Tutorial No 1: Basics of Set Theory & PMI, Strong PMI

1. Suppose A and B are sets. Give venn diagram to represent $(A-B) \cup (B-A) \cup (A \cap B)$ and using same, find simple expression.
2. Give truth table and find simple statement for $P \vee \neg(P \rightarrow Q)$
3. What is the relationship between $2^{A \cup B}$ and $2^A \cup 2^B$.
4. Show that for any language L, $L^* = (L^*)^* = (L^+)^* = (L^*)^+$
5. Find relation between $L_1(L_2 \cap L_3)$ and $L_1L_2 \cap L_1L_3$
6. Prove that for every n ≥ 0,
$$\sum_{i=1}^{n} i^2 = n(n+1)(2n+1)/6$$
7. Prove that for any string x, $|x^r| = |x|$
8. Give recursive definitions of each of the following sets.
    a. The set N of all natural numbers.
    b. The set S of all integers divisible by 7.
    c. The set U of all strings in $\{0,1\}^*$ containing the substring 00.

Tutorial No 2: Finite Automata

Q:1 In each case below, find a string of minimum length in {a, b}* not in the language corresponding to the given regular expression.

a. b* (ab) *a*
b. (a*+b*)(a*+b*)(a*+b*)
c. a*(baa*)*b*
d. b*(a+ba) *b*

Q:2 Consider the two regular expressions

r =a*+b*
s = ab*+ba*+b*a+(a*b) *

a. Find a string corresponding to r but not to s
b. Find a string corresponding to s but not to r
c. Find a string corresponding to both r and s
d. Find a string in { a, b } * corresponding to neither r nor s

Q:3 Find the regular expression and finite automaton for following languages. Σ={a,b}

1. The language of all strings containing exactly two a's.
2. The language of all strings containing at least two a's.
3. The language of all strings that do not end with ab
4. The language of all strings that begin or end with aa or bb
5. The language of all strings not containing the substring aa.

Q:4 Explain in brief any 3 applications of Finite State Machine.

Q.5 For the following sets, write the corresponding regular expression:

a. {1,12,112,1112,11112......}
b. {0,1}
c. {a^2, a^4, a^6, a^8, a^{10},.......}
d. {a^x | x is divisible by 3 or 5}

Tutorial No:3 More Examples on Finite Automata

Q:1 Solve the following MCQs with proper justification.

1 Which of the following is / are true?

(i). $(0^*1)^* = (0+1)^*$ (ii) $(0+1)^*01(0+1)^*+1^*0^* \neq (0 + 1)^*$

(A) (i) only (B) (ii) only (C) (i) and (ii) (D) None of these

2 Consider the following finite state machine. Now if the language accepted by the given DFA is $(a+b(b+aa)^*ab)^*$ then the final state of the machine is, ($q0$:initial state)

(A) q0 (B) q1 and q0 (c) q2 (D) None of these

State	$\delta(q,a)$	$\delta(q,b)$
q_0	q_0	q_1
q_1	q_2	q_1
q_2	q_1	q_0

3 Which of the following regular expression is equivalent to $(a+b)^* a(a+b)^* b(a+b)^*$

(A) $(a+b)^* ab(a+b)^*$
(B) $a(a+b)^* b(a+b)^*$
(c) $(a+b)^* a(a+b)^* a(a+b)^*$
(D) $(a+b)^* b(a+b)^* a(a+b)^*$

4 A finite state machine with the following state table has a single input 'x' and a single output 'z' if the initial state is unknown, then the shortest input sequence to reach the final state 'c' is

(A) 01 (B) 10 (C) 101 (D) 110

Present State	Next state z	
	X=1	X=0
A	D,0	B,0
B	B,1	C,1
C	B,0	D,1
D	B,1	C,0

5 Consider the DFA given below: Initial state is q0 and final state is q1

State	δ(q,a)	δ (q,b)
q_0	Q_1	Q_2
q_1	q_2	q_1
q_2	Q_2	Q_2

This automation accepts the language

(A) $L = \{a_n b_n \mid n \geq 0\}$

(B) $L = \{a_n b_n \mid n \geq 1\}$

(C) $L = \{a_n b \mid n \geq 0\}$

(D) $L = \{a b_n \mid n \geq 0\}$

Q.2 Draw the Finite Automata over alphabet set {0,1} which when considered as a binary number is divisible by 5.

Q-3 Explain the Mealy Machine and Moore Machine. For the following Mealy Machine find the equivalent Moore Machine, consider q2 is the start state.

	Input Symbol			
Current State	a		B	
	Next State	Output	Next State	Output
q_0	q_1	1	q_3	1
q_1	q_1	0	q_0	1
q_2	q_0	1	q_2	0
q_3	q_3	0	q_1	1

Tutorial No 4 Non Deterministic Finite Automata

Q.1 In the NFA given below, Calculate each of the following:

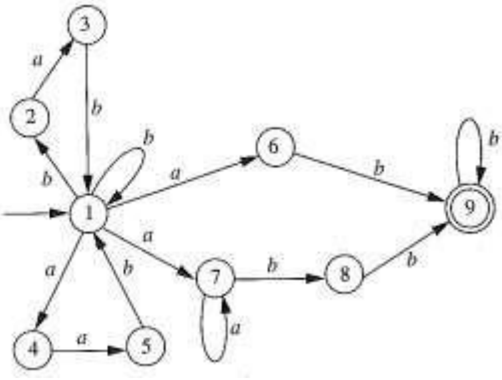
a. $\delta^*(1, bb)$

b. $\delta^*(1, bab)$

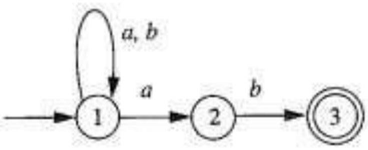
c. $\delta^*(1, aabb)$

d. $\delta^*(1, aabbab)$

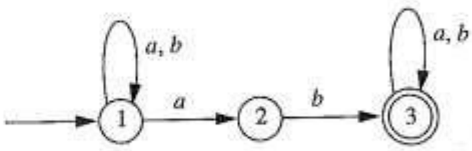
e. $\delta^*(1, aba)$



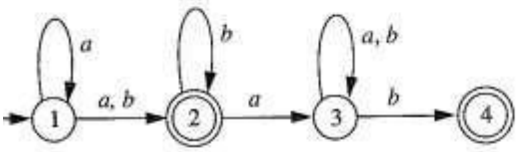
Q:2 Convert the following NFA to DFA



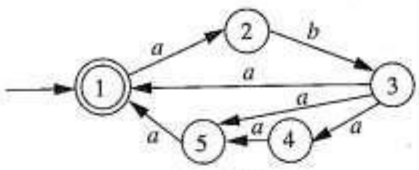
(a)



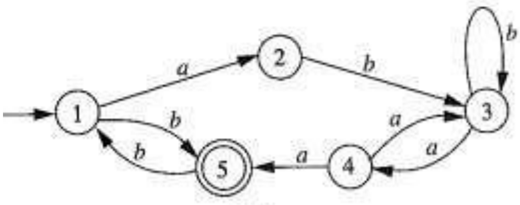
(b)



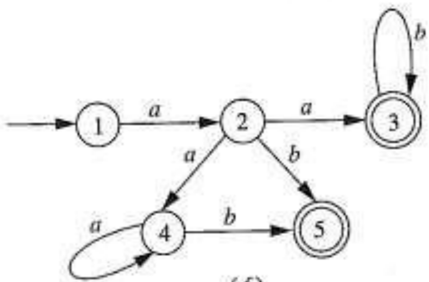
(c)



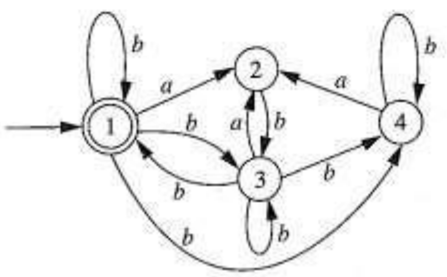
(d)



(e)

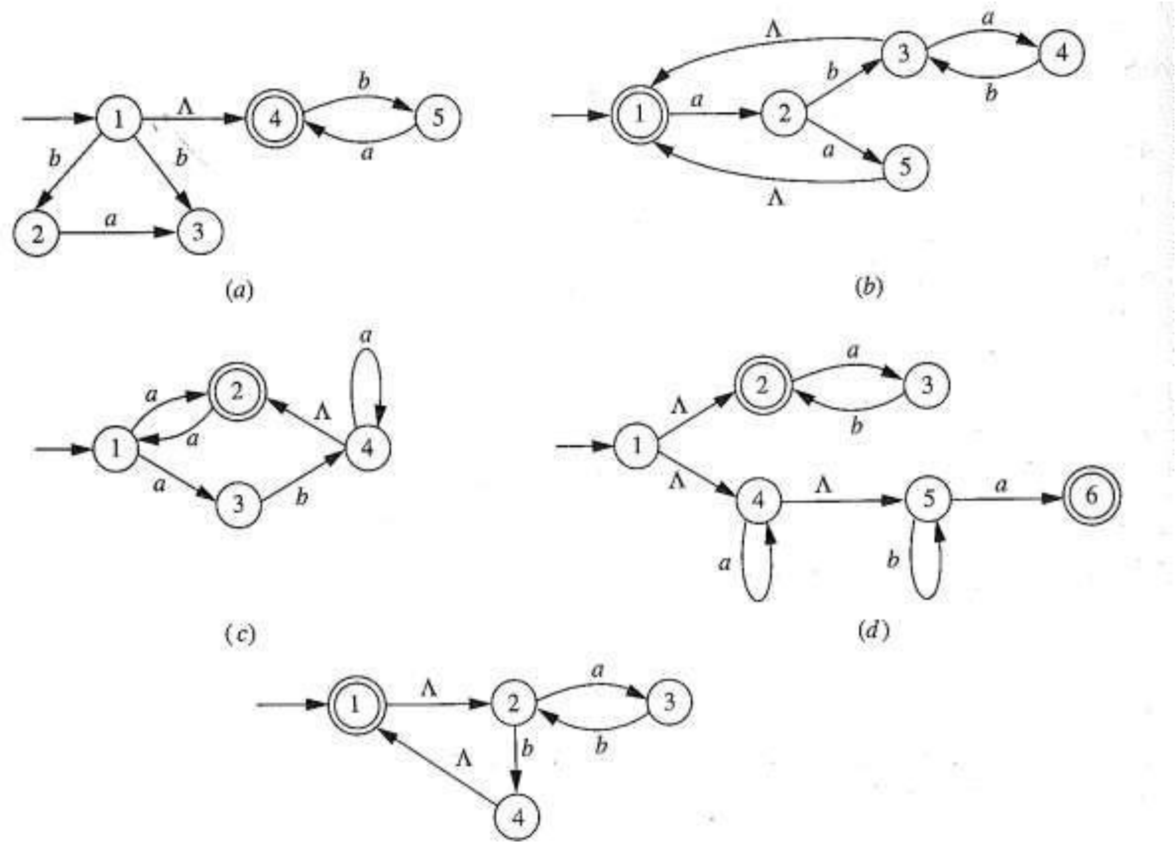


(f)



(g)

Q:3 Convert the following NFA-^ to corresponding DFA



Nirma University

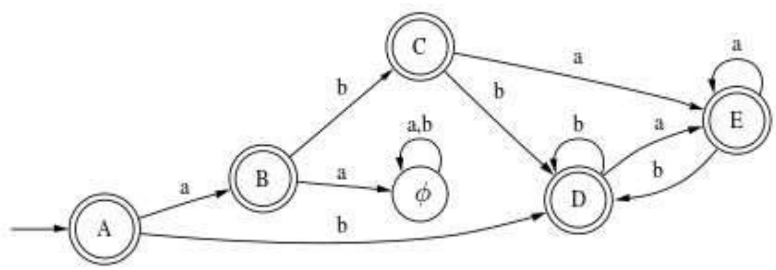
B Tech Sem V (CSE)

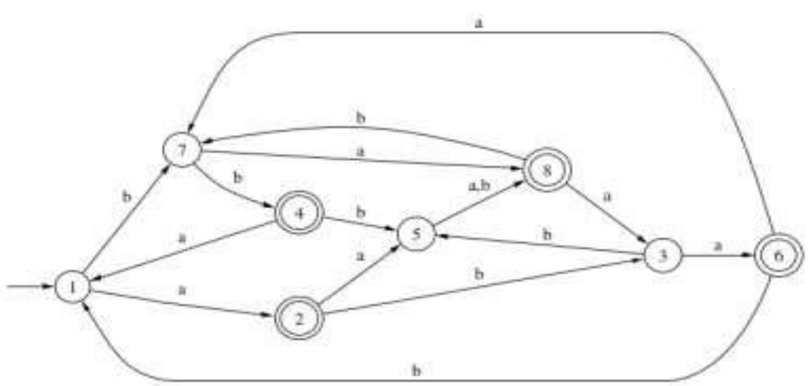
Sub: 2CS601 Theory of Computation

Tutorial No 5

Minimization of an Finite Automata and Pumping Lemma

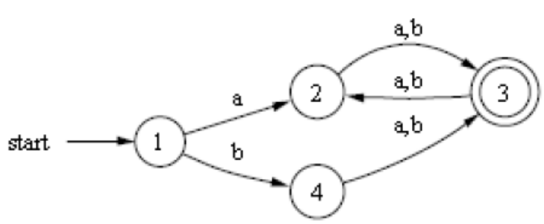
Q:1 Minimize the following FAs



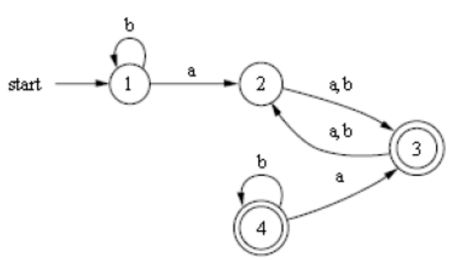


Q-2 Which languages are accepted by the following automata:

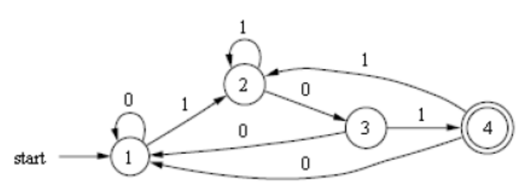
(i)



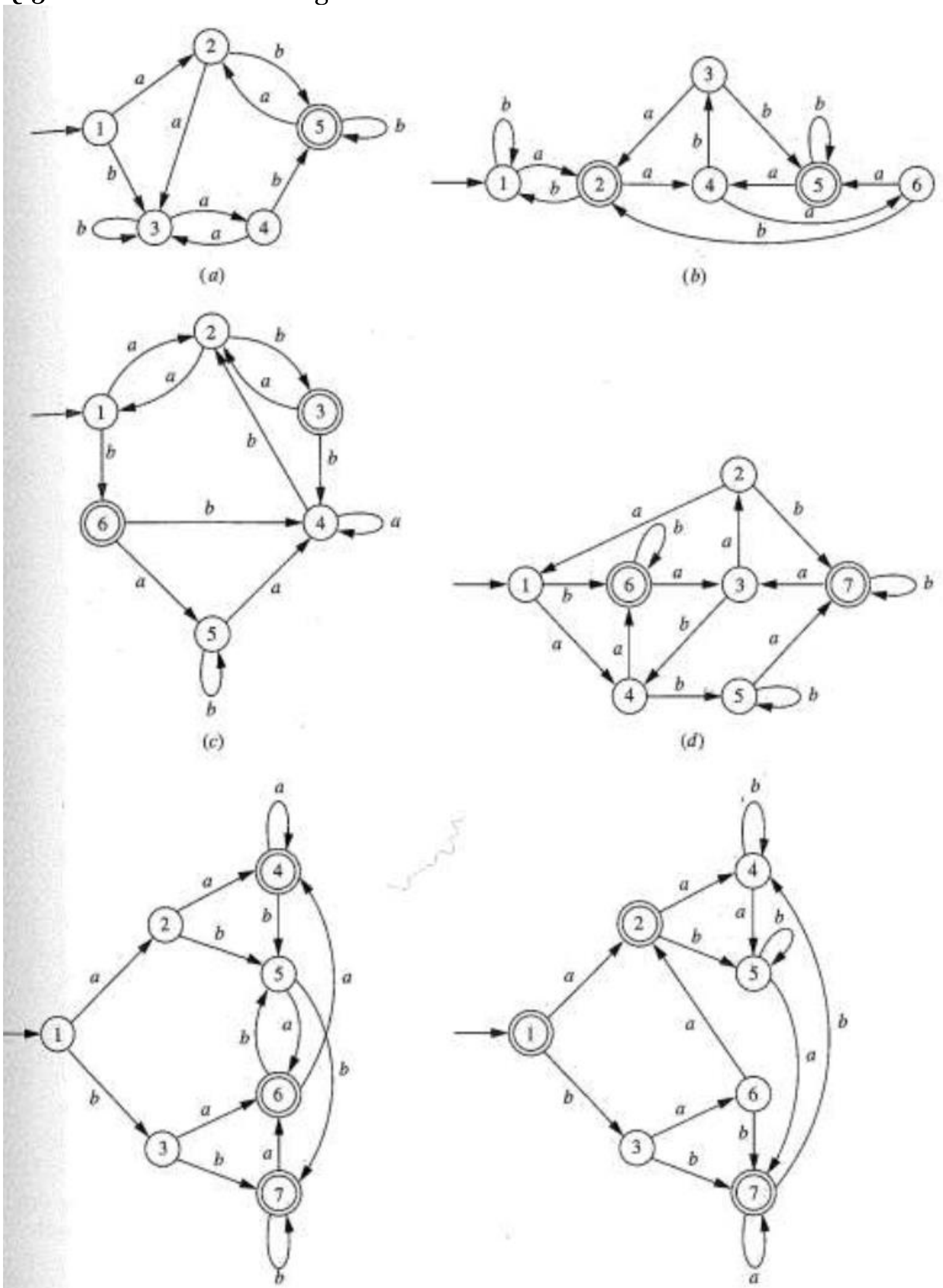
(ii)



(iii)



Q:3 Minimize the following FAs



Q:4 Prove that the following languages are regular or not.

1. L = {$a_i b_i$ | i ≥ 0}
2. L = {xx | x ∈ {0, 1}*}
3. L = {WW_r | W Є{a,b}* and |W| = 2} (Hint: W_r is a reverse of string w, |W| is a length of W)
4. L = {$a_n b_m c_k$ | n, m, k ≥ 1}
5. L = {a_n | n is even}
6. L = {a_n | n is odd}
7. L = {a_n | n is prime number}

Institute Of Technology, Nirma University

B.Tech. (CE) Sem VI

2CS601 Theory of Computation

Tutorial 6

Q-1 Design CFG for the following languages:

a. $\{a^i b^j c^k \mid i = j + k\}$

b. $\{a^i b^j c^k \mid j = i + k\}$

c. $\{a^i b^j c^k \mid j = i \text{ or } j = k\}$

d. $\{a^i b^j c^k \mid i = j \text{ or } i = k\}$

e. $\{a^i b^j c^k \mid i < j \text{ or } i > k\}$

f. $\{a^i b^j \mid i \leq 2j\}$

g. $\{a^i b^j \mid i < 2j\}$

h. $\{a^i b^j \mid i \leq j \leq 2i\}$

Q.2 In each each case, what languages are generated for the following CFGs:

1. S → aSa | bSb | ∧
2. . S → aSa | bSb | a | b
3. . S → aSb | bSa | ∧
4. S → aSa | bSb |aAb | bAa|
   A → aAa | bAb | a | b | ∧
5. S → aS | bS | a
6. S → SS | bS | a
7. S → SaS | b
8. S → aT | bT |∧
   T → aS bS

Q.3 Consider the CFG with productions S → aSbScS | aScSbS | bSaScS | bScSaS | cSaSbS | cSbSaS | ∧. Does this generate the language $\{x \in \{a,b,c\}^* \mid n_a(x) = n_b(x) = n_c(x)\}$? Prove your answer.

Q:4 Convert the following CFG to Chomsky Normal Form:

1. S → aAbB
   A → Ab / b
   B → Ba / a
2. S → aA | bB
   A → bAA | a
   B → BBa | b
3. S →aAC
   A →aB|bAB

B→b
C→c

4. S→0X1Y
   X → 0X | 0
   Y →1Y / 1
5. S → abSab | a | aAAb
   A → bS | aAAb | c

Q:5 Explain the term ambiguity and prove that the following grammar is ambiguous grammar.
S → S+S | S-S| S*S| S-S|a

Q.6 Remove unit productions from the following grammar and generate equivalent grammar:
1. S → ABC | 0
   A → 1
   B → C | 0
   C → D
   D → E
   E → 2
2. S →ABCD|0
   A→ BC|1
   B→C
   C→D
   D→d

Institute Of Technology, Nirma University

B.Tech. (CSE) Sem VI

2CS601 Theory of Computation

Tutorial 7 (CFG Examples)

Q-1 Design CFG for the following languages, $\Sigma = \{0,1\}^*$

1. $L = \{0^n 1^{2n} \mid n >= 0\}$
2. String of even length.
3. Alternate sequence of 0 and 1.
4. $a^n\ b^n\ c^m \mid n, m >=1$
5. $a^n\ b^n\ c^m \mid n, m >=0$
6. $a^n\ b^n\ c^m\ d^m \mid n, m >=0$

Q.2 Remove unit production from the following:

S → ABCD
A → a
B → C | b
C → D
D → c

Q.3 Remove null productions from the following.

1.
S → ABC | A0A
A → 0A | B0C | 000 | B
C → CA | AC
D → ∧

2.
S → AAA | B
A → 0A | B
B → ∧

Q.4 Describe the language generated by the following CFGs:

S → aA | bC |b
A → aS | bB
B → aC | bA | a
C → aB | bS

Q.5 Convert the following grammar into CNF:

S → AACD
A → aAb | ∧
C → aC | a
D → aDa | bDb | ∧

Q.6 Remove unit productions from the following grammar and generate equivalent grammar:

S →ABCD|0
A→ BC|1
B→C
C→D
D→d

Institute Of Technology, Nirma University
B.Tech. (CSE) Sem VI
2CS601 Theory of Computation
Tutorial 8 (Push Down Automata)

Q-1 Design PDA for the following:

1. $L_1 = \{a^m\, c\, b^m \mid m \geq = 0\}$
2. $L_2 = \{a^m\, b^m\, c \mid m \geq = 0\}$
3. $L_3 = \{c\, a^m\, b^m \mid m \geq = 0\}$
4. $L_4 = \{a^n\, c\, b^m \mid n,\ m \geq = 0\}$
5. $L_5 = \{a^n\, b^m\, c \mid n,\ m \geq = 0\}$
6. $L_6 = \{\, c\, a^n\, b^m \mid n,\ m \geq = 0\}$
7. $L_7 = \{a^n\, c\, b^m \mid n,\ m \geq = 1\}$
8. $L_8 = \{a^n\, b^m\, c \mid n,\ m \geq = 1\}$
9. $L_9 = \{\, c\, a^n\, b^m \mid n,\ m \geq = 1\}$
10. More number of a's than b's
11. $L = \{a^n\, b^{2n} \mid n \geq 1\}$
12. $L = \{a^n\, b^m\, c \mid n \geq 1\}$

Q-2 Design PDA for the following CFGs and trance the string 0001101110

1. $S \rightarrow 0B \mid 1A$
   $A \rightarrow 0S \mid 1AA \mid 0$
   $B \rightarrow 1S \mid 0BB \mid 1$

2. $S \rightarrow b \mid bS \mid aSS \mid SSa \mid SaS$

Q-3 Give PDA for the following CFG and trace the string 01010101

$S \rightarrow XSX \mid Y$
$X \rightarrow 0 \mid 1$
$Y \rightarrow 0Z1 \mid 1Z0$
$Z \rightarrow XZX \mid X$

Q-4 Give a CFG for the following PDA

1. $\delta(q_0, a, Z_0) \vdash (q_0, aZ_0)$
   $\delta(q_0, a, a) \vdash (q_0, aa)$
   $\delta(q_0, c, a) \vdash (q_1, a)$
   $\delta(q_1, a, a) \vdash (q_2, \epsilon)$
   $\delta(q_2, a, a) \vdash (q_2, \epsilon)$
   $\delta(q_2, \epsilon, Z_0) \vdash (q_2, \epsilon)$

2. $\delta(q_0, 1, Z_0) \vdash (q_0, KZ_0)$
$\delta(q_0, \epsilon, Z_0) \vdash (q_0, \epsilon)$
$\delta(q_0, 1, K) \vdash (q_0, KK)$
$\delta(q_0, 0, K) \vdash (q_1, K)$
$\delta(q_1, 0, K) \vdash (q_1, \epsilon)$
$\delta(q_1, 0, Z_0) \vdash (q_0, Z_0)$

3.

Move Number	State	Input	Stack Symbol	Move(s)
1	q_0	a	Z_0	(q_0, AZ_0)
2	q_0	b	Z_0	(q_0, BZ_0)
3	q_0	a	A	(q_0, AA)
4	q_0	b	A	(q_0, BA)
5	q_0	a	B	(q_0, AB)
6	q_0	b	B	(q_0, BB)
7	q_0	c	Z_0	(q_1, Z_0)
8	q_0	c	A	(q_1, A)
9	q_0	c	B	(q_1, B)
10	q_1	a	A	(q_1, Λ)
11	q_1	b	B	(q_1, Λ)
12	q_1	Λ	Z_0	(q_1, Λ)

Q:5 Design a PDA for Odd length and Even length palindrome and trace the strings: aabbaa, abcba and aaabbb.

Q:6 In both cases below, a transition table is given for a PDA with initial state q_0 and Accepting state q_2. Describe in each case the language that is accepted.

Move Number	State	Input	Stack Symbol	Move(s)
1	q_0	a	Z_0	(q_0, XZ_0)
2	q_0	b	Z_0	(q_0, XZ_0)
3	q_0	a	X	(q_0, XX)
4	q_0	b	X	(q_0, XX)
5	q_0	c	X	(q_1, X)
6	q_0	c	Z_0	(q_1, Z_0)
7	q_1	a	X	(q_1, Λ)
8	q_1	b	X	(q_1, Λ)
9	q_1	Λ	Z_0	(q_2, Z_0)

Nirma University

2CS601 Theory of Computation

BTech (CSE)

Tutorial No 9

Q 1: Identify the nonterminals and terminals in the following grammars.

(1) S -> Aba | b	A -> BB | aa	B -> bB | c	C -> cC | d
(2) S -> XY1 | 0	X -> 00X | 1	Y -> 1X1	
(3) S -> XY	X -> YSY	X -> YY | a	Y -> aXb | b
(4) S -> XY	X -> YSY	X -> YY | 1	Y -> 0X1 | 1

Q 2: Convert the following CFG to CNF:

(1) S -> aAC	A -> aB | bAB	B -> b	C -> c
(2) S -> 0X1Y	X -> 0X | 0	Y -> 1Y | 1	

(3) S -> abSab | a |aAAb   A -> bS | aAAb | c

Q 3: Identify and remove the nonreachable nonterminals from the following grammars:

(1) S - > XY1 | 0	X -> 00X | 1	Y -> 1X1	Z -> 00	
(2) S -> XZ | 0	X -> YA | 1	Y -> Z1 | A2	A -> 01	B-> X | 2

Q 4: Identify Language

(1) L={ $a^i b^i c^i$ | i>=1 }
   a. Regular Language
   b. CFL
   c. Both CFL & Regular
   d. Neither CFL nor Regular

(2) L={ $a^i b^i c^j$ | I,j>=1 }
   a. Regular Language
   b. CFL
   c. Both CFL & Regular
   d. Neither CFL nor Regular

(3) L={ $a^n b^n c^m d^m$| n,m>=1 }
   a. Regular Language
   b. CFL
   c. Both CFL & Regular

d. Neither CFL nor Regular

(4) $L=\{ 0^n1^m2^{m+n} \mid n,m>=1 \}$

a. Regular Language
b. CFL
c. Both CFL & Regular
d. Neither CFL nor Regular

Q 5: Define Property

(1) CFLs are closed under

a. Union
b. Complementation
c. Intersection
d. All the above

(2) The CFLs and regular languages are both closed over

a. Union
b. Complementation
c. Intersection
d. None of the above

(3) The CFLs and regular languages are both closed over

a. Difference
b. Intersection
c. Complement
d. Concatenation

(4) CFLs are not closed under

a. Union
b. Concatenation
c. Intersection
d. Homomorphism

Q 6:

The regular expression corresponding to the CFG S -> aS | bS | a | b is

a. a+b
b. (a+b)*
c. (a+b)*(a+b)
d. None of the above

(1) The CFG corresponding to the language $L=\{0^k1^k \mid k>=1\}$ is

a. S -> 0S1 | 01
b. S -> 0S1 | 01 | ϵ
c. S -> 0A1, A -> 01
d. All the above

(2) The CFL L={$a^n b^n$ | n>0} can be generated by the following CFG:

a. S -> ϵ | ab | aSb
b. S -> ab | aSb
c. S -> ϵ | aSb
d. All of the above

Nirma University

2CS601 Theory of Computation (CSE)

Tutorial No 10

Turing Machines

Q-1 Solve the following:

1. Design a TM over $\Sigma = \{1\}$ to accept the language $L = \{ 1^m \mid m \text{ is odd}\}$
2. Design a TM over $\Sigma = \{0\}$ to accept the language $L = \{ 0^m \mid m \text{ is even}\}$
3. Design a TM over $\Sigma = \{0\}$ to accept the language $L = \{ 0^m \mid m \text{ is multiple of } 3\}$
4. Design a TM over $\Sigma = \{1\}$ to accept the language $L = \{ 1^m \mid m \text{ is odd}\}$
5. Design a TM to accept the language $\{a,b\}^* \{aba\}\{a,b\}^*$
6. Design a TM to accept the language $L = \{a^n b^n \mid n \geq 1\}$
7. Design a TM to accept the language $L = \{a^n b^n c^n \mid n \geq 1\}$

Q:2 Design a Turing Machine to accept the language of odd length and even length palindrome. Trace the strings: ababa, abbb, abbbba

Q:3 Design a Turing Machine to accept the language of $\{ SS \mid S \in \{a,b\}^*\}$. Trace the string aaabbaaabb

Q:4 Design a Turing Machine for reversing a string.

Q:5 Design a Turing Machine for copying a string.

Q:6 Design a Turing Machine for deleting a symbol.

Q-7 Multiple Choice Questions:

(1) A Turing Machine is more powerful than the PDA because
   a. The head can move in both directions
   b. The current input symbol can be changed
   c. The tape is infinite
   d. All of the above

(2) A PDA can behave like Turing Machine when
   a. It has no stack
   b. It has two or more stacks
   c. It has a stack of infinite size
   d. All of the above

(3) The difference between an LBA and a Turning machine is that
   a. The LBA has limited number of states
   b. The LBA has an additional stack
   c. The length of the tape in the LBA is limited

d. All of the above

(4) A Post Machine has

a. An auxiliary queue
b. An auxiliary queue and a stack
c. Two auxiliary stacks
d. Two auxiliary queues

(5) A Turing Machine can be simulated by a semi-infinite tape

a. With two tracks
b. With one track
c. With three tracks
d. Cannot be simulated by a semi-infinite tape

Q-8 (Extra Problems for Practice)

1: Design a Turing Machine M to compute $\sum_{k=1}^{n} \mathrm{k}$ for a given positive integer n.

2: Design a Turing Machine M over {0,1} such that L(M)={$0^{2n}1^{n}$|n>=1}

3: Design a Turing Machine M over {0,1,2,3} such that L(M)={$0^{2n}1^{n}2^{n}3^{2n}$|n>=1}

4: Design a Two track Turing Machine M to compute $\sum_{k=1}^{n} \mathrm{k}$ for a given positive integer n.

5: Design a Turing Machine M to find the successor of a positive integer.