	Course: Program: Duration: Paper Date: Section: Exam:	Theory of Automata BS(Computer Science) 180 min Feb 13, 2021 ALL Final	Course Code: Semester: Total Marks: Weight: Page(s): Roll No.	CS301 Fall 2020 70 45% 10
--	--	---	--	---------------------------------------

Instructions/Notes: Please show all the working/reasoning, no marks will be given without it. Please attempt questions on the question paper, in the given space.  
You can use rough sheets but those will not be collected.

## Question 1 (10 points): Decidability

Identify if the following decision problem is decidable or not. Justify your answer by first providing a high-level algorithm (or TM) for this problem and then specify for what type of inputs the algorithm will Accept/Reject/Loop Forever.

*Given two Deterministic Finite Automaton,  $M_1$  and  $M_2$ , identify if  $L(M_1) \cap L(M_2)$  is empty or not?*

Note:

$L(M)$  means the language of  $M$ , that is, a set of strings accepted by  $M$ .  
 $\cap$  is a symbol of set intersection.

Solution:

Algorithm: Input  $M_1$  and  $M_2$

Steps:

1. Create a DFA  $M$  of  $L(M_1)$  and  $L(M_2)$  using the closure properties of intersection of FAs.
2. Find a path from initial state to final states of  $M$ . (Path can be found using BFS).
3. If there is a path from initial state to final state/s Accept else Reject.

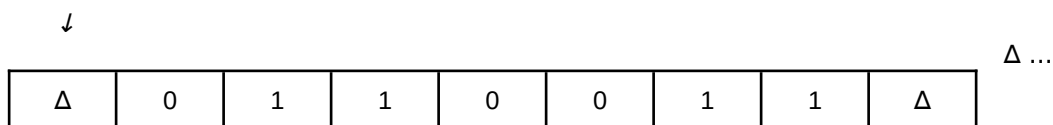
The problem is decidable as the algorithm will reject or accept for every input, and will not loop forever for any input.

Name: \_\_\_\_\_ Roll Number: \_\_\_\_\_

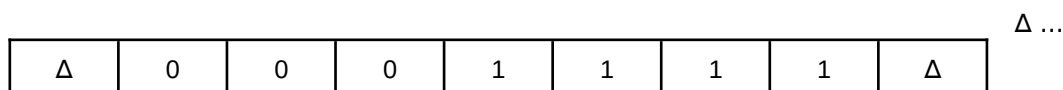
## Question 2 (15 points): Designing Turing machine

Assume  $\Sigma = \{0, 1\}$ . Design a Single Tape Deterministic Turing Machine that sorts the input tape such that all 0's (if any) must appear before all 1's (if any).

For example, if the input tape is initially.



Then once your Turing Machine halts, the tape should be



Where  $\Delta =$  Blank

Briefly describe steps of your algorithm in the space given below. Do not give a 1 or 2 line statement. Write a proper algorithm.

Then draw a state diagram of TM in space provided on the next page. Your TM design must have a maximum of 6 states, including ACCEPT ( $h_a$ ) state.

Assume that initially pointer head (tape head) is placed on the left-most tape square that contains a  $\Delta$ , followed by input string (as shown in the example initial input tape above). Tape head can be at

any position when the machine stops.

Solution:

Option 1:

Swap consecutive 10 in input iteratively until input is sorted.

Steps:

1. On state  $q_0$ , keep moving right if 0 found on the input tape.
  - a. if blank found , ACCEPT
2. Once a 1 is found on the input tape, move to state  $q_1$  and keep moving right on your input tape. a. If blank found, ACCEPT
3. If a 0 is found, convert it into 1 and move left.
4. Convert the 1 into 0 and move left.
5. Keep moving left until blank is encountered
6. Goto step 1

Department of Computer Science Page 2 of 10  
Name:\_\_\_\_\_ Roll Number:\_\_\_\_\_

Option 2: Swap left most 1 with left most 0 on RHS of that 1 (if any).

Steps:

Start:

Find the left most 1 and mark it as  $1^*$  (if not found goto ha)

Find the left most 0 on RHS of  $1^*$

    If found

        Change 0 to 1

        Go back and change  $1^*$  to 0

        Goto Start

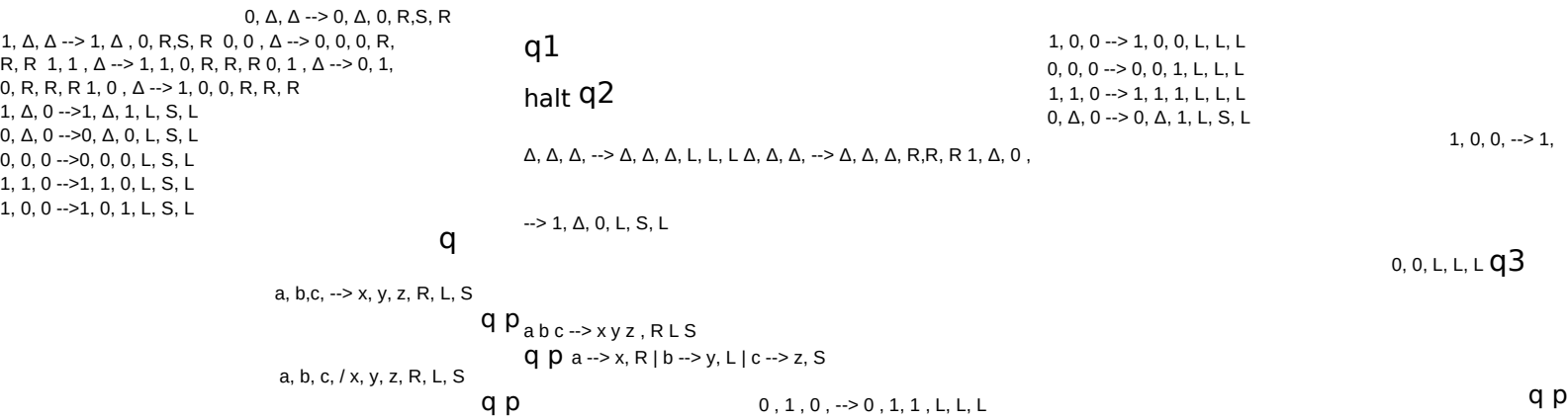
    else

        Change  $1^*$  to 1

        Goto ha.

Question 3 (10 points): Dry run Turing Machine

Consider a multi-tape (3 tape) Turing machine shown in Figure 1.

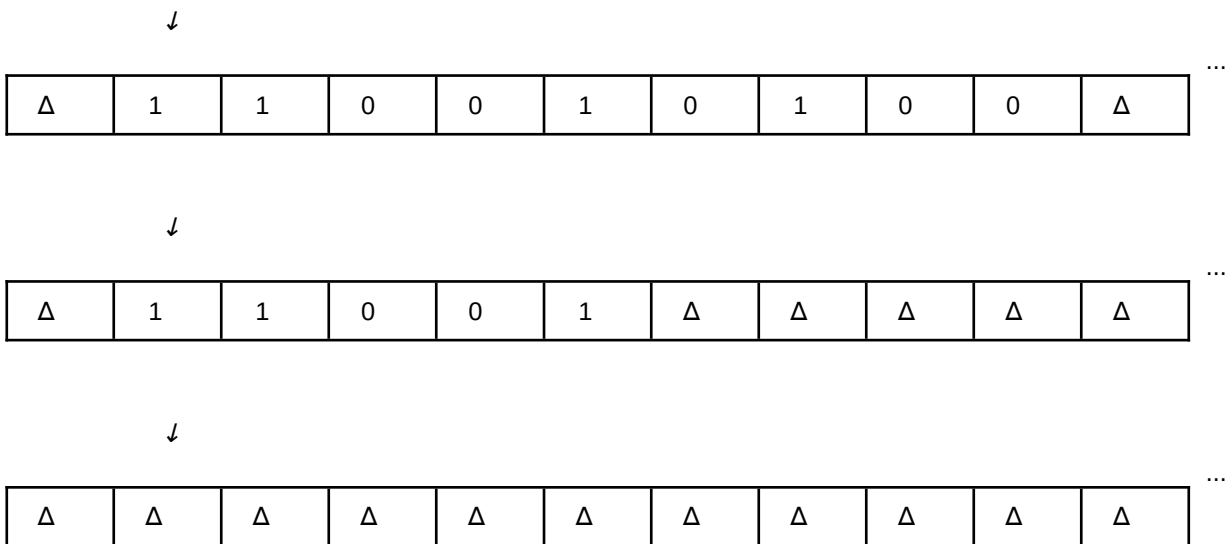


(a) Turing Maching for Question 3.

(b) Different notations for writing transition.<sup>1</sup>

Figure 1: Question 3, <sup>1</sup>Note that you might have used a different way to write transitions of TM in class. For your ease please see notations given in (b), they all mean the same. Where a,b,c are read and x,y,z are written on tape 1,2,3 resp. And R,L,S are tape 1,2,3 heads' movement resp.

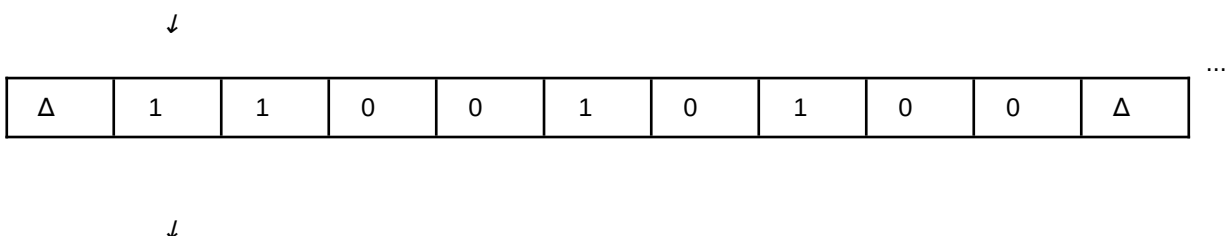
When machine starts, the contents and the tape heads are shown below.



Where Δ = =Blank

a). Run the Turing machine on it and show your working and the final configuration of tapes. Write your final answer in empty tapes provided below. Your answer should show the final position of tape heads and tapes contents. Show the working on space provided on next page (no marks without showing working).

SOLUTION:



...

Δ	1	1	0	0	1	Δ	Δ	Δ	Δ	Δ
---	---	---	---	---	---	---	---	---	---	---

↓

...

Δ	1	0	1	1	1	1	0	1	1	Δ
---	---	---	---	---	---	---	---	---	---	---

Department of Computer Science Page 4 of 10  
Name:\_\_\_\_\_ Roll Number:\_\_\_\_\_

b). Write down, in the box below, what calculation is this Turing Machine (Figure 1a) performing? Your answer should be of one line only.

SUBTRACTION

*Space to show working of running machine*

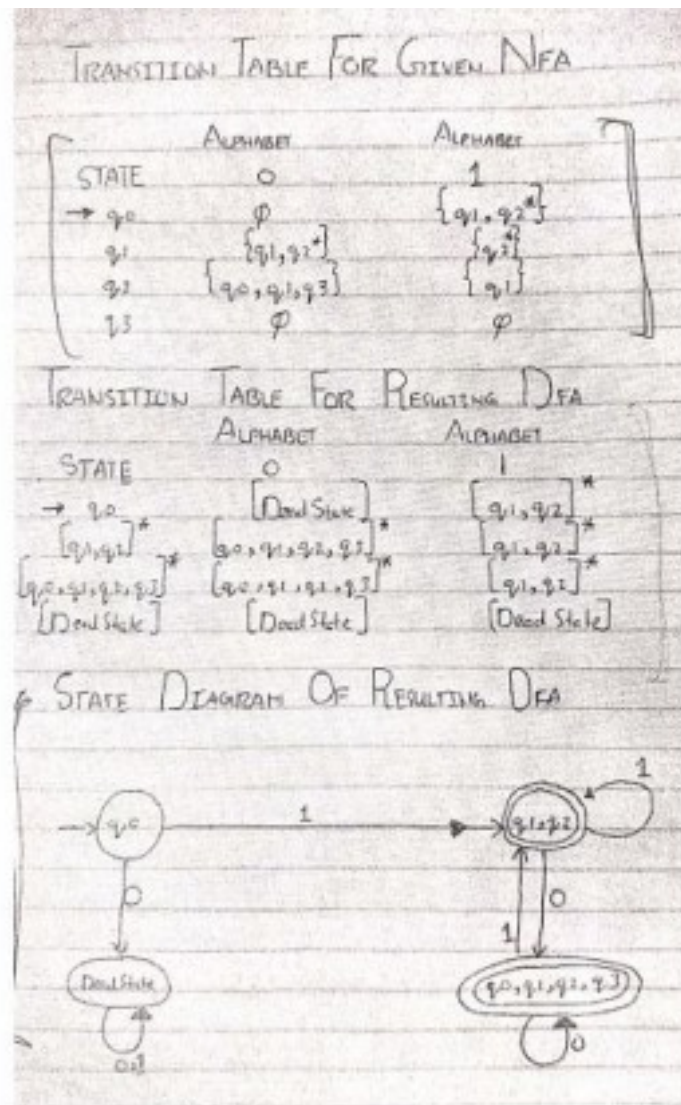
Question 4 (10 points): NFA to DFA

Convert the NFA given in Figure 2 to DFA. Draw the Transition table as well as the State diagram of the resulting DFA. The set of alphabet,  $\Sigma = \{0, 1\}$

Note: Your final DFA must not contain any unreachable states.



Figure 2: NFA for Question 4



Question 5 (5 points): CFG to PDA

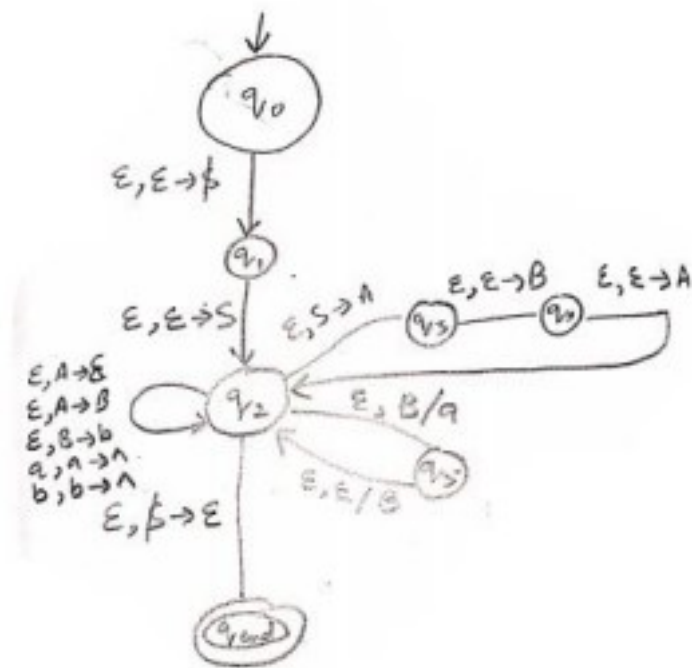
Using the proof of the theorem "Any language accepted by a CFG has an equivalent PDA", convert the following CFG to PDA.

$S \rightarrow ABA$

$A \rightarrow \epsilon \mid B$

$B \rightarrow Ba \mid b$

Note:  $\epsilon = \lambda = \text{null}$



Department of Computer Science Page 7 of 10  
 Name: \_\_\_\_\_ Roll Number: \_\_\_\_\_

#### Question 6 (10 points): Designing CFG

Construct a CFG for the language  $L$  consisting of words  $w_i$ . The alphabet set is  $\{a, b\}$ . Every word  $w_i = xyz$  i.e. it can be split into substrings  $x$ ,  $y$ , and  $z$  where

$x$  contains even number of  $a$ 's and any number of  $b$ 's.  $|x| \geq 0$ . Some examples of  $x$  are  $aa$ ,  $abaaa$ ,  $bbbaa$ , etc.  $y$  consists of odd number of  $a$ 's and any number of  $b$ 's.  $y$  can also be  $\epsilon$ . So,  $|y| \geq 0$ . Some examples of  $y$  are  $\epsilon$ ,  $a$ ,  $abaa$ ,  $bbba$ , etc.

$z$  is of the form  $a^m b^n$  where  $m \leq n \leq 2m$  and  $|z| \geq 0$ .

While designing the CFG, you must also fulfill the following constraints.

1. At max, you can only use 3 non-terminals (also know as variables).
2. The CFG must begin with the non-terminal  $S$ .
3. The CFG has 3 lines.

4. At max, you can use 4 '|' symbols in CFG.

Some valid words for this language are *aaaab*, *aaab*, *abaaabbaabb*, *abbbab*, etc.

*Note:*  $\varepsilon = \lambda = \text{null}$

Solution:

The idea is to consider  $x.y$  together as  $(a+b)^*$

$S \rightarrow MN$

$M \rightarrow aM \mid bM \mid \varepsilon$

$N \rightarrow aNb \mid aNb b \mid \varepsilon$

Department of Computer Science Page 8 of 10  
Name:\_\_\_\_\_ Roll Number:\_\_\_\_\_

Question 7 (10 points): Pumping Lemma for CFL

Using Pumping Lemma for CFL, show that the following language is not a context free

language.  $L = \{ a^i b^j c^k \mid k = i * j \text{ and } i, j, k \geq 0 \}$

Assume  $\Sigma = \{a, b, c\}$





3)  $vxy$  contain all cs  $\Rightarrow$  if  $i=2$   $uv^2xy^2z = a^p b^p c^{p^2+|v|+|y|}$   
 as  $|v| \geq 1$   $n(a) * n(b) \neq n(c)$

4)  $vxy$  contain some as and some bs.  ~~$i=2$~~

$\hookrightarrow$  if  $v$  contains on as &  $y$  contains on bs  
 $i=2$   $a^{p+|v|} b^{p+|y|} c^{p^2}$   $n(a) * n(b) \neq n(c)$

$\hookrightarrow$  if  $v$  or  $y$  contain string ~~ab~~ both ab.  
 pumping up will change the order and  
 new strings will not be in  $L$ .

5)  $vxy$  contain some bs and some cs

$\hookrightarrow$  if  $v$  contains only b's and  $y$  contains only cs.  
 $i=2$   $a^p b^{p+|v|} c^{p^2+|y|}$

$n(a) * n(b) \neq n(c)$  because  $|y| \geq p$

$\hookrightarrow$  if  $v$  or  $y$  contain some bs and some c.  
 then pumping up will change the order & new string

There fore 2 is not a prime

**SOCIETY for INDUSTRIAL and APPLIED MATHEMATICS**

3600 Market Street, 6th Floor - Philadelphia, PA USA 19104-2688 - +1-215-382-9800 - Fax +1-215-386-7999 - [meetings@slam.org](mailto:meetings@slam.org) - [www.slam.org](http://www.slam.org)