National University of Computer and Emerging Sciences, Lahore Campus

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

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 M1 and M2
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 form 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.

Question 2 (15 points): Designing Turing machine

Assume Σ = {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.

↓									
Δ	0	1	1	0	0	1	1	Δ	Δ ...

Then once your Turing Machine halts, the tape should be

Δ	0	0	0	1	1	1	1	Δ	Δ ...

Where Δ = ␣ = 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.

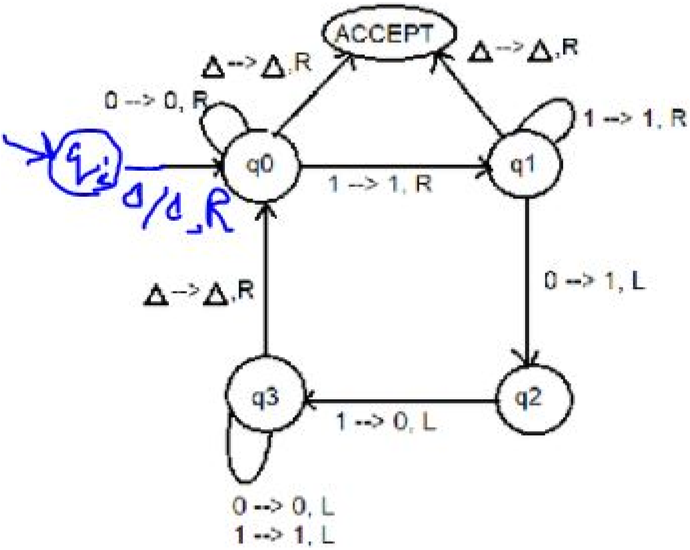
Then draw a state diagram of TM in space provided on the next page. Your TM design must have a maximum of 6 states, including ACCEPT (h_a) state.
Assume that initially pointer head (tape head) is placed on the left-most tape square that contains a Δ, followed by input string (as shown in the example 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.

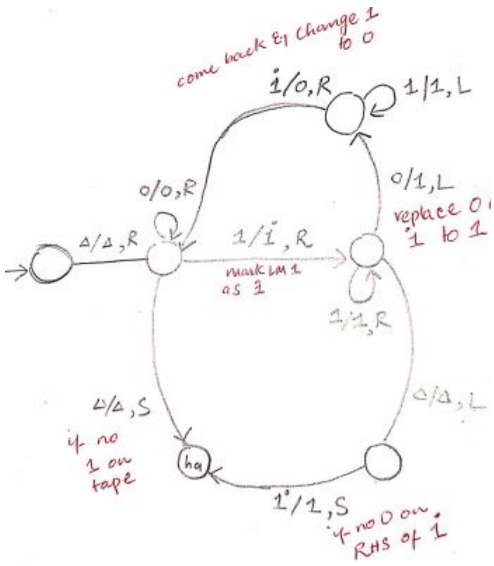
Steps:
1. On state q0, keep moving right if 0 found on the input tape.
   a. if blank found , ACCEPT
2. Once a 1 is found on the input tape, move to state q1 and keep moving right on your input tape.
   a. If blank found, ACCEPT
3. If a 0 is found, convert it into 1 and move left.
4. Convert the 1 into 0 and move left.
5. Keep moving left until blank is encountered
6. Goto step 1



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

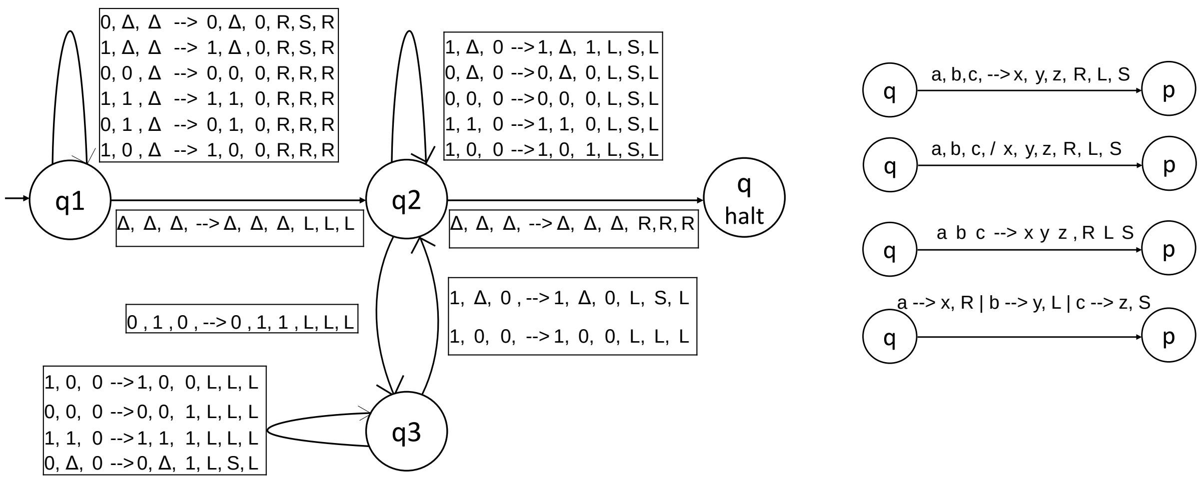
Steps:

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.[1]

Figure 1: Question 3, [1]*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.

	↓										
Δ	1	1	0	0	1	0	1	0	0	Δ	...

	↓										
Δ	1	1	0	0	1	Δ	Δ	Δ	Δ	Δ	...

	↓										
Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	...

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	0	1	0	0	Δ	...

	↓										
Δ	1	1	0	0	1	Δ	Δ	Δ	Δ	Δ	...

	↓										
Δ	1	0	1	1	1	1	0	1	1	Δ	...

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.

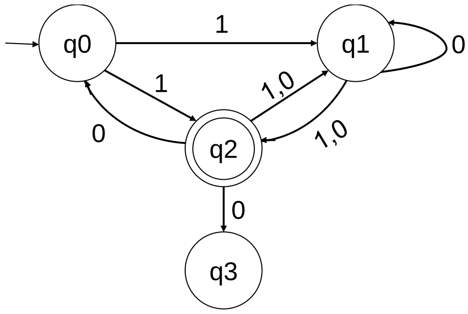


Figure 2: NFA for Question 4

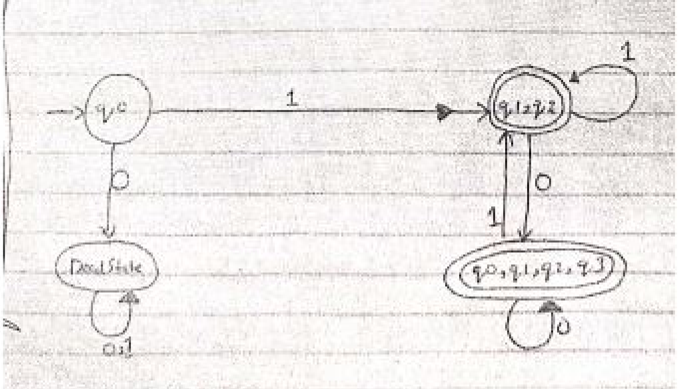
TRANSITION TABLE FOR GIVEN NFA

STATE	ALPHABET 0	ALPHABET 1
→ q0	φ	{q1, q2*}
q1	{q1, q2*}	{q2*}
q2	{q0, q1, q3}	{q1}
q3	φ	φ

TRANSITION TABLE FOR RESULTING DFA

STATE	ALPHABET 0	ALPHABET 1
→ q0	[Dead State]	[q1, q2]*
[q1, q2]*	[q0, q1, q2, q3]*	[q1, q2]*
[q0, q1, q2, q3]*	[q0, q1, q2, q3]*	[q1, q2]*
[Dead State]	[Dead State]	[Dead State]

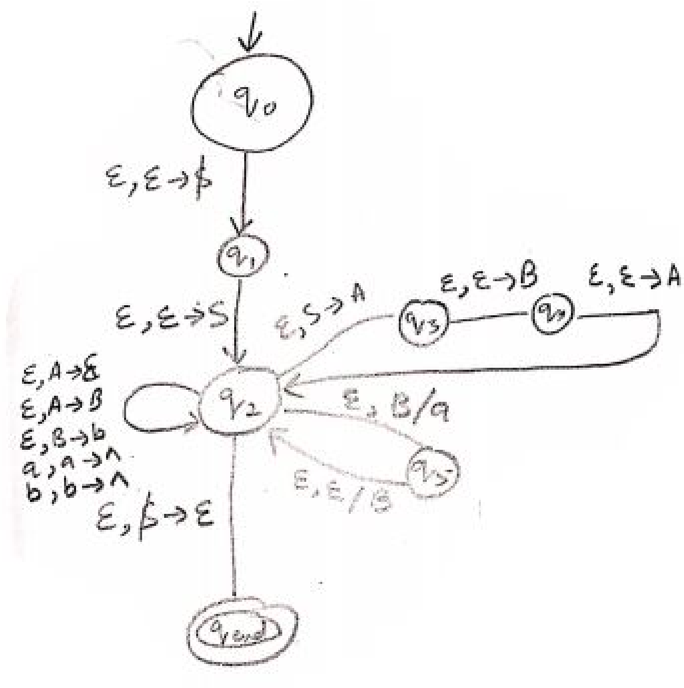
STATE DIAGRAM OF RESULTING DFA



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 $\rightarrow$ ε **| B**
B $\rightarrow$ **Ba | b**
Note: $\varepsilon = \lambda$ = null



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 ε. So, $|y| \geq 0$. Some examples of y are ε, *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: $\varepsilon = \lambda$ = null

Solution:

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

S $\rightarrow$ **MN**
M $\rightarrow$ **aM | bM |** ε
N $\rightarrow$ **aNB | aNbb |** ε

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= { $a^i b^j c^k$ | $k = i * j$ and $i, j, k \geq 0$}

Assume Σ={a, b, c}

$L = \{a^i b^j c^k \mid k = ij\}$

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

A.

Assume L is a CFL. Let p be the pumping length. Consider string $s = a^p b^p c^{p^2}$.

Since $|s| = 2p + p^2 \geq p$ $\therefore$ $s = uvxyz$ such that

1. $uv^ixy^iz \in L$ for $i \geq 0$
2. $|vy| > 0$ or $|vy| \geq 1$
3. $|vxy| \leq p$

Consider the following all possible cases and sub-cases.

1. Both v & y contain the same symbol.
   a. both v & y are in a^p
      pump down uv^0xy^0z
      no. of as $= \boxed{n < p}$; $a^n b^p c^{p^2} \notin L$ (contradiction) since $\boxed{np < p^2}$
   b. both v & y are in b^p
      pump down uv^0xy^0z
      no. of bs $= n < p$; $a^p b^n c^{p^2} \notin L$ (contradiction) since $pn < p^2$
   c. both v & y are in c^{p^2}
      pump down uv^0xy^0z
      no. of cs $= n < p^2$; $a^p b^p c^n \notin L$ (contradiction) since $p^2 > n$ $\underline{(n < p^2)}$
2. Both v & y contain only 1 type of symbol.
   Due to condition 3 ($|vxy| \leq p$):
   a. as do not appear → v has only bs & y has only cs.
      suppose $v = b^m$ & $y = c^n$
      According to condition 1, $uv^0xy^0z \in L$ i.e. $a^p b^{p-m} c^{p^2-n} \in L$ (pumping down)
      $p(p-m) = p^2 - n$ is impossible b/c according to conditions 2 & 3
      $1 \leq m+n \leq p \Rightarrow m+n \geq 1$ && $m+n \leq p$
      $p(p-m) = p^2 - pm \leq p^2 - p \underbrace{< p^2 - n}_{\text{contradiction}}$ (for $m \geq 1$; $n < p$)
      $p^2 \neq p^2 - n$; $\underbrace{p^2 > p^2 - n}_{\text{contradiction}}$ (for $m = 0$; $n \geq 1$; $a^p b^p c^{p^2-n}$)
      these are the only 2 possibilities for m
   b. bs do not appear → either as or cs must appear in v or y. (according to condition 2, $|vy| > 0$)
      if as appear, uv^0xy^0z (pumping down); no. of as $= n < p$; $a^n b^p c^{p^2} \notin L$ (contradiction) since $np < p^2$
      if cs appear, uv^0xy^0z (pumping down); no. of cs $= n < p^2$; $a^p b^p c^n \notin L$ (contradiction) since $p^2 > n$ $(n < p^2)$
   c.☆
3. v or y contains more than 1 type of symbol i.e. as and bs or bs and cs.
   uv^2xy^2z (pumping up) $\notin L$ (contradiction) since as and bs or bs and cs will be mixed (i.e. not appear in correct order).

In none of the above all possible cases s can be pumped $\therefore$ L is not a CFL.

---

c.☆ cs do not appear → v has only as & y has only bs.
$v = a^m$ & $y = b^n$
According to condition 1, $uv^0xy^0z \in L$ i.e. $a^{p-m} b^{p-n} c^{p^2} \in L$ (pumping down).
$(p-m)(p-n) = p^2$ is impossible b/c according to condition 2 both m and n cannot be 0 (zero).

Scanned by CamScanner

Q. Prove $a^i b^j c^k \mid k = j * i$ is non context language using PL for CFL

Assume L is RL with pumping length P.

Let $S = a^P b^P c^{P*P} = a^P b^P c^{P^2}$ [$n(c) = n(a) * n(b)$

$|S| = 2p + p^2 > p$ so it should fulfil 3 conditions of PL of CFL.

(i) $uv^ixy^iz \in L$ for $i \geq 0$ (ii) $|vy| > 0$ (iii) $|vxy| \leq p$.

Considering all possible divisions of vxy.

1) vxy contain all as ⇒ if $i = 2$ $uv^2xy^2z = a^{P+|v|+|y|} b^P c^{P^2}$

⇒ $n(a) * n(b) \neq n(c)$

as $|vy| \geq 1$

2) vxy contain all bs ⇒ if $i = 2$ $uv^2xy^2z = a^P b^{P+|v|+|y|} c^{P2}$

as $|vy| \geq 1$ ⇒ $n(a) * n(b) \neq n(c)$

3) vxy contain all cs ⇒ if $i = 2$ $uv^2xy^2z = a^P b^P c^{P2+|v|+|y|}$

as $|vy| \geq 1$ $n(a) * n(b) \neq n(c)$

4) vxy contain some as and some bs.

↳ if v contains on as & y contains on bs

$i = 2$ $a^{P+|v|} b^{P+|y|} c^{P^2}$ $n(a) * n(b) \neq n(c)$

↳ if v or y contain string ~~ab~~ both ab.
pumping up will change the order and new strings will no be in L.

5) vxy contain some bs and some cs

↳ 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 $|y| \leq p$

↳ if v or y contain some bs and some c then pumping up will change the order & new string will not be in L .

As for all the possible cases of vxy,
pumping up make new string that $\notin L$
Therefore L is not a CFL .