STANLEY
COLLEGE OF ENGINEERING & TECHNOLOGY FOR WOMEN
(Approved by AICTE , New Delhi | Affiliated to Osmania University ,Hyderabad)
Address : Chapel Road, Abids ,Hyderabad

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AutomataTheory,Languages and Computation

Third Year CSE(Sem:I)

2 marks Questions and Answers

UNIT I

1. **Why are switching circuits called as finite state systems?**

A switching circuit consists of a finite number of gates, each of which can be in any one of the two conditions 0 or 1.Although the voltages assume infinite set of values,the electronic circuitry is designed so that the voltages corresponding to 0 or 1 are stableand all others adjust to these value. Thus control unit of a computer is a finite statesystem.

2. **What is a : (a) String (b) Regular language**

A string x is accepted by a Finite Automaton M=(Q, Σ, δ.q0,F) if δ (q0,x)=p, for some p in F.FA accepts a string x if the sequence of transitions corresponding to the symbols of x leads from the start state to accepting state.

The language accepted by M is L(M) is the set {x | _(q0,x) is in F}. A language is regular if it is accepted by some finite automaton.

3. **Define: (i) Finite Automaton(FA) (ii)Transition diagram**

FA consists of a finite set of states and a set of transitions from state to state that occur on input symbols chosen from an alphabet _. Finite Automaton is denoted by a 5- tuple(Q,Σ, δ,q0,F), where Q is the finite set of states , _ is a finite input alphabet, q0 in Q is the initial state, F is the set of final states and _ is the transition mapping function Q * _ to Q.

Transition diagram is a directed graph in which the vertices of the graph correspond to the states of FA. If there is a transition from state q to state p on input a, then there is an arc labeled ‘ a ‘ from q to p in the transition diagram.

4. . **What are the applications of automata theory?**

_ In compiler construction.
_ In switching theory and design of digital circuits.
_ To verify the correctness of a program.
_ Design and analysis of complex software and hardware systems.
_ To design finite state machines such as Moore and mealy machines.

5. **What is Moore machine and Mealy machine?**

A special case of FA is Moore machine in which the output depends on the state of the machine. An automaton in whch the output depends on the transition and current input is called Mealy machine.

6. **What are the components of Finite automaton model?**

The components of FA model are Input tape, Read control and finite control.

(a)The input tape is divided into number of cells. Each cell can hold one i/p symbol

. (b)The read head reads one symbol at a time and moves ahead.

(c)Finite control acts like a CPU. Depending on the current state and input symbol read from the input tape it changes state.

7. **Differentiate NFA and DFA**

NFA or Non Deterministic Finite Automaton is the one in which there exists many paths for a specific input from current state to next state. NFA can be used in theory of computation because they are more flexible and easier to use than DFA .

Deterministic Finite Automaton is a FA in which there is only one path for a specific input from current state to next state. There is a unique transition on each input symbol.(Write examples with diagrams).

8. **What is _-closure of a state q0?**

_-closure(q0) denotes a set of all vertices p such that there is a path from q0 to p labeled _. Example :closure(q0)={q0,q1}

9. **.Give the examples/applications designed as finite state system.**

Text editors and lexical analyzers are designed as finite state systems. A lexical analyzer scans the symbols of a program to locate strings corresponding to identifiers, constants etc, and it has to remember limited amount of information .

10. **Define automaton.**

Automaton is a abstact computing device. It is a mathematical model of a system,with discrete inputs,outputs,states and set of tranitions from state to state that occurs on input symbols from alphabet Σ.

11. what is the principle of mathematical induction.

Let P(n) be a ststement about a non negative integer n. Then the principle of mathematical induction is that P(n) follows from

(i) P(1) and

(ii) P(n-1) implies P(n) for all n>1.

Condition(i) is called the basis step and condition (ii) is called the inductive step. P(n-1) is called the induction hypothesis.

12. List any four ways of theorem proving

(i) Deductive
(ii) If and only if
(iii) Induction
(iv) Proof by contradiction.

13. Define TOC

TOC describes the basic ideas and models underlying computing. TOC suggests various abstract models of computation, represented mathematically.

14. What are the applications of TOC?

Compiler Design
Robotics
Artificial Intelligence
Knowledge Engineering.

15. Define Transition Diagram.

Transition Diagram associated with DFA is a directed graph whose vertices corresponds to states of DFA, The edges are the transitions from one state to another.

16. What are the properties of Transition Function(δ)

(i) $\delta(q.\varepsilon)=q$
(ii) For all strings w and input symbol a

$\Delta(q,aw)= \delta(\delta(q.a),w)$

$\Delta(q,wa)= \delta(\delta(q,w).a)$

(iii) The transition function δ can be extended that operates on states and strings.

17. Lists the operations on Strings.

(i) Length of a string
(ii) Empty string
(iii) Concatenation of string
(iv) Reverse of a string
(v) Power of an alphabet
(vi) Kleene closure
(vii) Substring
(viii) Palindrome

18. Lists the operations on Languages.

(i) Product
(ii) Reversal
(iii) Power
(iv) Kleene star

(v) Kleene plus
(vi) Union
(vii) Intersection

19. Define Graphs.

A graph denoted by G=(V,E) consists of a finite set of vertices (or) nodes V and a set E, a pair of vertices called edges.

20. Define Substring.

A string v appears within another string w(w=uv) is called “substring of w.” IF w=uv,then substrings u & v are said to be prefix and suffix of w respectively.

UNIT II

1. What is a regular expression?

A regular expression is a string that describes the whole set of strings according to certain syntax rules. These expressions are used by many text editors and utilities to search bodies of text for certain patterns etc. Definition is: Let _ be an alphabet. The regular expression over _ and the sets they denote are:

i. _ is a r.e and denotes empty set.
ii. _ is a r.e and denotes the set {_}
iii. For each ‘a’ in _ , a+ is a r.e and denotes the set {a}.
iv. If ‘r’ and ‘s’ are r.e denoting the languages R and S respectively then (r+s),
(rs) and (r*) are r.e that denote the sets RUS, RS and R* respectively.

2. Differentiate L* and L+

_
L* denotes Kleene closure and is given by L* =U Li i=0
example : 0* ={_ ,0,00,000,…………………………………}
Language includes empty words also.

_
L+ denotes Positive closure and is given by L+= U Li i=1 q0 q1

3. What is Arden’s Theorem?
Arden’s theorem helps in checking the equivalence of two regular expressions. Let P and Q be the two regular expressions over the input alphabet _. The regular expression R is given as : R=Q+RP Which has a unique solution as R=QP*.

4. Write a r.e to denote a language L which accepts all the strings which begin or end with either 00 or 11.

The r.e consists of two parts:

L1=(00+11) (any no of 0’s and 1’s) =(00+11)(0+1)*

L2=(any no of 0’s and 1’s)(00+11) =(0+1)*(00+11)
Hence r.e R=L1+L2 =[(00+11)(0+1)*] + [(0+1)* (00+11)]

5. Construct a r.e for the language over the set _={a,b} in which total number of a’s are divisible by 3

   (b* a b* a b* a b*)*

6. what is: (i) (0+1)* (ii)(01)* (iii)(0+1) (iv)(0+1)+

   (0+1)*= { _ , 0 , 1 , 01 , 10 ,001 ,101 ,101001,…………………}

   Any combinations of 0’s and 1’s.

   (01)*={_ , 01 ,0101 ,010101 ,…………………………………..}

   All combinations with the pattern 01.

   (0+1)= 0 or 1,No other possibilities.

   (0+1)+= {0,1,01,10,1000,0101,………………………………….}

7. Reg exp denoting a language over _ ={1} having (i) even length of string (ii) odd length of a string

   (i) Even length of string R=(11)*

   (ii) Odd length of the string R=1(11)*

8. Reg exp for: (i) All strings over {0,1} with the substring ‘0101’ (ii) All strings beginning with ’11 ‘ and ending with ‘ab’ (iii) Set of all strings over {a,b}with 3 consecutive b’s. (iv) Set of all strings that end with ‘1’and has no substring ‘00’

   (i)(0+1)* 0101(0+1)*

   (ii)11(1+a+b)* ab

   (iii)(a+b)* bbb (a+b)*

   (iv)(1+01)* (10+11)* 1

9. Construct a r.e for the language which accepts all strings with atleast two c’s over the set Σ={c,b}

   (b+c)* c (b+c)* c (b+c)*

10. What are the applications of Regular expressions and Finite automata Lexical analyzers and Text editors are two applications.

    Lexical analyzers:

    The tokens of the programming language can be expressed using regular expressions.

    The lexical analyzer scans the input program and separates the tokens.For eg identifier can be expressed as a regular expression as: (letter)(letter+digit)*

    If anything in the source language matches with this reg exp then it is recognized as an identifier.The letter is{A,B,C,………..Z,a,b,c….z} and digit is {0,1,…9}.Thus reg exp identifies token in a language.

    Text editors:

These are programs used for processing the text. For example UNIX text editors uses the reg exp for substituting the strings such as: S/bbb*/b/

Gives the substitute a single blank for the first string of two or more blanks in a given line. In UNIX text editors any reg exp is converted to an NFA with Єtransitions, this NFA can be then simulated directly.

11. .Reg exp for the language that accepts all strings in which 'a' appears tripled overthe set Σ ={a}

    reg exp=(aaa)*

12. .What are the applications of pumping lemma?

    Pumping lemma is used to check if a language is regular or not.

    (i) Assume that the language(L) is regular.
    (ii) Select a constant 'n'.
    (iii) Select a string(z) in L, such that |z|>n.
    (iv) Split the word z into u,v and w such that |uv|<=n and |v|>=1.
    (v) You achieve a contradiction to pumping lemma that there exists an 'i' Such that uvi

w is not in L.Then L is not a regular language.

13. What is the closure property of regular sets?

The regular sets are closed under union, concatenation and Kleene closure.

r1Ur2= r1 +r2

r1.r2= r1r2

(r)*=r*

The class of regular sets are closed under complementation, substitution, homomorphism and inverse homomorphism.

14. .Reg exp for the language such that every string will have atleast one 'a' followed by atleast one 'b'.

    $R=a^+b^+$

15. Write the exp for the language starting with and has no consecutive b's .

    reg exp=(a+ab)*

16. Lists on the closure properties of Regular sets.
    (i) Union
    (ii) Concatenation
    (iii) Closure
    (iv) Complementation
    (v) Intersection
    (vi) Transpose
    (vii) Substitutions

(viii) Homomorphism

17. Let R be any set of regular languages. IsUR regular? Prove it.
Yes. Let P,Q be any two regular languages .As per theorem
L(R)=L(P UQ)
=L(P+Q)
Since ‘+’ is a operator for regular expresstions L(R) is also regular.

18. Show that $(r^*)^*=r^*$ for a regular expression r,
$(r^*)^*==\{\varepsilon,r,rr,\ldots\ldots\ldots\ldots\}= r^*$

19. What are the three methods of conversion of DFA to RE?
(i) Regular Expression equation method
(ii) Arden’s Theorem.
(iii) State elimination technique,

20. What are the algorithms of minimization DFA?
(i) Myhill-Nerode Theorem
(ii) Construction of π_{final} from π.

UNIT III

1. What are the applications of Context free languages
Context free languages are used in :
(i) Defining programming languages.
(ii) Formalizing the notion of parsing.
(iii) Translation of programming languages.
(iV) String processing applications.

2. What are the uses of Context free grammars?
- Construction of compilers.
- Simplified the definition of programming languages.
- Describes the arithmetic expressions with arbitrary nesting of balanced parenthesis { (,) }.
- Describes block structure in programming languages.
- Model neural nets.

3. Define a context free grammar
A context free grammar (CFG) is denoted as G=(V,T,P,S) where V and T are finite set of variables and terminals respectively. V and T are disjoint. P is a finite set of productions each is of the form A->_ where A is a variable and _ is a string of symbols from (V U T)*.

4. What is the language generated by CFG or G?

The language generated by G (L(G)) is {w | w is in T* and S=>w.
That is a G string is in L(G) if:

(1) The string consists solely of terminals.

(2) The string can be derived from S.

5. .What is : (a) CFL (b) Sentential form

L is a context free language (CFL) if it is L(G) for some CFG G.

A string of terminals and variables α is called a sentential form if: S => α ,where S is the start symbol of the grammar.

6. What is the language generated by the grammar G=(V,T,P,S) where P={S->aSb, S->ab}?

S=> aSb=>aaSbb=>…………………………..=>$a^n b^n$

Thus the language L(G)={ $a^n b^n$ | n>=1}.The language has strings with equal number of a's and b's.

7. What is :(a) derivation (b)derivation/parse tree (c) subtree

(a) Let G=(V,T,P,S) be the context free grammar. If A-> β is a production of P and α and γ are any strings in (VUT)* then $\alpha A \gamma \Rightarrow \alpha\beta\gamma$

(b) A tree is a parse \ derivation tree for G if:

(i) Every vertex has a label which is a symbol of VU TU{_}.

(ii) The label of the root is S.

(iii) If a vertex is interior and has a label A, then A must be in V.

(iv) If n has a label A and vertices n1,n2,….. nk are the sons of the vertex n in order from left with labels X1,X2,………..Xk respectively then A X1X2…..Xk must be in P.

(v) If vertex n has label _ ,then n is a leaf and is the only son of its father.

(c) A subtree of a derivation tree is a particular vertex of the tree together with all its descendants ,the edges connecting them and their labels.The label of the root may not be the start symbol of the grammar.

8. If S->aSb | aAb , A->bAa , A->ba .Find out the CFL

soln. S->aAb=>abab

S->aSb=>a aAb b =>a a ba b b(sub S->aAb)

S->aSb =>a aSb b =>a a aAb b b=>a a a ba b bb

Thus L={$a^n b^m a^m b^n$, where n,m>=1}

9. What is a ambiguous grammar?

A grammar is said to be ambiguous if it has more than one derivation trees for a sentence or in other words if it has more than one leftmost derivation or more than one rightmost derivation.

10. Find CFG with no useless symbols equivalent to : S→AB | CA ,
B→BC | AB, A→a , C→aB | b.

S-> AB
S->CA
B->BC
B->AB
A->a
C->aB
C->b are the given productions.

A symbol X is useful if S => αXβ => w

The variable B cannot generate terminals as B->BC and B->AB. Hence B is useless symbol and remove B from all productions. Hence useful productions are: S->CA , A->a , C->b

11. Construct CFG without Є production from : S →a | Ab | aBa , A →b | Є , B →b | A.

S->a
S->Ab
S->aBa
A->b
A->Є
B->b
B->A are the given set of production.

A->Є is the only empty production. Remove the empty production

S-> Ab , Put A->Є and hence S-> b.

If B-> A and A->Є then B ->Є

Hence S->aBa becomes S->aa .

Thus S-> a | Ab | b | aBa | aa
A->b
B->b

Finally the productions are: S-> a | Ab | b | aBa | aa
A->b
B->b

12. What are the three ways to simplify a context free grammar?
(i) removing the useless symbols from the set of productions.
(ii) By eliminating the empty productions.
(iii) By eliminating the unit productions.

13. What are the properties of the CFL generated by a CFG?

Each variable and each terminal of G appears in the derivation of some word in L .here are no productions of the form A->B where A and B are variables.

14. Find the grammar for the language L={a^{2n} bc ,where n>1 }

let G=({S,A,B}, {a,b,c} ,P , {S}) where P:

S->Abc
A->aaA | Є

15. .Find the language generated by :S->0S1 | 0A | 0 |1B | 1
A->0A | 0 , B->1B | 1

The minimum string is S-> 0 | 1
S->0S1=>001
S->0S1=>011
S->0S1=>00S11=>000S111=>0000A111=>00000111
Thus L={ $0^n 1^m$ | m not equal to n, and n,m >=1}

16. Construct the grammar for the language L={ $a^n b a^n$ | n>=1}.

The grammar has the production P as:
S->aAa
A->aAa | b
The grammar is thus : G=({S,A} ,{a,b} ,P,S)

17. . Construct a grammar for the language L which has all the strings which are all palindrome over Σ={a, b}.

G=({S}, {a,b} , P, S)
P:{ S -> aSa ,
S-> b S b,
S-> a,
S->b,
S->Є } which is in palindrome.

18. Differentiate sentences Vs sentential forms

A sentence is a string of terminal symbols.

A sentential form is a string containing a mix of variables and terminal symbols or all variables.This is an intermediate form in doing a derivation.

19. .Define Pushdown Automata.

A pushdown Automata M is a system (Q, Σ, Ґ ,δ ,q_0, Z_0,F) here
Q is a finite set of states.
Σ is an alphabet called the input alphabet.
Ґ is an alphabet called stack alphabet.
q_0 in Q is called initial state.
Zo in Ґ is start symbol in stack.
F is the set of final states.
Δ is a mapping from Q X (Σ U {Є}) X Ґ to finite subsets of Q X Ґ*.

20. 3.Specify the two types of moves in PDA.

The move dependent on the input symbol(a) scanned is:
δ(q,a,Z) = { (p1, γ1), (p2,γ2),……..(p^m,γ^m) }
where q qnd p are states , a is in Σ ,Z is a stack symbol and

γi is in $Г^*$. PDA is in state q , with input symbol a and Z the top symbol on state enter state p iReplace symbol Z by string γi

The move independent on input symbol is (Є-move):
$\delta(q,Є,Z)= \{ (p1,\gamma 1), (p2,\gamma 2),\ldots\ldots\ldots\ldots (p^m,\gamma^m) \}$. Is that PDA is in state q , independent of input symbol being scanned and with Z the top symbol on the stack enter a state p i and replace Z by γi.

21. What are the different types of language acceptances by a PDA and define them.

For a PDA $M=(Q, \Sigma, Г, \delta, q0, Z0, F)$ we define :

(i) Language accepted by final state L(M) as:

$* \{ w \mid (q0, w, Z0) \mid\!-- (p, Є, \gamma)$ for some p in F and γ in $Г^* \}$.

(ii) Language accepted by empty / null stack N(M) is:

$\{ w \mid (q0,w,Z0) \mid\!---- (p, Є, Є)$ for some p in $Q\}$.

22. Is it true that the language accepted by a PDA by empty stack and final states are different languages.

No, because the languages accepted by PDA 's by final state are exactly the languages accepted by PDA's by empty stack.

23. Define Deterministic PDA.

A PDA $M =(Q, \Sigma, Г, \delta, q0, Z0, F)$ is deterministic if:

- For each q in Q and Z in Г , whenever $\delta(q,Є,Z)$ is nonempty then $\delta(q,a,Z)$ is empty for all a in Σ.
- For no q in Q , Z in Г , and a in Σ U { Є} does $\delta(q,a,Z)$ contains more than one element. (Eg): The PDA accepting {wcw R | w in (0+1) * }.

24. Define Instantaneous description(ID) in PDA.

ID describe the configuration of a PDA at a given instant.ID is a triple such as (q, w, γ) , where q is a state , w is a string of input symbols and
γ is a string of stack symbols.

If $M =(Q, \Sigma, Г, \delta, q0, Z0, F)$ is a PDA we say that
$(q,aw,Z\alpha) \mid\!----- (p, , \beta\alpha)$ if $\delta(q,a,Z)$ contains (p, β).

M 'a' may be Є or an input symbol. Example: (q1, BG) is in $\delta(q1, 0,)$ ells that (q1, 011, GGR)|---- (q1, 11,BGGR).

25. .What is the significance of PDA?

Finite Automata is used to model regular expression and cannot be used to represent non regular languages. Thus to model a context free language, a Pushdown Automata is used.

26. When is a string accepted by a PDA?

The input string is accepted by the PDA if:

- The final state is reached .
- The stack is empty.

27. . Give examples of languages handled by PDA.

(1) L={ $a^n b^n$ | n>=0 },here n is unbounded , hence counting cannot be done by finite memory. So we require a PDA ,a machine that can count without limit.

(2) L= { wwR | w Є {a,b}* } , to handle this language we need unlimited counting capability .

28. Is NPDA (Nondeterministic PDA) and DPDA (Deterministic PDA)equivalent?

The languages accepted by NPDA and DPDA are not equivalent. For example: wwR is accepted by NPDA and not by any DPDA.

29. State the equivalence of acceptance by final state and empty stack.

- If L = L(M2) for some PDA M2 , then L = N(M1) for some PDA M1.
- If L = N(M1) for some PDA M1 ,then L = L(M2) for some PDA M2

where L(M) = language accepted by PDA by reaching a final state.
N(M) = language accepted by PDA by empty stack.

UNIT IV

1. What is a formal language?

   Language is a set of valid strings from some alphabet. The set may be empty,finite or infinite. L(M) is the language defined by machine M and L(G) is the language defined by Context free grammar. The two notations for specifying formal languages are: Grammar or regular expression Generative approach) Automaton(Recognition approach)

2. .What is Backus-Naur Form(BNF)?

Computer scientists describes the programming languages by a notation called Backus- Naur Form. This is a context free grammar notation with minor changes in format and some shorthand.

3. Let G= ({S,C} ,{a,b},P,S) where P consists of S->aCa , C->aCa |b. FindL(G).

   S-> aCa => aba
   S->aCa=> a aCa a=>aabaa
   S->aCa=> a aCa a=> a a aCa a a =>aaabaaa
   Thus L(G)= { $a^n b a^n$,where n>=1 }

4. Find L(G) where G= ({S} ,{0,1}, {S->0S1 ,S->_ },S) S->_ , _ is in L(G)

   S-> 0S1 =>0_1=>01
   S->0S1=>0 0S11=>0011
   Thus L(G)= { 0n1n | n>=0}

5. What is a parser?

A parser for grammar G is a program that takes as input a string w and produces as output either a parse tree for w ,if w is a sentence of G or an error message indicating that w is not a sentence of G.

6. What are the closure properties of CFL?

CFL are closed under union, concatenation and Kleene closure. CFL are closed under substitution , homomorphism. CFL are not closed under intersection , complementation. Closure properties of CFL's are used to prove that certain languages are not context free.

7. State the pumping lemma for CFLs.

Let L be any CFL. Then there is a constant n, depending only on L, such that if z is in L and |z| >=n, then z=uvwxy such that :
(i) |vx| >=1
(ii) |vwx| <=n and
(iii) for all i>=0 uviwxiy is in L.

8. What is the main application of pumping lemma in CFLs?

The pumping lemma can be used to prove a variety of languages are not context free . Some examples are:
L1 ={ aibici | i>=1} is not a CFL.
L2= { aibjcidj | i>=1 and J>=1 } is not a CFL.

9. What is Ogden's lemma?

Let L be a CFL. Then there is a constant n such that if z is any word in L, and we mark any n or more positions of z " distinguished" then we can write z=uvwxy suchthat:
(1) v and x together have atleast one distinguished position.
(2) vwx has at most n distinguished positions and
(3) for all i>=0 uviwxiy is in L.

10. Give an example of Deterministic CFL.

The language L={anbn : n>=0} is a deterministic CFL

11. What are the properties of CFL?

Let G=(V,T,P,S) be a CFG
- The fanout of G , _(G) is largest number of symbols on the RHS of any rule in R.
- The height of the parse tree is the length of the longest path from the root to some leaf.

12. What is a turing machine?

Turing machine is a simple mathematical model of a computer. TM has unlimited and unrestricted memory and is a much more accurate model of a general purpose computer. The turing machine is a FA with a R/W Head. It has an infinite tape divided into cells ,each cell holding one symbol.

13. What are the special features of TM?

In one move ,TM depending upon the symbol scanned by the tape head and state of the finite control:

- Changes state.
- Prints a symbol on the tape cell scanned, replacing what was written there.
- Moves the R/w head left or right one cell.

14. Define Turing machine.

A Turing machine is denoted as M=(Q,Σ,Ґ ,δ ,q0, B,F)

Q s a finite set of states.

Σ is set of i/p symbols ,not including B.

Ґ is the finite set of tape symbols.

q0 in Q is called start state.

B in Ґ is blank symbol.

F is the set of final states.

Δ is a mapping from Q X Ґ to Q X Ґ X {L,R}.

15. Define Instantaneous description of TM.

The ID of a TM M is denoted as α1q α2 . Here q is the current state of M s in Q; α1 α2 is the string in Ґ * that is the contents of the tape up to the rightmost nonblank symbol or the symbol to the left of the head, whichever is the rightmost.

16. What are the applications of TM?

TM can be used as:

- Recognizers of languages.
- Computers of functions on non negative integers.
- Generating devices.

17. What is the basic difference between 2-way FA and TM?

Turing machine can change symbols on its tape , whereas the FA cannot change symbols on tape. Also TM has a tape head that moves both left and right side ,whereas the FA doesn't have such a tape head.

18. What is (a)total recursive function and (b)partial recursive function

If f(i1,i2,………ik) is defined for all i1,…..ik then we say f is a total recursive function. They are similar to recursive languages as they are computed by TM that always halt.

A function f(i1,…ik) computed by a Turing machine is called a partial recursive function. They are similar to r.e languages as they are computed by TM that may or may not halt on a given input.

19. Define a move in TM.

Let X1 X2…X i-1 q Xi…Xn be an ID.

The left move is: if _ (q, Xi)= (p, Y,L) ,if i>1 then X1 X2…X i-1 q Xi…Xn |---- X1X2… X i-2 p X i-1 Y X i+1…Xn. M The right move is if _ (q, Xi)= (p, Y,R) ,if i>1 then X1 X2…X i-1 q Xi…Xn |---- X1X2… X i-1Y p X i+1…Xn. M

20. What is the language accepted by TM?

The language accepted by M is L(M) , is the set of words in _ * that cause M to enter a final state when placed ,justified at the left on the tape of M, with M at qo and the tape head of M at the leftmost cell. The language accepted by M is: { w | w in _ * and q0w |--- _1 p _2 for some p in F and _1 ,_2 in _ * }.

21. Give examples of total recursive functions.

All common arithmetic functions on integers such as multiplication , n!, [log2n] and 22n are total recursive functions.

UNIT V

1. What are(a) recursively enumerable languages (b) recursive sets?

The languages that is accepted by TM is said to be recursively enumerable (r. e) languages. Enumerable means that the strings in the language can be enumerated by the TM. The class of r. e languages include CFL's.

The recursive sets include languages accepted by at least one TM that halts on all inputs.

2. What are the various representation of TM?

We can describe TM using:

- Instantaneous description.
- Transition table.
- Transition diagram.

3. What are the possibilities of a TM when processing an input string?

- TM can accept the string by entering accepting state.
- It can reject the string by entering non-accepting state.
- It can enter an infinite loop so that it never halts.

4. What are the techniques for Turing machine construction?

- Storage in finite control.
- Multiple tracks.
- Checking off symbols.
- Shifting over
- Subroutines.

5. What is the storage in FC?

The finite control(FC) stores a limited amount of information. The state of the Finite control represents the state and the second element represent a symbol scanned.

6.When is checking off symbols used in TM?

Checking off symbols is useful method when a TM recognizes a language with repeated strings and also to compare the length of substrings.
(eg) : { ww | w _ _ * } or $\{a^i b^i \mid i>=1\}$. This is implemented by using an extra track on the tape with symbols Blank or √.

7.When is shifting over Used ?

A Turing machine can make space on its tape by shifting all nonblank symbols a finite number of cells to the right. The tape head moves to the right , repeatedly storing the symbols in the FC and replacing the symbols read from the cells to the left. The TM can then return to the vacated cells and prints symbols.

8. What is a multihead TM?

A k-head TM has some k heads. The heads are numbered 1 through k, and move of the TM depends on the state and on the symbol scanned by each head. In one move, the heads may each move independently left or right or remain stationary.

9.What is a 2-way infinite tape TM?

In 2-way infinite tape TM, the tape is infinite in both directions. The leftmost square is not distinguished. Any computation that can be done by 2-way infinite tape can also be done by standard TM.

10. How can a TM used as a transducer?

A TM can be used as a transducer. The most obvious way to do this is to treat the entire nonblank portion of the initial tape as input , and to treat the entire blank portion of the tape when the machine halts as output. Or a TM defines a function y=f(x) for strings x ,y _ _* if: q0X | --- qfY, where qf is the final state.

11. What is a multi-tape Turing machine?

. A multi-tape Turing machine consists of a finite control with k-tape heads and ktapes ; each tape is infinite in both directions. On a single move depending on the state of finite control and symbol scanned by each of tape heads ,the machine can change state print a new symbol on each cells scanned by tape head, move each of its tape head independently one cell to the left or right or remain stationary.

12. .What is a multidimensional TM?

The device has a finite control , but the tape consists of a k-dimensional array of cells infinite in all 2k directions, for some fixed k. Depending on the state and symbol scanned , the device changes state , prints a new symbol and moves its tapehead in one of the 2k directions, either positively or negatively ,along one of the k-axes.

13. When a recursively enumerable language is said to be recursive ? Is it true that the language accepted by a non-deterministic Turing machine is different from recursively enumerable language?

A language L is recursively enumerable if there is a TM that accepts L and recursive if there is a TM that recognizes L. Thus r.e language is Turing acceptable and recursive language is Turing decidable languages. No , the language accepted by non-deterministic Turing machine is same as recursively enumerable language.

13. What is Church's Hypothesis?

The notion of computable function can be identified with the class of partial recursive functions is known as Church-hypothesis or Church-Turing thesis. The Turing machine is equivalent in computing power to the digital computer.

14. When we say a problem is decidable? Give an example of undecidable problem?

A problem whose language is recursive is said to be decidable. Otherwise the problem is said to be undecidable. Decidable problems have an algorithm that takes as input an instance of the problem and determines whether the answer to that instance is "yes" or "no".

(eg) of undecidable problems are

(1)Halting problem of the TM.

15. Give examples of decidable problems.

1. Given a DFSM M and string w, does M accept w?
2. Given a DFSM M is L(M) = _ ?
3. Given two DFSMs M1 and M2 is L(M1)= L(M2) ?
4. Given a regular expression _ and a string w ,does _ generate w?
5. Given a NFSM M and string w ,does M accept w?

16. Give examples of recursive languages?

i. The language L defined as L= { "M" ,"w" : M is a DFSM that accepts w} is recursive.

ii. L defined as { "M1" U "M2" : DFSMs M1 and M2 and L(M1) =L(M2) } is recursive.

17. What are UTMs or Universal Turing machines?

Universal TMs are TMs that can be programmed to solve any problem, that can be solved by any Turing machine. A specific Universal Turing machine U is:

Input to U: The encoding "M " of a Tm M and encoding "w" of a string w.

Behavior : U halts on input "M" "w" if and only if M halts on input w.

18. What is the crucial assumptions for encoding a TM?

There are no transitions from any of the halt states of any given TM . Apart from the halt state , a given TM is total.

19. What properties of recursive enumerable seta are not decidable?

- o Emptiness
- o Finiteness
- o Regularity
- o Context-freedom.

20. Define L .When is a trivial property?

L is defined as the set { <M> | L(M) is in . } is a trivial property if is empty or it consists of all r.e languages.

21.What is a universal language Lu?

The universal language consists of a set of binary strings in the form of pairs (M,w) where M is TM encoded in binary and w is the binary input string.
Lu = { < M,w> | M accepts w }.

22.What is a Diagonalization language Ld?

The diagonalization language consists of all strings w such that the TM M whose code is w doesnot accept when w is given as input.

23. What properties of r.e sets are recursively enumerable?

- o $L \neq \Phi$
- o L contains at least 10 members.
- o w is in L for some fixed w.
- o $L \cap Lu \neq \Phi$

24. What properties of r.e sets are not r.e?

- $L = \Phi$
- $L = \Sigma$ *.
- L is recursive
- L is not recursive.
- L is singleton.
- L is a regular set.
- $L - Lu \neq \Phi$

25. What are the conditions for L to be r.e?

L is recursively enumerable iff satisfies the following properties:

i. If L is in and L is a subset of L_ ,then L_ is in (containment property)
ii. If L is an infinite language in ,then there is a finite subset of L in .
iii. The set of finite languages in is enumaerable.

26. What is canonical ordering?

Let Σ * be an input set. The canonical order for Σ *as follows . List words in order of size, with words of the same size in numerical order. That is let _ ={

x0,x1,…x t-1 } and xi is the digit i in base t.
(e.g) If _ ={ a,b } the canonical order is Σ * , a ,b , aa, ab ,……..

27. How can a TM acts as a generating device?

In a multi-tape TM ,one tape acts as an output tape, on which a symbol, once written can never be changed and whose tape head never moves left. On that output tape , M writes strings over some alphabet _ , separated by a marker symbol # , G(M) (where G(M) is the set w in Σ * * such that w is finally printed between a pair of #'s on the output device).

28. What are the different types of grammars/languages?

- Unrestricted or Phase structure grammar.(Type 0 grammar).(for TMs)
- Context sensitive grammar or context dependent grammar (Type1)(for Linear Bounded Automata)
- Context free grammar (Type 2) (for PDA)
- Regular grammar (Type 3) (for Finite Automata).

This hierarchy is called as Chomsky Hierarchy.

29. Show that AMBIGUITY problem is un-decidable.

Consider the ambiguity problem for CFGs. Use the "yes-no" version of AMB. An algorithm for FIND is used to solve AMB. FIND requires producing a word with two or more parses if one exists and answers "no" otherwise. By the reduction of AMB to FIND we conclude there is no algorithm for FIND and hence no algorithm for AMB.

30.State the halting problem of TMs.

The halting problem for TMs is: Given any TM M and an input string w, does M halt on w? This problem is undecidable as there is no algorithm to solve this problem.

31.Define PCP or Post Correspondence Problem.

An instance of PCP consists of two lists , A = w1,w2,….wk and B = x1,…..xk of strings over some alphabet _ .This instance of PCP has a solution if there is any sequence of integers i1,i2,..im with m >=1 such that wi1, wi2,…wim = xi1,xi2 ,…xim The sequence i1 ,i2 ,…im is a solution to this instance of PCP.

32.Define MPCP or Modified PCP.

The MPCP is : Given lists A and B of K strings from _ * ,say A = w1 ,w2, …wk and B= x1, x2,…..xk does there exists a sequence of integers i1,i2,…ir such that w1wi1wi2…..wir = x1xi1xi2…xir?

33 . What is the difference between PCP and MPCP?

The difference between MPCP and PCP is that in the MPCP ,a solution is required to start with the first string on each list.

34. What are the concepts used in UTMs?

- _ Stored program computers.

- Interpretive Implementation of Programming languages.
- Computability.

UNIT-I AUTOMATA

Part B

1. a) If L is accepted by an NFA with ε-transition then show that L is accepted by an NFA without ε-transition.

b) Construct a DFA equivalent to the NFA.

M=({p,q,r},{0,1}, δ,p,{q,s})

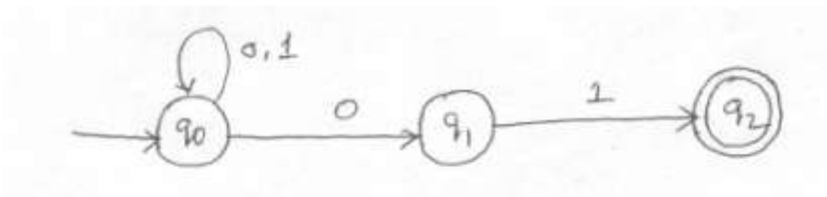
Where δis defined in the following table.

δ	0	1
p	{q,s}	{q}
q	{r}	{q,r}
r	{s}	{p}
s	-	{p}

2. a)Show that the set L={$a^n b^n$/n>=1} is not a regular. (6) b)Construct a DFA equivalent to the NFA given below: (10)

	0	1
p	{p,q}	P
q	r	R
r	s	-
s	s	S

3. a)Check whether the language L=($0^n 1^n$/n>=1) is regular or not? Justify your answer.
   b) Let L be a set accepted by a NFA then show that there exists aDFA that accepts L.
4. Define NFA with ε-transition. Prove that if L is accepted by an NFA with ε-transition then L is also accepted by a NFA without ε-transition.
5. a) Construct a NDFA accepting all string in $\{a,b\}^+$ with either two consecutive a's or two consecutive b's.
   b) Give the DFA accepting the following language:set of all strings beginning with a 1 that when interpretedas a binary integer is a multiple of 5.
6. Draw the NFA to accept the following languages.
   (i) Set of Strings over alphabet {0,1,…….9} such that the final digit has appeared before. (8)
   (ii)Set of strings of 0's and 1's such that there are two 0's separated by a number of positions that is a multiple of 4.
7. a)Let L be a set accepted by an NFA.Then prove that there exists a deterministic finite automaton that accepts L.Is the converse true? Justify your answer. (10)
   b)Construct DFA equivalent to the NFA given below: (6)



8.a) Prove that a language L is accepted by some ε–NFA if and only if L is accepted by some DFA. (8)

b) Consider the following ε–NFA.Compute the ε–closure of each state and find it's equivalent DFA. (8)

	ε	A	b	C
p	{q}	{p}	Φ	Φ
q	{r}	ϕ	{q}	Φ
*r	Φ	ϕ	ϕ	{r}

9.a) Prove that a language L is accepted by some DFA if L is accepted by some NFA.

b) Convert the following NFA to it's equivalent DFA

	0	1
p	{p,q}	{p}
q	{r}	{r}
r	{s}	ϕ
*s	{s}	{s}

10.a) Explain the construction of NFA with εtransition from any given regular expression.

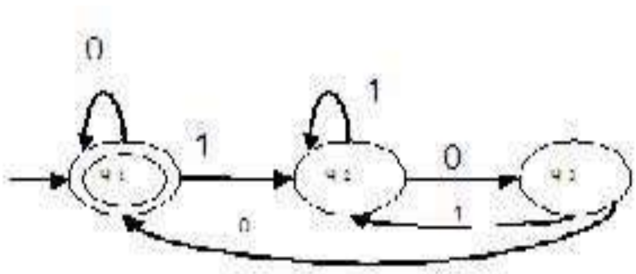
b) Let A=(Q,∑, δ, q0 ,{qf) be a DFA and suppose that for all a in ∑wehave δ(q0, a)= δ(qf ,a). Show that if x is a non empty string in L(A),then for all k>0,x^k is also in L(A).

UNIT-II REGULAR EXPRESSIONS AND LANGUAGES

PART-B

1.a)Construct an NFA equivalent to (0+1)*(00+11)

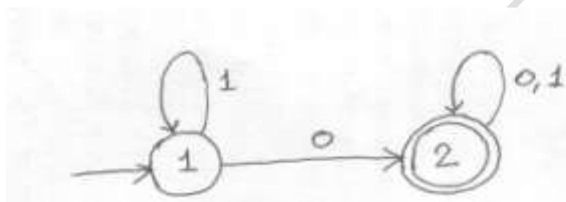
2.a)Construct a Regular expression corresponding to the state diagram given in the following figure.



b) Show that the set E=$\{0^i 1^i | i>=1\}$ is not Regular. (6)

3.a)Construct an NFA equivalent to the regular expression (0+1)*(00+11)(0+1)*.

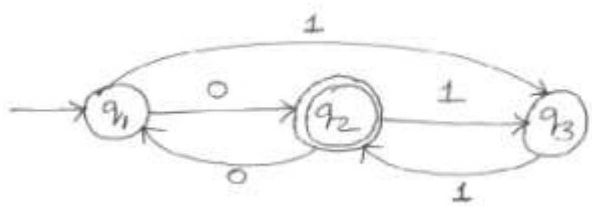
b)Obtain the regular expression that denotes the language accepted by the following DFA.

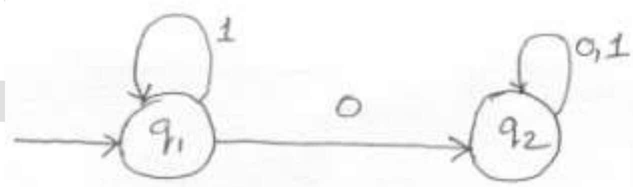


4.a)Construct an NFA equivalent to the regular expression ((0+1)(0+1)(0+1))*

b)Construct an NFA equivalent to 10+(0+11)0*1

5.a)Obtain the regular expression denoting the language accepted by the following DFA (8) b)Obtain the regular expression denoting the language accepted by the following DFA by using the formula R_{ij}^{k}





6. a)Show that every set accepted by a DFA is denoted by a regular Expression
   b)Construct an NFA equivalent to the following regular expression01*+1.

7. a)Define a Regular set using pumping lemma Show that the language L=$\{0^{i^2}$ / i is an integer,i>=1} is not regular
   b)Construct an NFA equivalent to the regular expression 10+(0+11)0*1

8. a) Show that the set L=$\{O^{n2/n}$ is an integer,n>=1} is not regular.
   b)Construct an NFA equivalent to the following regular expression $((10)+(0+1)^{*}$ 01. (10) 9.a)Prove that if L=L(A) for some DFA A,then there is a regular expression R such that L=L(R).

b) Show that the language $\{0^p, p \text{ is prime}\}$ is not regular.

10.Find whether the following languages are regular or not.

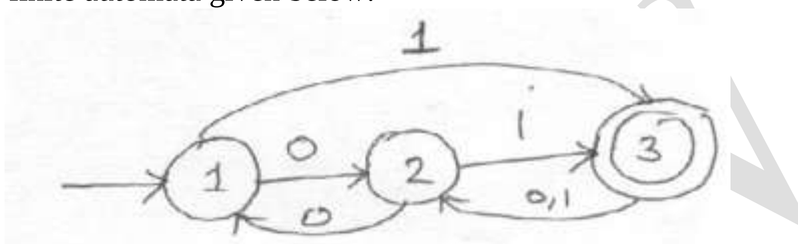
(i) $L=\{w \in \{a,b\} | w=w^R\}$.

(ii) $L=\{0^n 1^m 2^{n+m}, n,m>=1\}$

(iii) $L=\{1^k | k=n^2, n>=1\}$. (4)

(iv) $L1/L2=\{x \mid \text{for some } y \in L2, xy \in L1\}$,where L1 and L2 are any two languages and L1/L2 is the quotient of L1 and L2.

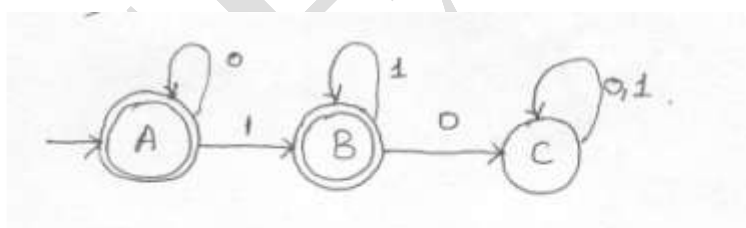
11.a) Find the regular expression for the set of all strings denoted by R^2_{13} from the deterministic finite automata given below:



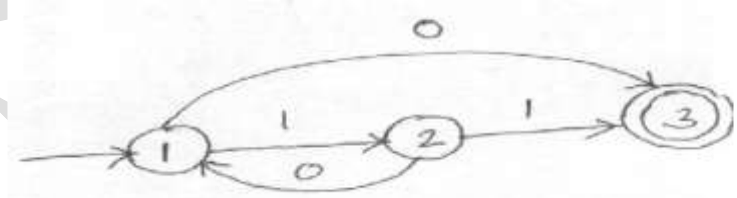
b)Verify whether the finite automata M1 and M2 given below are equivalent over {a,b}.

12.a)Construct transition diagram of a finite automaton corresponding to the regular expression $(ab+c^*)^*b$.

13.a)Find the regular expression corresponding to the finite automaton given below.



b)Find the regular expression for the set of all strings denoted by R^2_{23} from the deterministic finite automata given below.



14.a) Find whether the languages {ww,w is in (1+0)*} and $\{1^k \mid k=n^2, n>=1\}$ are regular or not.

b) Show that the regular languages are closed under intersectionand reversal.

UNIT-III CONTEXT FREE GRAMMARS AND LANGUAGES

PART-B

1. a) Let G be a CFG and let a=>w in G. Then show that there is a leftmost derivation of w.

   b) Let G=(V,T,P,S) be a Context free Grammar then prove that if S=> αthen there is a

   derivation tree in G with yield α.

2. Let G be a grammar s->OB/1A, A->O/OS/1AA, B->1/1S/OBB. For the string 00110101 find its leftmost derivation and derivation tree.

3) a) If G is the grammar S->Sbs/a, Show that G is ambiguous.

   b) Give a detailed description of ambiguity in Context free grammar

4. a) Show that E->E+E/E*E/(E)/id is ambiguous. (6) b)Construct a Context free grammar G which accepts N(M), where M=({q0, q1},{a,b},{z0,z},δ,q0,z0,Φ) and where δis given by

   δ(q0,b,z0)={(q0,zz0)}

   δ(q0, ε,z0)={(q0, ε)}

   δ(q0,b,z)={(q0,zz)}

   δ(q0,a,z)={(q1,z)}

   δ(q1,b,z)={(q1, ε)}

   δ(q1,a,z0)={(q0,z0)}

5. a)If L is Context free language then prove that there exists PDA M such that L=N(M).

   b)Explain different types of acceptance of a PDA.Are they equivalent in sense of language acceptance? Justify your answer.

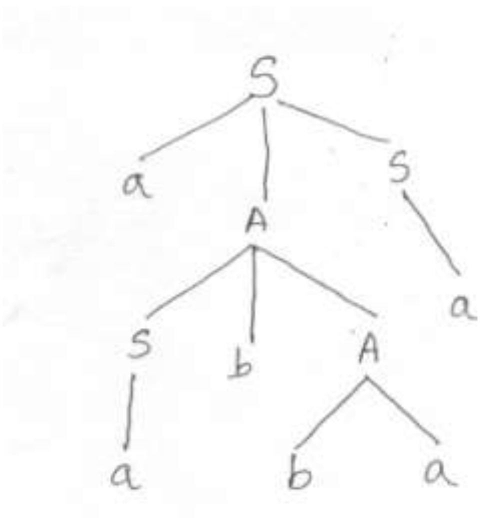
6. Construct a PDA accepting $\{a^n b^m a^n / m,n>=1\}$ by empty stack. Also construct the corresponding context-free grammar accepting the same set.

7. a) Prove that L is L(M2) for some PDA M2 if and only if L is N(M1) for some PDA M1.

   b)Define deterministic Push Down Automata DPDA. Is it true that DPDA and PDA are equivalent in the sense of language acceptance is concern? Justify Your answer.

8.a) Construct a equivalent grammar G in CNF for the grammar G1 where G1 =({S,A,B},{a,b},{S->bA/aB,A->bAA/aS/a, B->aBB/bS/b},S)

   b)Find the left most and right most derivation corresponding to the tree.



9. a) Find the language generated by a grammar
G=({S},{a,b},{S->aSb, S->ab},S) (4)

b) Given G=({S,A},{a,b},P,S) where P={S->AaS|S|SS,A->SbA|ba}
S-Start symbol. Find the left most and right most derivation of the string w=aabbaaa. Also construct the derivation tree for the string w.

c) Define a PDA. Give an Example for a language accepted byPDA by empty stack.

10. G denotes the context-free grammar defined by the following rules. ` S->ASB/ab/SS A->aA/A ,B->bB/A

(i)Give a left most derivation of aaabb in G. Draw the associated parse tree.

(ii)Give a right most derivation of aaabb in G. Draw the associated parse tree.

(iii) Show that G is ambiguous. Explain with steps.

(iv) Construct an unambiguous grammar equivalent to G. Explain.

11 a) Construct the grammar for the following PDA.
M=({q0, q1},{0,1},{X,z0},δ,q0,Z0,Φ) and where δis given by
δ(q0,0,z0)={(q0,XZ0)}, δ(q0,0,X)={(q0,XX)},δ(q0,1,X)={(q1, ε)}, δ(q1,1,X)={(q1, ε)},δ(q1, ε,X)={(q1, ε)}, δ(q1, ε, Z0)={(q1, ε)}. (12)

b) Prove that if L is N(M1) for some PDA M1 then L is L(M2) for somePDA M2.

12.a) Construct a PDA that recognizes the language
$\{a^i b^j c^k \mid i,j,k>0$ and $i=j$ or $i=k\}$.

b) Discuss about PDA acceptance
(1) From empty Stack to final state.
(2) From Final state to Empty Stack.

UNIT-IV PROPERTIES OF CONTEXTFREE LANGUAGES

PART-B

1.a)Find a grammar in Chomsky Normal form equivalent to S->aAD;A->aB/bAB; B->b, D->d. (6)

b)Convert to Greibach Normal Form the grammar G=({A1, A2, A3},{a,b},P,A1) where P consists of the following. A1 ->A2 A3, A2 ->A3 A1 /b,A3 ->A1 A2 /a. (10)

2.a)Show that the language $\{0^n 1^n 2^2 /n>=1\}$ is not a Context free language. (6)
b)Convert the grammar S->AB, A->BS/b, B->SA/a into Greibach NormalForm. (10)
3.a)Construct a equivalent grammar G in CNF for the grammar G1 where G1 =({S,A,B},{a,b},{S->bA/aB,A->bAA/aS/a, B->aBB/bS/b},S) (12)
b)Obtain the Chomsky Normal Form equivalent to the grammarS->bA/aB, A->bAA/aS/a, B->aBB/bS/b. (4)
4.a)Begin with the grammar
S->0A0/1B1/BBA->CB->S/AC->S/ ε
and simplify using the safe order Eliminate ε-Productions Eliminate unit production Eliminate useless symbols Put the (resultant) grammar in Chomsky Normal Form (10)
b)Let G=(V,T,P,S) be a CFG. Show that if S=α, then there is a derivation tree in a grammar G with yield α. (6)
5.a)Let G be the grammar S->aS/aSbS/ ε. Prove that L(G)={x/each prefix of x has atleast as many a's as b's} (6)
b)Explain the Construction of an equivalent grammar in CNF for thegrammar G=({S,A,B}{a,b},P,S)
where P={S->bA|aB, A->bAA|aS|a, B->aBB|bS|b} (10)
6.a)Find a Context free grammar with no useless symbol equivalent to
S->AB/CA, B->BC/ABA->a, C->aB/b. (6)
b)Show that any CFL without εcan be generated by an equivalent grammar in Chomsky Normal Form. (10)
7.a)Convert the following CFG to CNF S->ASA|aB A->B|S B->b| ε(12)
b)Explain about Greibach Normal Form. (4)
8.a)Is $L=\{a^n b^n c^n /n>=1\}$ a context free language? Justify Your answer. (8)
b)Prove that for every context free language L without εthere exists an equivalent grammar in Greibach Normal Form. (8)
9.State and Prove pumping lemma for Context free languages. (16)
10.a)State Pumping Lemma for context free language. Show that $(0^n 1^n 2^n /n>=1\}$ is not a Context free language. (6)
b)State Pumping lemma for context free language σshow that language $\{a^i b^j c^i d^j /i>=1$, and $j>=1\}$ is not context-free. (6)
11.a)Design a Turing Machine M to implement the function “multiplication” using the subroutine ‘copy’. (12)
b)Explain how a Turing Machine with the multiple tracks of the tape can be used to determine the given number is prime or not. (4)
12.a)Design a Turing Machine to compute f(m+n)=m+n, V m,n>=0 and simulate their action on the input 0100. (10)
b)Describe the following Turing machine and their working. Are they more powerful than the Basic Turing Machine? Multi-tape Turing Machine Multi-Dimensional Turing Machine
(3) Non-Deterministic Turing Machine. (6)
13.a) Define Turing machine for computing f(m,n)=m-n (proper subtraction).
(10)
b)Explain how the multiple tracks in a Turing Machine can be used for testing given

positive integer is a prime or not. (6)

14.a)Explain in detail:" The Turing Machine as a Computer of integer functions". (8)

b)Design a Turing Machine to accept the language L={$0^n 1^n$/n>=1} (8)

15.a)What is the role of checking off symbols in a Turing Machine? (4)

b)Construct a Turing Machine that recognizes the language
{wcw/w in {a+b}+ } (12)

16.Prove that the language L is recognized by a Turing Machine with a two way infinite tape if and only if it is recognized by a Turing Machine with a one way infinite tape. (16)

17.For each of the following Context free languages L, find the smallest pumping length that will satisfy the statement of the Context free pumping lemma. In each case, Your answer should include a number(the minimum pumping length), a detailed explanation of why that the number is indeed a valid pumping length for the given language L, and a detailed explanation of why no smaller number qualifies as a valid pumping length for that particular language L.

(i) L={$a^n b^n$|n>=0} (6)

(ii) L={w in {a,b}*|w has the same number of a's and b's} (6) (iii)L={w in {a,b}*| w has twice as many a's as b's.} (4)

18.Design a Turing Machine M that decides A={0^k/n>0 and k=2^n} the language consisting of all strings of 0's whose length is a power of 2. (16)

19.a)Give a High level implementation description with a neat sketch of a Turing Machine M that performs the following computation.M=on input w: writes a copy of w on the tape immediately after w,leaving the string w#w on the tape.Assume that the input string initially appears at the left most end of the tape and that the input alphabet does not contain the blank character' : The end of the input string is therefore determined by the location of the first blank cell on the input tape. The symbol # is assumed to be in the tape alphabet,and the input alphabet is {a,b}. (12)

b)Demonstrate the working of your TM with an example. (4)

20.a)Show that the language{$0^n 1^n 2^n$/n>=1}is not context free. (8)

b)Show that the context free languages are closed under union operation but not under intersection. (8)

UNIT-V UNDECIDABILITY

PART-B

1.a)Show that union of recursive languages is recursive. (4)

b)Define the language Ld and show that Ld is not recursively enumerable language. (8)

c)Explain the Halting problem. Is it decidable or undecidable problem

2.Define Universal language Lu.Show that Lu is recursively enumerable but not recursive.

3.a)Obtain the code for the TM M=({q1,q2,q3},{0,1},{0,1,B}, δ,q1,B,{q2}) With the moves δ(q1,1)=(q3,0,R) δ(q3,0)=(q1,1,R) δ(q3,1)=(q2,0,R) δ(q3,B)=(q3,1,L) δ(q3,B)=(q3,1,L)

b)Show that Ln is recursively enumerable.

4.a)Define Ld and show that Ld is not recursively enumerable. (12)
b)Whether the problem of determining given recursively enumerable language is empty or not? Is decidable? Justify your answer. (4)
5.Define the language Lu. Check whether Lu is recursively enumerable? or Lu is recursive? Justify your answer. (16) 6.a)Show that the language Ld is neither recursive nor recursively enumerable. (12) b)Describe how a Turing Machine can be encoded with 0 and 1 and give an example. (4) 7.a)Show that any non trivial property J of the recursively enumerable languages is undecidable. (8) b)Show that if L and L are recursively enumerable then L and L recursive.
8.Define the universal language and show that it is recursively enumerable but not recursive. (16)

9.Prove that the universal language Lu is recursively enumerable. (16)

10.State and Prove Rice's Theorem for recursive index sets. (16)

11.a)Show that the following language is not decidable. L={<M>| M is a TM that accepts the string aaab}. (8)
b)Discuss the properties of Recursive and Recursive enumerable languages. (8)
12.a)Define Post correspondence problem with an example. (8)
b)Prove that the function $f(n)=2^n$ does not grow at a polynomial rate, in other words, it does not satisfy $f(n)=O(n^p)$ for any finite exponent p.
13.a)Define the language Ld.Show that Ld is neither recursive nor recursively enumerable. (12)
b)Show that if a language L and its complement L are both recursively enumerable then L is recursive. (4) 14.a)What are the features of a Universal Turing Machine? (4)
b)Show that "If a language L and its compliment L are both recursively enumerable,then both languages are recursive". (6)
c)Show that halting problem of Turing Machine is undecidable. (6)
15.a)Does PCP with two lists x=(b,b ab^3,ba) and y=(b^3 ,ba , a)have a solution?. (6)
b)Show that the characteristic function of the set of all even numbers is recursive. (6) c)Let ∑={0,1}.Let A and B be the lists of three strings each,defined as List A List B
i Wi Xi1 1 1112 10111 10310 0
Does this PCP have a solution? (4)
16.a)Show that it is undecidable for arbitrary CFG's G1 and G2 whether L(G1)∩L(G2)Is

a CFL. (8)

b)Show that "finding whether the given CFG is ambiguous or not" is undecidable by reduction technique. (8)

17.Find whether the following languages are recursive or recursively enumerable.

(i) Union of two recursive languages. (4)

(ii) Union of two recursively enumerable languages. (4)

(iii) L if L and complement of L are recursively enumerable. (4)(iv)Lu (4)

18. Consider the Turing Machine M and w=01, where M=({q1,q2,q3},{0,1},{0,1,B},δ,q1,B,{q3}) and δis given by

Reduce the above problem to Post's correspondence Problem and find whether that PCP has a solution or not. (16)

18)Explain the Post's Correspondence Problem with an example (16)

19)Find the languages obtained from the following operations:

qi	δ(qi, 0)	δ(qi, 1)	δ(qi, B)
Q1	(q2,1,R)	(q2, 0,L)	(q2, 1,L)
q2	(q3, 0,L)	(q1, 0,R)	(q2, 0,R)
q3			

(i) Union of two recursive languages. (6)

(ii) Union of two recursively enumerable languages (6)

(iii) L if L and complement of L are recursively enumerable (4)