SRN

PES University, Bangalore

(Established under Karnataka Act No. 16 of 2013)

UE17CS204

END SEMESTER ASSESSMENT (ESA) B. TECH IV SEMESTER- MAY 2020

UE18CS254 – Theory Of Computation

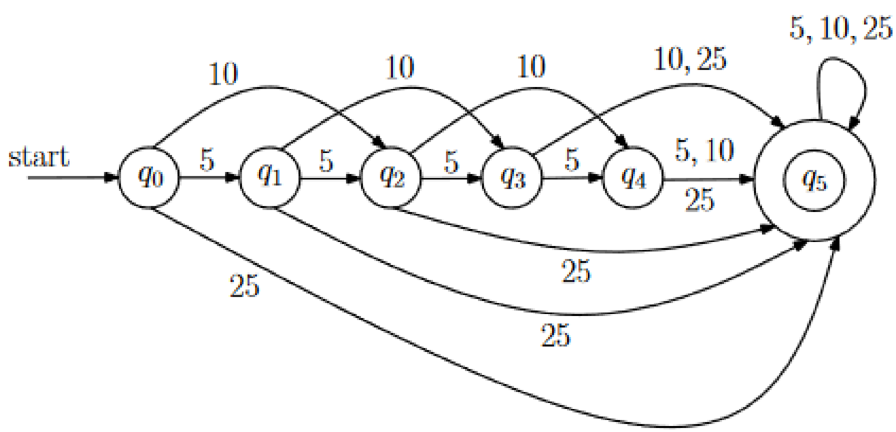
Time: 3 Hrs	Answer All Questions	Max Marks: 100

Note: All answers must be precise and to the point.

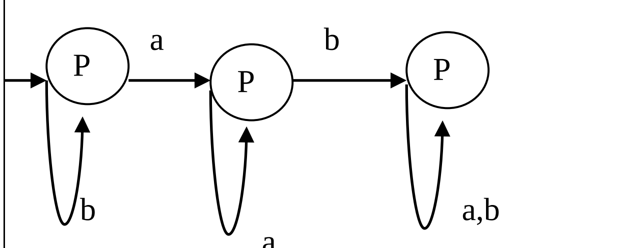
1. a) Consider a problem of designing a machine (or a "computer") that controls a toll gate. When a car arrives at the toll gate, the gate is closed. The gate opens as soon as the driver has paid coins worth `25. We assume that we have only three coin denominations: `5, `10, and `25. We also assume that no excess change is returned. After having arrived at the toll gate, the driver inserts a sequence of coins into the machine. At any moment, the machine has to decide whether or not to open the gate, i.e., whether or not the driver has inserted the coins worth `25 (or more). **10**

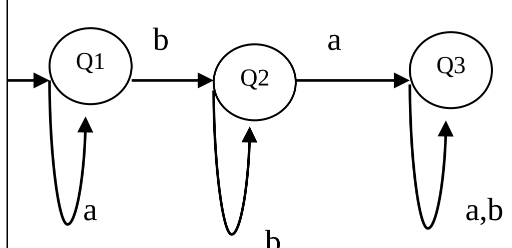
Observe that this machine has a property that it only needs to remember which state it is in at any given time. Draw a DFA for depicting the behavior of the machine for $\sum = \{5, 10, 25\}$. Assume that initial state is q0. In this state the car arrives at the toll gate and the machine has not collected any money yet.

<u>Sol:</u>



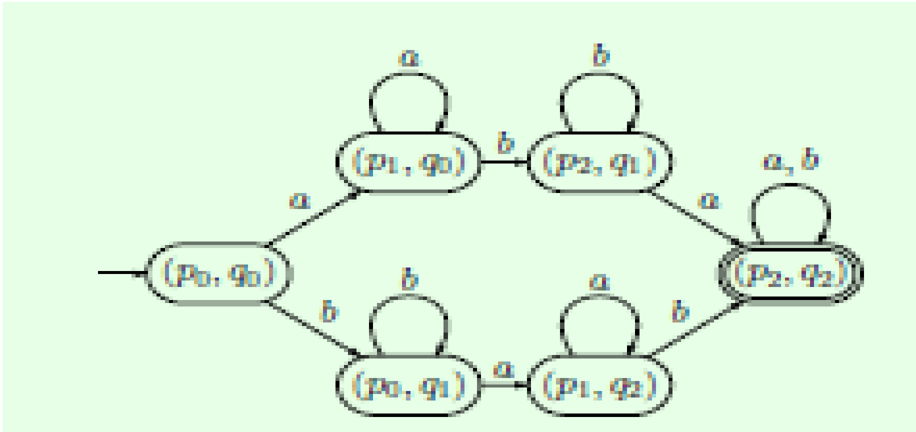
b) Construct the product of the following two DFAs that accepts the intersection of the languages of the two DFAs. **10**





<table>
<tr><td></td><td></td><td>

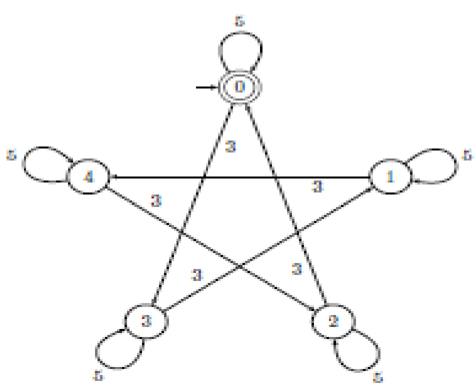
Sol:

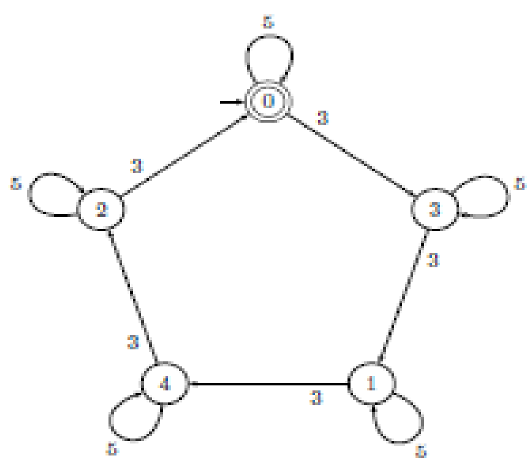


</td><td></td></tr>
<tr><td colspan="4"></td></tr>
<tr><td>2</td><td>a)</td><td>

Consider the language L over the alphabet ∑ = {3, 5} of all words for which the arithmetic sum of the constituent symbols is divisible by 5. For example, ε ∈ L since 0 is divisible by 5), 555 ∈ L (5 + 5 + 5 = 15 which is divisible by 5), and 335333 ∈ L (3 + 3 + 5 + 3 + 3 + 3 = 20 which is divisible by 5), but not 333
Is L a regular language or not? If it is, construct a DFA A such that L(A) = L.
If L is not a regular language, prove this by using the pumping lemma for regular languages.

Sol:





</td><td>13</td></tr>
<tr><td></td><td>b)</td><td>

Let ∑ =(a,b). Let double be the function from ∑* to ∑* that doubles each character in a string. For example, double(baaba) = bbaaaabbaa.
For the definition of double, say true or false for the following statements.

i. If L is a regular language over ∑, then double(L) is regular.
ii. If L is a finite language over ∑, then double(L) is finite.
iii. If L ⊆ ∑* is not regular, then double(L) is not regular.
iv. If L1 and L2 are regular languages over ∑, then L1∪ L2 is regular.

</td><td>7</td></tr>
</table>

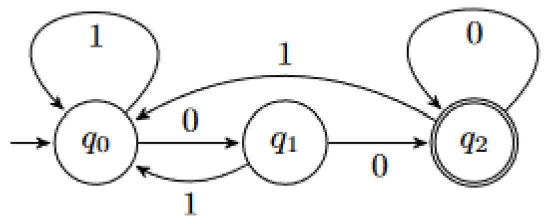
SRN

v. If L1 and L2 are finite languages over ∑, then L1 ∪ L2 is finite.
vi. If L1 ⊆ ∑* and L2 ⊆ ∑*are not regular, then L1 ∪ L2 is not regular.
vii. double(Ɛ) = Ɛ

Sol:

(i) True	(ii) True	(iii) True	(iv) True	(v) True	(vi) False	(vii) True

3. a) Construct a context-free grammar for the following DFA: **7**



Sol:

The language of the DFA is defined by the grammar $G = (V, \Sigma, R, S_0)$ with $V = \{S_0, S_1, S_2\}$, $\Sigma = \{0, 1\}$, and R being the following set of rules:

$$S_0 \to 0S_1 \mid 1S_0$$
$$S_1 \to 0S_2 \mid 1S_0$$
$$S_2 \to 0S_2 \mid 1S_0 \mid \epsilon$$

b) Show that the grammar ({S},{a,b},R,S) with rules R=S→aS|aSbS|∈ is ambiguous. **6**

Sol:

Consider the string aab. We can give two different leftmost derivations of this string: $S \rightsquigarrow$ **aS** $\rightsquigarrow a$**aSbS** $\rightsquigarrow aabS \rightsquigarrow aab$ and $S \rightsquigarrow$ **aSbS** $\rightsquigarrow a$**aS**$bS \rightsquigarrow aabS \rightsquigarrow aab$.

c) What is wrong with the following "proof" that $a^nb^{2n}a^n$ is context free? **7**

Step1: Both $\{a^nb^n: n>=0\}$ and $\{b^na^n:n>=0\}$ are context free
Step2: $a^nb^{2n}a^n=\{a^nb^n\}\{b^na^n\}$
Step3: since the context free languages are closed under concatenation, $a^nb^{2n}a^n$ is context free

Sol:

(1) is fine. (2) is fine if we don't over interpret it. In particular, although both languages are defined in terms

of the variable n, the scope of that variable is a single language. So within each individual language definition, the two occurrences of n are correctly interpreted to be occurrences of a single variable, and thus the values must be same both times. However, when we concatenate the two languages, we still have two separate language definitions with separate variables. So the two n's are different. This is the key. It means that we can't assume that, given $\{a^n b^n\}\{b^n a^n\}$, we choose the same value of n for the two strings we choose. For example, we could get $a^2b^2b^3a^3$, which is $a^2b^5a^3$, which is clearly not in $\{a^n b^{2n} a^n\}$.

4. a) Give a grammar in Chomsky Normal Form that generates the same language as the grammar G= (V,Σ,R,S) with V={S,X,Y}, Σ ={a,b,c}, and R being the following set of rules: **13**

$$
\begin{aligned}
S &\to XY \\
X &\to abb \mid aXb \mid \epsilon \\
Y &\to c \mid cY
\end{aligned}
$$

Sol:

Using the algorithm from the lecture, we get the grammar $G' = (V', \Sigma, R', S)$ with $V = \{S, X, X_1, X_2, Y, A, B, C\}$, $\Sigma = \{a, b, c\}$, and R' being the following set of rules:

$$
\begin{aligned}
S_0 &\to S \\
S &\to XY \\
X &\to abb \mid aXb \mid \epsilon \\
Y &\to c \mid cY
\end{aligned}
$$

$$
\begin{aligned}
S_0 &\to S \\
S &\to XY \mid Y \\
X &\to abb \mid aXb \mid ab \\
Y &\to c \mid cY
\end{aligned}
$$

$$
\begin{aligned}
S_0 &\to S \\
S &\to XY \mid c \mid cY \\
X &\to abb \mid aXb \mid ab \\
Y &\to c \mid cY
\end{aligned}
$$

$$
\begin{aligned}
S_0 &\to S \\
S &\to XY \mid c \mid cY \\
X &\to aX_1 \mid aX_2 \mid ab \\
X_1 &\to bb \\
X_2 &\to Xb \\
Y &\to c \mid cY
\end{aligned}
$$

$$
\begin{aligned}
S &\to XY \mid c \mid CY \\
X &\to AX_1 \mid AX_2 \mid AB \\
X_1 &\to BB \\
X_2 &\to XB \\
Y &\to c \mid CY \\
A &\to a \\
B &\to b \\
C &\to c
\end{aligned}
$$

b) Use the pumping lemma to prove the following language is not CFL — 7

$\{ww^R w \mid w \in \{a,b\}^*\}$

<u>Sol:</u>

SRN

Assume that language $L = \{ww^Rw \mid w \in \{a,b\}^*\}$ is context-free. By the pumping lemma, there is a number k such that every string in L with length k or more can be written $uvwxy$ where

(i) $length(vwx) \leq k$

(ii) v and x are not both null

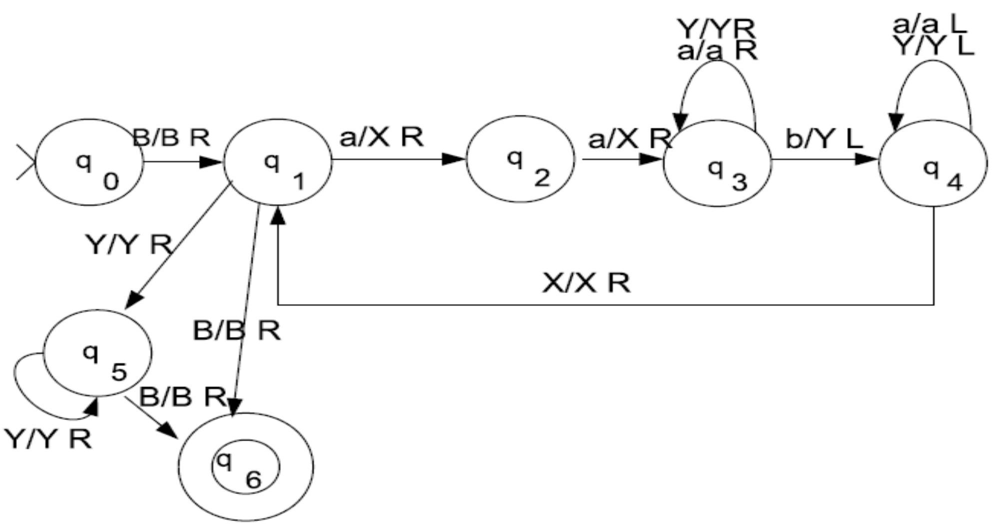
(iii) $uv^iwx^iy \in L$ for all $i \geq 0$.

The string $z = (a^kb^k)(a^kb^k)^R(a^kb^k) = a^kb^{2k}a^{2k}b^k$ must have a decomposition $uvwxy$ that satisfies the preceding conditions. By condition (ii), we have v and x have at least one terminal. Without loss of generality, assume that at least one a is in v or x (similar argument for the case of at least one b in v or x). Condition (i) requires the length of vwx to be at most k. This implies that the substring vwx of z cannot contain a's from both sides of b^{2k}. If the a's in the substring vwx of z are before b^{2k}, then uv^2wx^2y increases the number of a's before b^{2k} while keeping the number of a's after b^{2k} the same as $2k$. Hence uv^2wx^2y is no long in $L = \{ww^Rw \mid w \in \{a,b\}^*\}$. If the a's in the substring vwx of z are after b^{2k}, we have $uv^2wx^2y \notin L$ by similar argument. Therefore L is not context-free.

5. a) Design a Standard Turing Machine with $\Sigma=\{a,b\}$ that accepts the language L **11**

$L=\{a^{2i}b^i \mid i>=0\}$

<u>Sol:</u>



b) State ***true or false*** for the following statements: **5**

i. A Turing machine has a single start state, but may have many accept states.

ii. It is possible to make a Turing machine with only one state.

iii. A Turing machine halts when its head reaches the end of its input

iv. All decidable languages are regular languages.

v. A nondeterministic TM can recognize more languages than a deterministic TM.

Sol:

i. A Turing machine has a single start state, but may have many accept states.

True

ii. It is possible to make a Turing machine with only one state.

True

iii. A Turing machine halts when its head reaches the end of its input.

False

iv. All decidable languages are regular languages.

False

v. A nondeterministic TM can recognize more languages than a deterministic TM.

False

c) Classify each of the following problems as either (4)

(D) decidable,
(R) recognizable but not decidable,
(U) not recognizable

A. {<M> | M is a Turing machine that accepts at least 42 different strings}.
B. {<M> | M is a Turing Machine that has at least 42 states}.
C. {<M> | M is a Turing Machine that runs for at least 42 steps when started with a blank input tape}.
D. {<M> | L (M) is recognized by a Turing Machine that has an even number of states}.

Sol:

A. {<M> | M is a Turing machine that accepts at least 42 different strings}. ANS: R
B. {<M> | M is a Turing Machine that has at least 42 states}. D
C. {<M> | M is a Turing Machine that runs for at least 42 steps when started with a blank input tape}. D
D. {<M> | L (M) is recognized by a Turing Machine that has an even number of states}. R