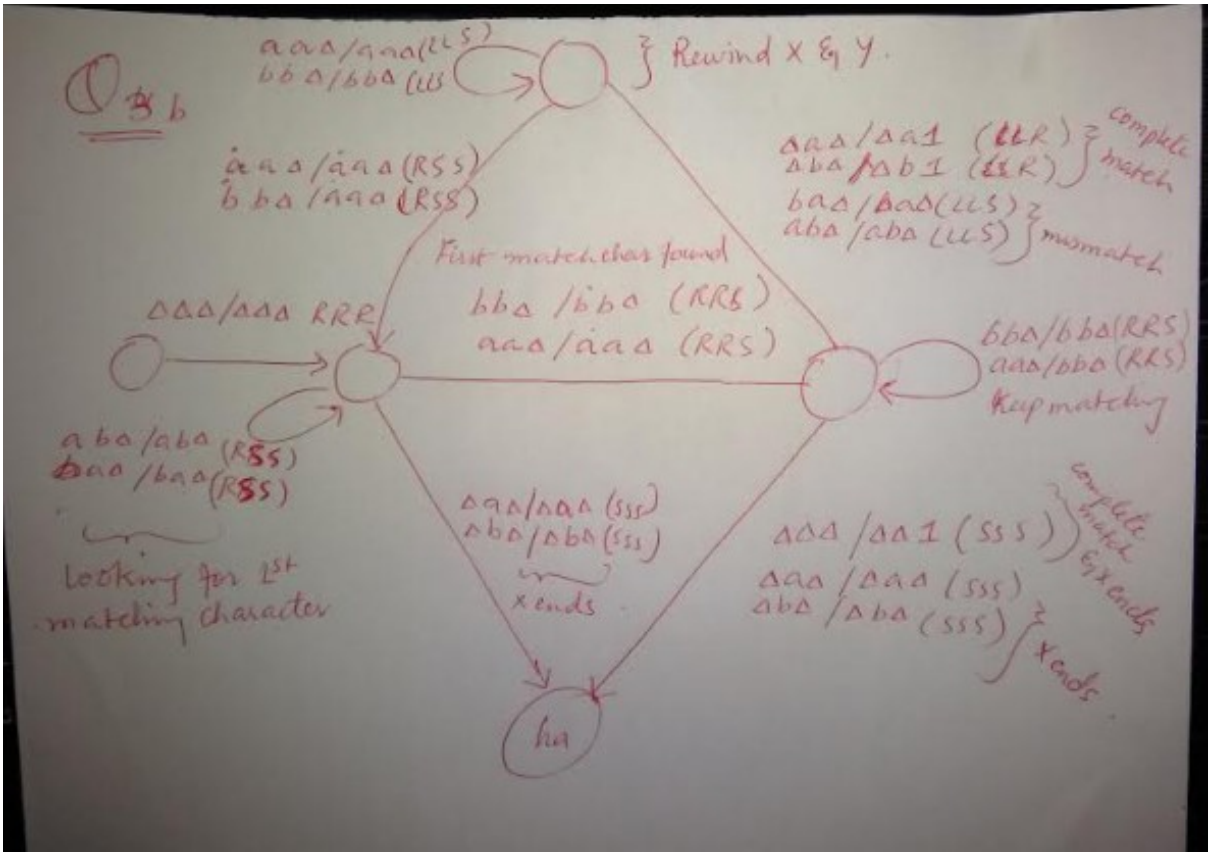
NOTE: Output is 11 are there are two occurrences of Y in X (aaababab)

You also have to write a brief algorithm of your machine

Algorithm:

0).If X has not ended
1)……Look for character on X that matches first character of Y, left to right starting from current head of ……tape
2)……If found, mark that character in X as xdot
3)……..Now Keep matching X and Y from left to right
4)……..If Y matches with X completely
5)……..Place 1 in tape 3
6)……..Rewind tape1 and tape2 heads by moving tape 1 to one right from last xdot and ……..tape 2 to first symbol of Y
……..Goto 0
7). Else goto accept state

NOTE: Rewinding tape 1 is necessary to find overlapping Y's in X. It is also necessary in case where Y and X partially match before complete match, for example if X=aaab and Y=aab. Most of the students made mistake in this part, so if you see this comment in you paper "Rewind X aswell, the explanation precedes.


Rewind x & y.
complete match
mismatch
First match char found
bba/bba (RRS)
aaa/aaa (RRS)
looking for 1st matching character
x ends
complete match & x ends
x ends
ha