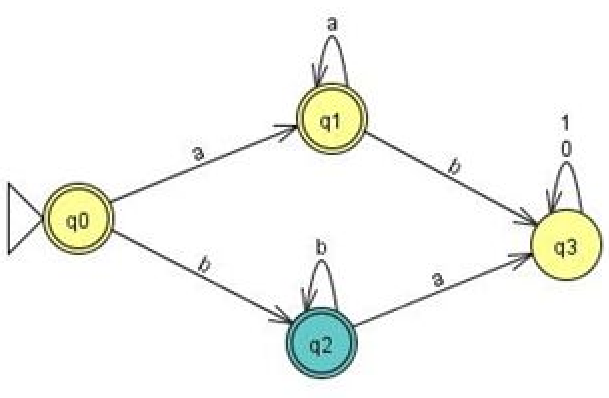
PES University, Bangalore
(Established under Karnataka Act No. 16 of 2013) UE19CS205

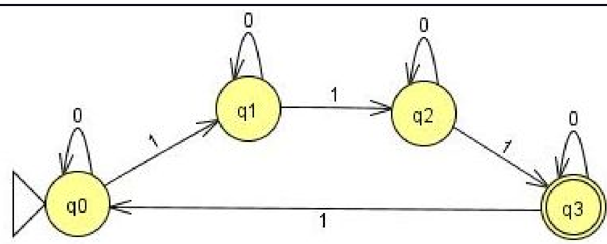
SAMPLE PAPER-II SOLUTION FOR

IN SEMESTER ASSESSMENT (ISA-1)- B.TECH III SEMESTER
October, 2020

Automata Formal Languages & Logic

Time: 2 Hrs | Answer All Questions | Max Marks: 60

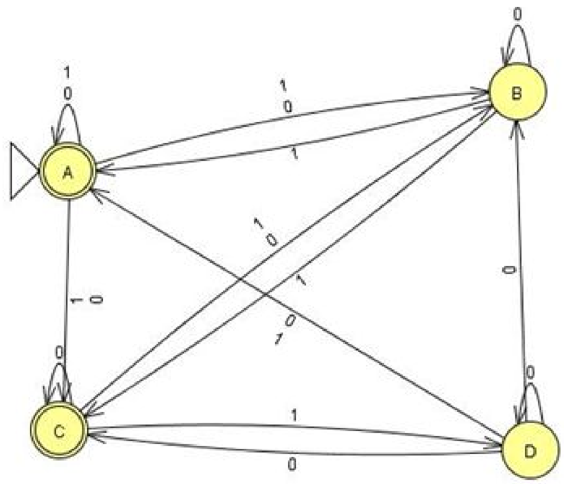
1.	a)	**Consider the function defined by the rule f(x)={2x+2 for 0<x<5}, specify the range and domain of the function.** **Solution:** Range:4,6,8,10 Domain:1,2,3,4	2
	b)	**Let L be the language , L={ w ∈ {a*} or w ∈{b}* , Σ {a,b}*} Construct a DFA that accepts all the strings that are in L and rejects all the strings that are not in L.** **Solution:**  	4
	c)	**Answer the following:** **i) Consider the following DFA,**	2+2



Give one sentence description of the above DFA
Solution:
L(DFA)= {n_1(w) mod 4=3, Σ = {0,1}*}

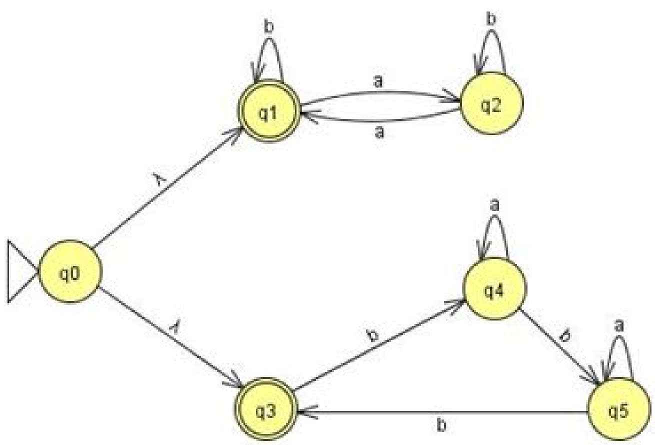
ii) Draw the transition diagram for the FA
M={(A,B,C,D),(0,1), δ ,c,(A,C)}
δ (A,0)=, δ (A,1)={A,B,C)
δ (B,0)=B, δ (B,1)={A,C}
δ (C,0)={B,C}, δ (C,1)={B,D}
δ (D,0)={A,B,C,D} δ (D,1)={A}

Solution:

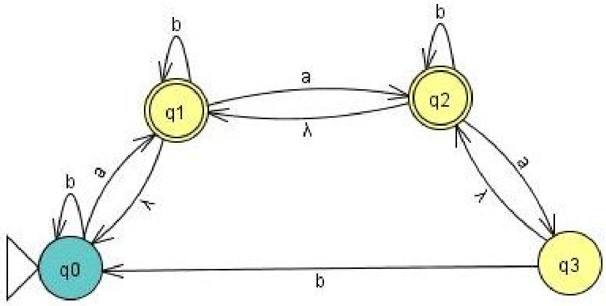


2 a) **Construct an NFA with six states that accepts the string over the alphabet {a,b} with either even number of a's or the number of b's is multiple of 3.** 6

Solution:



b) **Convert the following NFA to DFA** 4



Solution:

Transition table of λ-NFA:

state	**a**	**b**	**λ-closure**
→ q0	q1	q1	{q0}
q1*	q2	q1	{q0,q1}
q2*	q3	q2	{q1,q2}
q3	Φ	Φ	{q2,q3}

Transitions:

Start state of DFA is the λ-closure of start state of NFA i.e, q0

δ(q0,a): λ-closure(δ(q0,a)) ={q0,q1}

δ(q0,b): λ-closure(δ(q0,b)) ={q0}

δ({q0,q1},a): λ-closure(δ(q0,a)) U λ-closure(δ(q1,a)) ={q0,q1,q2}

δ({q0,q1},b): λ-closure(δ(q0,b)) U λ-closure(δ(q1,b)) ={q0,q1}

δ({q0,q1,q2},a): λ-closure(δ(q0,a)) U λ-closure(δ(q1,a)) U
λ-closure(δ(q2,a))
={q0,q1,q2,q3}

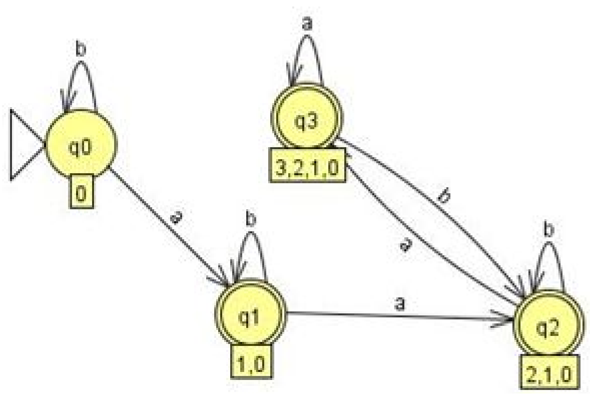
δ({q0,q1,q2},a): λ-closure(δ(q0,b)) U λ-closure(δ(q1,b)) U
λ-closure(δ(q2,b))
={q0,q1,q2}

δ({q0,q1,q2,q3},a): λ-closure(δ(q0,a)) U λ-closure(δ(q1,a)) U
λ-closure(δ(q2,a))U
λ-closure(δ(q3,a))={q0,q1,q2,q3}

δ({q0,q1,q2,q3},b): λ-closure(δ(q0,b)) U λ-closure(δ(q1,b)) U
λ-closure(δ(q2,b))U λ-closure(δ(q3,a)) ={q0,q1,q2}

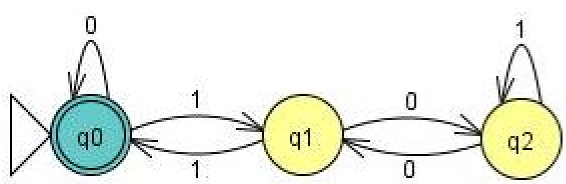
DFA transition table:

state	a	b
→ q0	{q0,q1}	{q0}
{q0,q1}*	{q0,q1,q2}	{q0,q1}
{q0,q1,q2}*	{q0,q1,q2,q3}	{q0,q1,q2}
{q0,q1,q2,q3}*	{q0,q1,q2,q3}	{q0,q1,q2}



3. a) **Give a regular expression that accepts a binary string whose decimal value is divisible by 3.** — 5

Solution:



Regular expression:
(0+1(01*0)*1)*

The regular expression is obtained by eliminating the states in the order: q2, q1, q0

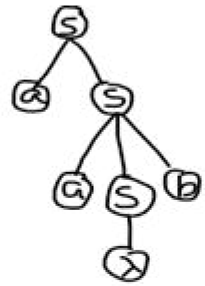
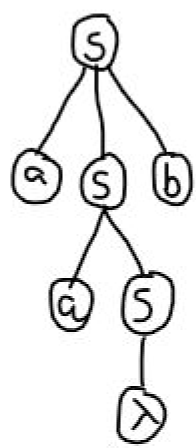
b) **Explain the closure properties of regular languages** — 5

Solution:

Closure properties on regular languages are defined as certain operations on regular languages which are guaranteed to produce another regular language as output.
Closure refers to some operation on a language, resulting in a new language that is of the same "type" as originally operated on.
Consider L and M are regular languages:
Regular languages are closed under the following operations.

- <u>Kleene Closure:</u>

		RS is a regular expression whose language is L, M. R* is a regular expression whose language is L*. ● <u>Positive closure:</u> RS is a regular expression whose language is L, M. R^+ is a regular expression whose language is L^+. ● <u>Complement:</u> The complement of a language L (with respect to an alphabet E such that E^* contains L) is E^*–L. Since E^* is surely regular, the complement of a regular language is always regular. ● <u>Reverse Operator:</u> Given language L, L^R is the set of strings whose reversal is in L. Example: L = {0, 01, 100}; L^R ={0, 10, 001}. Proof: Let E be a regular expression for L. We show how to reverse E, to provide a regular expression E^R for L^R. ● <u>Complement:</u> The complement of a language L (with respect to an alphabet E such that E^* contains L) is E^*–L. Since E^* is surely regular, the complement of a regular language is always regular. ● <u>Union</u>: Let L and M be the languages of regular expressions R and S, respectively.Then R+S is a regular expression whose language is(L U M). ● <u>Intersection:</u> Let L and M be the languages of regular expressions R and S, respectively then it a regular expression whose language is L intersection M.	
4.	a)	**Consider the following grammar:** **S → aSb** **S → aS** **S → ϵ** **(a) Give a one-sentence description of the language generated by this grammar.** The language accepts all the strings that have any number a's or any number of a's followed by any number of b's, It also accepts an empty string. The language accepts the strings of the form : {a* + a+b+}* **(b) Show that this grammar is ambiguous by giving a string that can be parsed in two different ways. Draw both parse trees.**	6

Derivation 1:	Parse tree 1
Derivation 1: $S \Rightarrow^{lm} aS$ $\Rightarrow^{lm} aaSb$ $\Rightarrow^{lm} aab$	Parse tree 1 
Derivation 2: $S \Rightarrow^{lm} aSb$ $\Rightarrow^{lm} aaSb$ $\Rightarrow^{lm} aab$	Parse tree 2: 

(c) Give an unambiguous grammar that accepts the same language as the grammar above.

$S \rightarrow aS \mid A$

$A \rightarrow aAb \mid \lambda$

b)	**Let the alphabet be {a, b} and the language be the set of strings with more a's than b's. Show that this language is not regular using Pumping Lemma for regular languages.** **Solution:** ● The opponent claims that the language $L=\{n_a(w) > n_b(w) : w \{a,b\}^*\}$ is regular ● Let n be the number of states in the opponent's machine for L (Pumping length) ● Consider the string, $w=b^n a^{n+1}$ such that $\|w\|>=n$ and $w \in L$. ● We can break such a string w in three parts xyz, satisfying the condition $\|xy\| <=n$ and $\|y\|>0$ ● Since the first n symbols are all b's, the loop(y) is made up of only b's. Let us assume the string is broken as : $b^{n-m}(b)^m a^{n+1}$	4

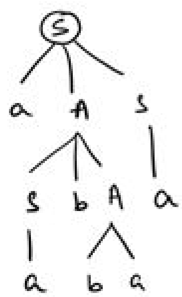
Here m is the number of b's over which the loop is framed.
1 <= m < n (as the loop must be made of minimum one symbol and must reside within the n states)

- $b^{n-m}(b^{m})^{i}a^{n+1}$
- if we choose m = 2 and i=4
  we get $b^{n-2}(b^{2})^{4}a^{n+1}$
  $= b^{n-2}(b^{2})^{4}a^{n+1}$
  $= b^{n+4}a^{n+1} \notin L$ because it does not have more a's than b's and we get a contradiction therefore Language is not regular

5. a) **Consider the following CFG** (6)
S -> aAS|a
A -> SbA | SS | ba
Answer the following questions:

i) What are the terminals, non-terminals and the start symbol of the grammar?
Solution: Terminals: a,b
Nonterminals: S,A
Start Symbol : S

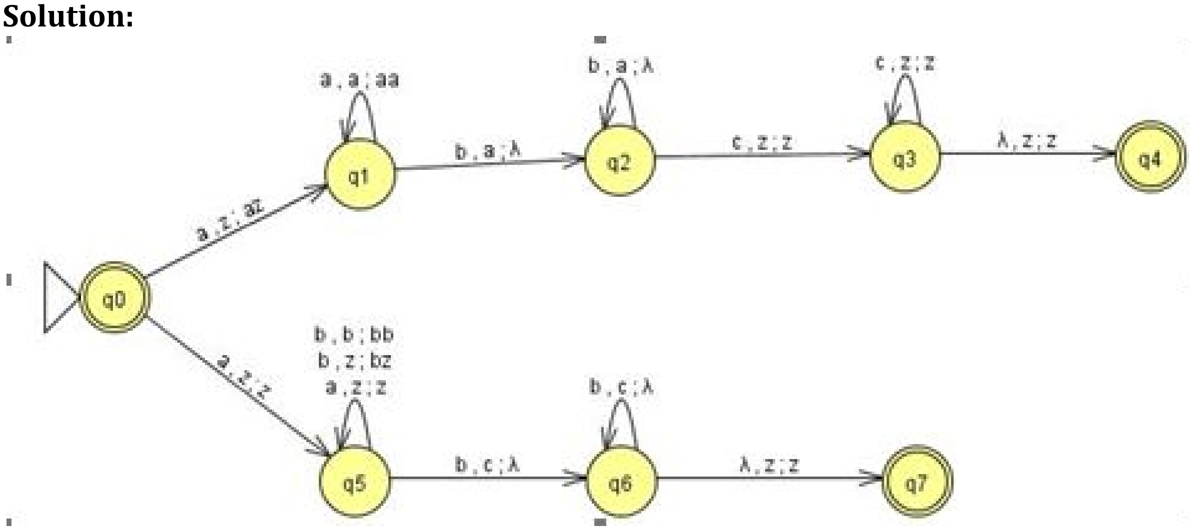
ii) Draw parse tree for the following: "aabbaa"
Parse tree :



iii) Give leftmost derivation for the above string.

$$S \overset{lm}{\Rightarrow} aAS \overset{lm}{\Rightarrow} aSbAS \overset{lm}{\Rightarrow} aabAS \overset{lm}{\Rightarrow} aabbaS \overset{lm}{\Rightarrow} aabbaa$$

b) **Give equivalent grammar in CNF for the following CFG** (4)
S -> aSbb | T
T -> bTaa | S | λ
Solution:
Remove λ, unit and useless production:
Step 1: Eliminate λ production:
<u>T -> λ :</u>
S -> aSbb | T | λ
T -> bTaa | baa | S

		S -> λ S-> aSbb | abb | T T -> bTaa | baa | S **Step 2: Eliminate unit production** **(S -> T and T -> S)** S -> aSbb | abb | bTaa | baa T -> bTaa | baa | aSbb | abb **Step 3:There are no useless production** A-> a B -> b S -> ASBB | ABB | BTAA | BAA | λ T -> BTAA | BAA | ASBB | ABB **Step 4:Conversion to CNF** S-> AF | AE |BD |BC | λ T-> AF | AE |BD|BC F-> SE E-> BB D-> AA C->TD B-> b A->a	
6.	a)	**Give PDA for the following language: D = { $a^i b^j c^k$ | i, j, k ≥ 0, and i = j or j = k }** **Solution:**  	4
	b)	**For the given grammar, check the acceptance of string w = 10010 using CYK Algorithm-** **S → XY | YZ** **X → YX | 0** **Y → ZZ | 1** **Z → XY | 0**	6

Solution:

SXZ				
Null	SXZ			
Null	Y	Y		
SX	Y	SZ	SX	
Y	XZ	XZ	Y	XZ
1	0	0	1	0

Since we get S in the topmost box, the string belongs to the language of the grammar.

Acknowledgement : The sample paper solution is prepared by Prof. Kavitha K N.