CMR ENGINEERING COLLEGE

UGC AUTONOMOUS

(Approved by AICTE - New Delhi. Affiliated to JNTUH and Accredited by NAAC & NBA)

Department of Computer Science & Engineering

FORMAL LANGUAGES AND AUTOMATA THEORY

Subject Code: CS

Step Material

Class: III Year I Semester

Branch: Computer Science & Engineering

Regulation: R20

A.Y.: 2023-24

2 MARKS QUESTIONS WITH ANSWERS & 16 MARK QUESTIONS

UNIT I AUTOMATA 2 MARKS QUESTION

AND ANSWERS

1. 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.

2.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,∑,6,q0,F), where Q is the finite set of states , ∑ is a finite input alphabet, q0
inQ is the initial state, F is the set of final states and 6 is the transition mapping function
Q * Σ 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.

3.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.

4. 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.

4.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).

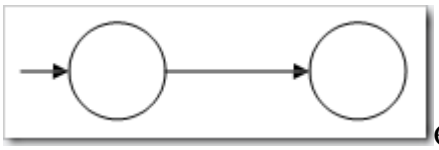
5. What is Є-closure of a state q0?

Є-closure(q0) denotes a set of all vertices p such that there is a path from q0 to p labeled Є.

Example :

Є

q0 q1



Є-closure(q0)={q0,q1}**6.What is a : (a) String (b) Regular language**

A string x is accepted by a Finite Automaton M=(Q,Σ,6.q0,F) if 6(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 | 6(q0,x) is in F}. A language is regular if it is accepted by some finite automaton.

UNIT II REGULAR EXPRESSIONS AND LANGUAGES 2 MARKS QUESTION AND ANSWERS

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 Σ be an alphabet. The regular expression over Σ and the sets they denote are:

i. Φ is a r.e and denotes empty set. ii. Є is a r.e and denotes the set {Є}

iii. For each 'a' in Σ , 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 Lii=0

example : 0* ={Є ,0,00,000,…..}

Language includes empty words also.

∞

L+ denotes Positive closure and is given by L+= U Li i=1 example:0+={0,00,000,…

.....

}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 Σ. 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 which accepts all strings with atleast two c's over the set Σ={c,b}

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

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

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

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

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

Any combinations of 0's and 1's.

(01)*={Є , 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,…..... }

8.**Reg exp denoting a language over Σ ={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)***9.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

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 Є –transitions, this NFA can be then simulated directly.

11.Reg exp for the language that accepts all strings in which 'a' appears tripled over the 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 uviw 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 atleastone ‘b’.

R=a+b+

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

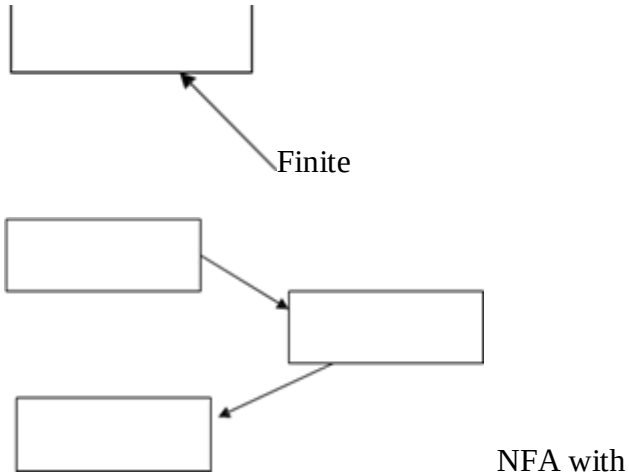
reg exp=(a+ab)*

16.What is the relationship between FA and regular expression.

Regular

Expression

Deterministic



NFA without

Є –moves

UNIT III CONTEXT FREE GRAMMAR AND LANGUAGES 2 MARKS QUESTION AND ANSWERS

1. **What are the applications of Context freeLanguages**

Cnotext free languages are used in

Defining programming languages.

Formalizing the notion of parsing. Translation of programming languages. 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 variablesand terminals respectively. V and T are disjoint. P is a finite set of productions each is of the form A->α where A is a variable and α 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 aG

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) whereP={S->aSb, S->ab}?

S=> aSb=>aaSbb=> ..... =>anbn

Thus the language L(G)={anbn | 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 andα and œ are any strings in (VUT)* then α A œ => αβœ.

G

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

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

(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 Є ,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

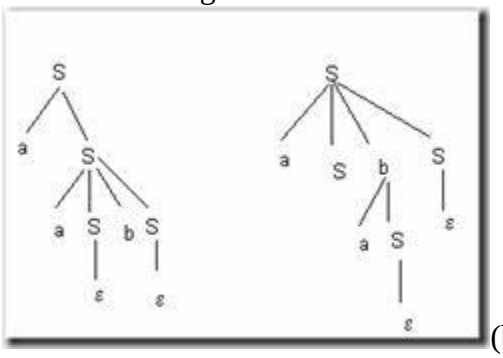
Thus L={anbmambn, where n,m>=1}

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.Consider the grammarP={S->aS | aSbS | Є } is ambiguous by constructing: (a) two parse trees (b) two leftmost derivation (c) rightmost derivation(a)

Consider a string aab :



(b) (i)S=>aS (ii) S=>aSbS

=>aaSbS =>aaSbS

=>aabS =>aabS

=>aab =>aab

(c)(i)S=>aS (ii) S=>aSbS

=>aaSbS =>aSb

=>aaSb =>aaSbS

=>aab =>aaSb

=>aab

11. 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

12. **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 pro duct

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 | aaA->b

B->b

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

A->b

B->b

13. **What are the three ways to simplify a context free grammar**?

By removing the useless symbols from the set of productions. By eliminating the empty productions.

By eliminating the unit productions.

14. 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

There are no productions of the form A->B where A and B are variables.

15. **Find the grammar for the language L={a2n bc ,where**

n>1 }let G=({S,A,B}, {a,b,c} ,P , {S}) where P: S->Abc

A->aaA | Є

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

1A->0A | 0 , B->1B | 1

The minimum string is S-> 0 |

1S->0S1=>001

S->0S1=>011

S->0S1=>00S11=>000S111=>0000A111=>00000111

Thus L={ 0n 1 m | m not equal to n, and n,m >=1}

17.Construct the grammar for the language L={ an b an |

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)

18. 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.

19. 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.

20. 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)

22. Let G= ({S,C} ,{a,b},P,S) where P consists of S->aCa , C->aCa |b. Find L(G).S-> aCa => aba

S->aCa=> a aCa a=>aabaa

S->aCa=> a aCa a=> a a aCa a a

=>aaabaaaThus L(G)= { anban ,where n>=1 }

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

S-> 0S1 =>0Є1=>01

S->0S1=>0 0S11=>0011

Thus L(G)= { 0n1n | n>=0}

24.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.

25.Define Pushdown Automata.

A pushdown Automata M is a system (Q, Σ, Ґ ,6 ,q0, Z0,F) where

Q is a finite set of states.

Σ is an alphabet called the input alphabet.

Γ is an alphabet called stack alphabet. q0 in Q is called initial state.

Zo in Γ is start symbol in stack. F is the set of final states.

6 is a mapping from Q X (Σ U {Є}) X Γ to finite subsets of Q X Γ *.

26.Compare NFA and PDA.

NFA	PDA
1.The language accepted by NFA is theregular language.	The language accepted by PDA isContext free language.
2.NFA has no memory.	PDA is essentially an NFA witha stack(memory).
3. It can store only limited amount of information.	It stores unbounded limit of information.
4.A language/string is accepted only by reaching the final state.	It accepts a language either by empty Stack or by reaching a final state.

27. Specify the two types of moves in PDA.

The move dependent on the input symbol(a) scanned is:

6(q,a,Z) = { (p1, œ1), (p2,œ2),….(pm,œm) }

where q qnd p are states , a is in Σ ,Z is a stack symbol and œi is in Γ*. PDA is in state q , with input symbol a and Z the top symbol on state enter state pi Replace symbol Z by string œi.

The move independent on input symbol is (Є-move):

6(q,Є,Z)= { (p1,œ1), (p2,œ2),… .... (pm,œm) }.

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 pi and replace Z by œi.

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

For a PDA M=(Q, Σ ,Γ ,6 ,q0 ,Z0 ,F) we define : Language accepted by final state L(M) as:

*

{ w | (q0 , w , Z0) |---(p, Є , œ) for some p in F and œ in Γ * }.

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

*

{ w | (q0,w ,Z0) |--- (p, Є, Є) for some p in Q}.

29. 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.

30. Define Deterministic PDA.

A PDA M =(Q, Σ ,Ґ ,6 ,q0 ,Z0 ,F) is deterministic if:

For each q in Q and Z in Ґ , whenever 6(q,Є,Z) is nonempty

,then6(q,a,Z) is empty for all a in Σ.

For no q in Q , Z in Ґ , and a in Σ U { Є} does 6(q,a,Z) contains more than one element.

(Eg): The PDA accepting {wcwR | w in (0+1) *

}.

32. 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.

33. 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.

34. Give examples of languages handled by PDA.

(1) L={ anbn | 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 .

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

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

36. 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 PROPERTIES OF CONTEXT FREE LANGUAGES AND PUSH DOWN AUOTOMATA 2 MARKSQUESTION AND ANSWERS

1. **State the equivalence of PDA and CFL**.

If L is a context free language, then there exists a PDA M such that L=N(M).

If L is N(M) for some PDA m, then L is a context free language.

2. 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.

3. 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.

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

The pumping lemma can be used to prove a variety of languages are not contextfree .

Some examples are:

L1 ={ aibici | i>=1} is not a CFL.

L2= { aibjcidj | i>=1 and J>=1 } is not a CFL.

5. Give an example of Deterministic CFL.

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

6. What are the properties of CFL?

Let G=(V,T,P,S) be a CFG

The fanout of G , Φ(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.

7. Compare NPDA and DPDA.

NPDA	DPDA
1. NPDA is the standard PDA used in automata theory.	1. The standard PDA in practical situation is DPDA.
2. Every PDA is NPDA unless otherwise specified.	2. The PDA is deterministic in the sense ,that at m move is possible from any ID.

8. What are the components of PDA ?

The PDA usually consists of four components: A control unit.

A Read Unit. An input tape.

A Memory unit.

9. What is the informal definition of PDA?

A PDA is a computational machine to recognize a Context free language. Computational power of PDA is between Finite automaton and Turing machines. The

PDA has a finite control , and the memory is organized as a stack.

10. Give an example of NonDeterministic CFL

The language L={ wwR : w Є {a,b} + } is a nondeterministic CFL.

11. 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.

12. 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.

13. Define Turing machine.

A Turing machine is denoted as M=(Q, Σ, Ґ ,6 ,q0, B,F) Q is 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.

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

14. Define Instantaneous description of TM.

The ID of a TM M is denoted as α1q α2 . Here q is the current state of M is 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.

15. What are the applications of TM?

TM can be used as:

Recognizers of languages.

Computers of functions on non negative integers. Generating devices.

16. 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.

17. What is the language accepted by TM?

The language accepted by M is L(M) , is the set of words in Σ * 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 Σ * and q0w |--- α1 p α2 for some p in F and α1 ,α2 in Γ * }.

18. What are the various representation of TM?

We can describe TM using: Instantaneous description.

Transition table. Transition diagram.

19. 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.

20. What are the techniques for Turing machine construction?

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

21. 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.

24.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.

25.Differentiate PDA and TM.

PDA	**TM**
1. PDA uses a stack for storage.	1. TM uses a tape that is infinite .
2.The language accepted by PDA is CFL.	2. Tm recognizes recursivelyenumerable languages.

26. What is a multi-tape Turing machine?

A multi-tape Turing machine consists of a finite control with k-tape heads and k- tapes ; 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.

UNIT V TURING MACHINE-2 MARKS QUESTION AND ANSWERS

1.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 tobe 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.

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

3. 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.

4. Differentiate recursive and recursively enumerable languages.

Recursive languages	Recursively enumerable languages
1. A language is said to be recursive if and only if there exists a membership algorithm for it.	1. A language is said to be r.e if there exists a TM that accepts it.
2. A language L is recursive iff there is a TM that decides L. (Turing decidable languages). TMs that decide languages are algorithms.	2. L is recursively enumerable iff there is a TM that semi-decides L. (Turing languages). TMs that semi-decides languages are not algorithms.

acc

5. 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 oninput "M" "w" if and only if M halts on input w.

6. **What properties of recursive enumerable seta are not decidable?**

Emptiness

Finiteness

Regularity

Context-freedom.

7. 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.

21.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.

22.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 Σ .This instance of PCP has asolution if there is any sequence of integers i1,i2,..im with m >=1 such thatwi1, wi2,…wim = xi1,xi2 ,…xim

The sequence i1 ,i2 ,…im is a solution to this instance of PCP.

23.Define MPCP or Modified PCP.

The MPCP is : Given lists A and B of K strings from Σ *

,sayA = w1 ,w2, …wk and B= x1, x2,…..xk

does there exists a sequence of integers i1,i2,…ir such that w1wi1wi2…..wir = x1xi1xi2…xir?

24 . 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.

25. **What are the concepts used in**

UTMs?

ANS : Stored program computers.

Interpretive Implementation of Programming languages.

Computability.

26.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 bythe 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.

27. 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.

Code No: R1622055 **R16** SET - 1

II B. Tech II Semester Supplementary Examinations, February - 2022
FORMAL LANGUAGES AND AUTOMATA THEORY
(Computer Science and Engineering)

Time: 3 hours Max. Marks: 70

Note: 1. Question Paper consists of two parts (**Part-A** and **Part-B**)
2. Answer **ALL** the question in **Part-A**
3. Answer any **FOUR** Questions from **Part-B**

~~~~~~

PART –A

1. a) Give an example for 2-way finite automata. (2M)
   b) Describe the operations on regular expressions. (2M)
   c) What is context sensitive language? Give example. (2M)
   d) How to construct push down automata equivalent to the given CFL (3M)
   e) With example discuss universal turing machine. (3M)
   f) What are the problems solved in polynomial time complexity? (2M)

PART –B

2. a) Describe the formal notation for NFA with epsilon closure and the uses of ϵ-closure. (7M)
   b) Design a Moore machine that determines the residue mod 4 for each binary string treated as integer. (7M)

3. a) Construct the regular expression for finite automata given transition function {δ(q1,1)→q1,δ(q1,0)→q2, δ(q2,0)→q2, δ(q2,1)→q1} q1 is start state and q2 is final state over string{0,1}. (7M)
   b) Write and explain the closure properties for regular sets. (7M)

4. a) Find a reduced grammar equivalent to the S→AB|CA, A→a, B→BC|AB,C→aB|b. (7M)
   b) Find a grammar equivalent to S→ AB/AC, A→ Aa/bAa/a, B→ bbA/aB/AB, C→aCa/aD D→ aD/bC with no useless symbols and put it into CNF. (7M)

5. a) **L = {a^n b^n | n >=1}**. Give the graphical representation for PDA obtained. Show the instantaneous description of the PDA on the input string ***aaaabbbb*** (7M)
   b) Prove that there exists an equivalent PDA for CFL with an example. (7M)

6. a) Design the Turing machine to recognize the language {$\mathbf{1^n2^n3^n|n>=1}$}. (7M)
   b) Explain with neat diagram the working of Turing machine and the types of Turing machines. (7M)

7. a) What is undecidability? Explain the usage of post correspondence theorem to solve undecidability problems. (7M)
   b) Write short notes on NP complete and NP hard problems. Explain with suitable examples. (7M)
~~~~~~

Code No: R1622055 **R16** SET - 1

II B. Tech II Semester Supplementary Examinations, August/September - 2021
FORMAL LANGUAGES AND AUTOMATA THEORY
(Computer Science and Engineering)

Time: 3 hours Max. Marks: 70

Note: 1. Question Paper consists of two parts (**Part-A** and **Part-B**)
2. Answer **ALL** the question in **Part-A**
3. Answer any **FOUR** Questions from **Part-B**

~~~~~~

PART –A

1.	a)	Write short notes on 5-tupple notation of finite automata.	(2M)
	b)	Write the regular expression for arithmetic expressions	(2M)
	c)	Construct a CFG generating all integers with sign	(3M)
	d)	Relate push down automata and instantaneous description languages	(2M)
	e)	Define Turing Machine and explain its model.	(2M)
	f)	Differentiate decidable and un decidable problems.	(3M)

PART –B

2.	a)	Explain the procedure for constructing minimum state DFA with an example.	(7M)
	b)	Design DFA which accepts language L={0,000,00000…….} over {0}.	(7M)
3.	a)	What is regular expression? Write the regular expression for the following languages over {0, 1}* (i)The set of all strings such that number of 0's is odd (ii) The set of all strings that contain exactly three 1's (iii) The set of all strings that do not contain 1101.	(7M)
	b)	Explain pumping lemma for regular languages with the applications of pumping lemma.	(7M)
4.	a)	is ambiguous grammar? Explain how to eliminate the ambiguity from the grammar? Consider the example grammar from E→E+E/E-E/E*E E→E/E E→(E)/id	(7M)
	b)	Eliminate unit productions and ε-production from the grammar S→Aa/B, B→A|bb, A→a|bc|B	(7M)
5.	a)	Design a non deterministic push down automata for the following languages **L1={$a^n b^n$|n>=0}, L2={ww^R |w in (0+1)*}**	(7M)
	b)	Construct the PDA for the given grammar S→AA|a, A→SA|b	(7M)
6.	a)	Design Turing machine over{a,b} which can compute concatenation function over Σ={1}	(7M)
	b)	Explain the following i)Language of Turing machine ii) Types of Turing machine.	(7M)
7.	a)	What is satisfiability problem? How Cook's theorem helps in deciding the NP completeness of problem.	(7M)
	b)	What is NP Problem? Explain with Travelling Sales person problem.	(7M)
~~~~~~

Code No: R1622055 **R16** SET - 1

II B. Tech II Semester Regular/Supplementary Examinations, November - 2020
FORMAL LANGUAGES AND AUTOMATA THEORY
(Computer Science and Engineering)

Time: 3 hours Max. Marks: 70

Note: 1. Question Paper consists of two parts (**Part-A** and **Part-B**)
2. Answer **ALL** the question in **Part-A**
3. Answer any **FOUR** Questions from **Part-B**

~~~~~~

PART –A

1.	a)	Why it is important to study Automata Theory for Computer science?	2M
	b)	Write the regular expression for the L={w €_{0,1}* ׀ w has no pair of consecutive zeros?	3M
	c)	Write the advantages of parse tree in identifying ambiguity.	2M
	d)	Write about the model of Push Down Automata.	3M
	e)	What is the name of the test that is used to evaluate whether a machine is intelligent human?	2M
	f)	Prove that integer linear programming is NP-Hard.	2M

PART -B

2.	a)	Describe the procedure of converting NFA to DFA with a suitable example..	7M
	b)	(0/1)*011 for this regular expression draw the NFA with ϵ-closures and convert it into NFA.	7M
3.	a)	Give a regular expression that generates the language L over the alphabet ∑={a, b} where each b in the string is followed by exactly one or three a's.	7M
	b)	Show that L={a^{2n}/n<0} is Regular.	7M
4.	a)	Define Context Free Grammar. State and Explain the closure properties of CFG.	7M
	b)	Discus various steps in signification of context free grammar. What is the need of such signification.	7M
5.	a)	Define Push Down Automata. Explain the basic structure of PDA with a neat graphical representation.	7M
	b)	Construct a PDA which accepts language of word over alphabet {a,b} canting {$a^ib^jc^k$/i,j,k €N,i+k=j}.	7M
6.	a)	Design a turing machines and its transition diagram to accept language greeted by {$a^ib^jc^k$/i,j,k €N,i+k=j}.	7M
	b)	Explain about types of Turing Machine warfare then.	7M
7.	a)	How to determine whether a problem is NP-Hard or P? Illustrate with an example.	7M
	b)	How can the Halting problem of Turing machine be Handled? Explain.	7M
~~~~~~

Code No: R1622055 **R16** SET - 1

II B. Tech II Semester Supplementary Examinations, December - 2022
FORMAL LANGUAGES AND AUTOMATA THEORY
(Computer Science and Engineering)

Time: 3 hours Max. Marks: 70

Note: 1. Question Paper consists of two parts (**Part-A** and **Part-B)**
2. Answer **ALL** the question in **Part-A**
3. Answer any **FOUR** Questions from **Part-B**

~~~~~~

PART –A (14M)

1.	a)	What is a state and write about few types of states?	2M
	b)	Write regular expressions for the following language over the alphabet $\sum = \{0, 1\}$ i) Strings with three consecutive 1's ii) Strings with three 1's	3M
	c)	Remove Null production from the following grammar S → ASA | aB | b A →B B → b | ∈	3M
	d)	Does push down automata have memory? Give explanation.	2M
	e)	What are the components of a Turing Machine?	2M
	f)	Why is the halting problem undecidable?	2M

PART –B (4X14M=56M)

2.	a)	What is Automata? Explain classification of Automata?	7M
	b)	Show with an example equivalence between NFA with and without Є- transitions.	7M
3.	a)	Prove that the following language L is not regular using pumping lemma $L = \{ a^{2n} b^{3n} a^{n} \mid n \geq 0\}$.	7M
	b)	Construct a NFA equivalent to the regular expression 10(0+11)0*1?	7M
4.	a)	Define Ambiguous Grammar? Check whether the grammar S→aAB, A→bC/cd, C→cd, B→c/d Is Ambiguous or not?	7M
	b)	Define Context Free Grammar. State and Explain the closure properties of CFG.	7M
5.	a)	S→aABB|aAA, A→aBB|a, B→bBB|A, construct the PDA that accepts the language generated by given grammar.	7M
	b)	What is deterministic Push Down Automata? Draw and explain a deterministic PDA for accepting $\{ 0^n1^n \mid n>1\}$	7M
6.	a)	Construct a Turing Machine for language $L = \{0^n1^n2^n \mid n \geq 1\}$.	7M
	b)	Explain the general structure of multi tape and non deterministic Turing machines and show that these are equivalent to basic Turing machines.	7M
7.	a)	Find whether post correspondence problem P={(10,101),(011,11),(101,011)} has match? Give the solution.	7M
	b)	Write the general working principle of post correspondence theorem. How it is modified? Explain.	7M
~~~~~~

Code No: R1622055 **R16** SET - 1

II B. Tech II Semester Model Examinations, March 2018
Formal Languages and Automata Thoery

Time: 3 hours Max. Marks: 70

Note: 1. Question Paper consists of two parts (**Part-A** and **Part-B)**
2. Answer **ALL** the question in **Part-A**
3. Answer any **THREE** Questions from **Part-B**

~~~~~~

PART –A

1. a) What is a state and write about few types of states? (4M)
   b) What is a string? Write about concatenation of two strings? (3M)
   c) Write the design strategy for NFA-ε ? (4M)
   d) Write about unreachable and dead states with illustration? (4M)
   e) Write about Leftmost derivation and rightmost derivation with example? (4M)
   f) Explain about offline Turing Machine? (3M)

PART -B

2. a) Explain the design of a finite state machine with an example? (10M)
   b) Explain the advantages of Finite State Machine? (6M)

3. a) What are Generative grammars? Write the components of such grammars? Explain with example the types of generative grammars? (8M)
   b) Show that the language $L=\{ww^R \mid w \in \{a,b\}^*\}$ is generated with context free grammar? (8M)

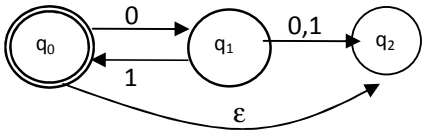
4. a) Write the Algorithm for minimizing DFA? (4M)
   b) Reduce the following DFA where q_1 is the start state and q_6 is the final state. (6M)

δ	0	1
q_1	q_2	q_3
q_2	q_4	q_5
q_3	q_6	q_7
q_4	q_4	q_5
q_5	q_6	q_7
q_6	q_4	q_5
q_7	q_6	q_7

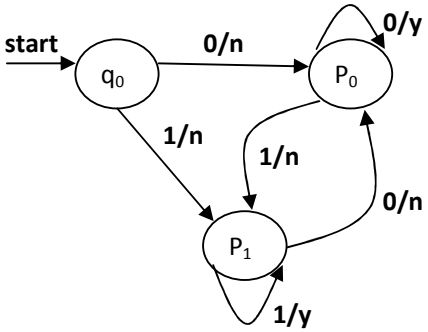
   c) Construct a regular expression corresponding to the DFA represented by the below transition table. q_1 is both the initial state and final state. (6M)

δ	0	1
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_1	q_2
~~~~~~

5. a) What is NFA? Explain the transitions of NFA? (4M)
   b) Construct an NFA that accepts the set of all strings over {0,1} that start with 0 or 1 and end with 10 or 01. (5M)
   c) Construct a DFA equivalent to the NFA given below (7M)



6. a) Convert the following Mealy machine to an equivalent Moore machine (8M)



   b) Explain different types of grammar with example? (8M)

7. a) Design a Turing Machine "Parity Counter" that outputs 0 or 1, depending on whether the number of 1's in the input sequence is even or odd respectively. (10M)
   b) What are P and NP class of Languages? What is NP Complete and give examples? (6M)

Code No: R1622055 **R16** SET - 2

II B. Tech II Semester Model Examinations, March 2018

Formal Languages and Automata Thoery

Time: 3 hours Max. Marks: 70

Note: 1. Question Paper consists of two parts (**Part-A** and **Part-B)**
2. Answer **ALL** the question in **Part-A**
3. Answer any **THREE** Questions from **Part-B**

~~~~~~

PART –A

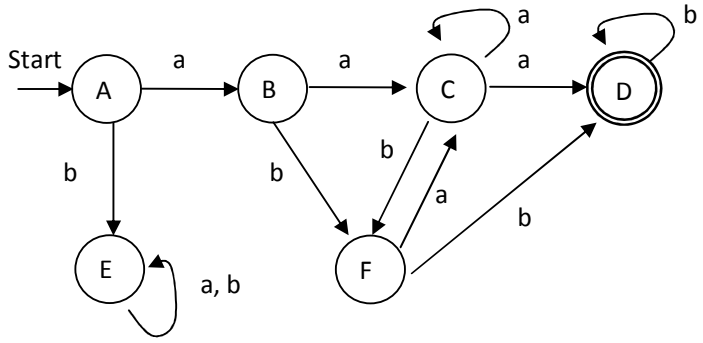
1. a) What is a transition? How are they represented? (4M)
   b) What is Kleene Closure and Positive Closure? (4M)
   c) What are the advantages of NFA over DFA? (3M)
   d) Differentiate DFA and 2DFA? (4M)
   e) Bring out the differences between Moore and Mealy machines? (4M)
   f) Explain about Multi Dimensional Turing Machine? (3M)

PART -B

2. a) Write about the Mathematical representation of Finite State Machine FSM? (8M)
   b) Explain the applications of Finite State Machine in real world? (8M)

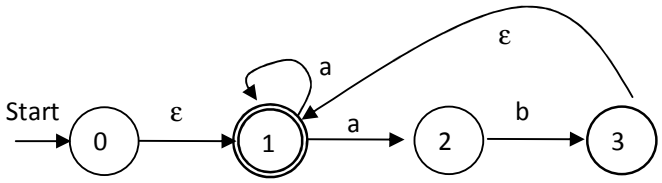
3. a) What is a context free Language? Give examples? Write about the properties of context free languages? (8M)
   b) Show that L={$a^{n!}$ | n>=0} can be generated with unrestricted grammar? (8M)

4. a) Reduce the DFA given below (6M)



   b) Construct an NFA with ε moves for 00^{*} +1 (6M)
   c) Write the steps to construct regular expression from given DFA? (4M)
~~~~~~

5. a) What is DFA? Explain the transitions of DFA? (4M)
   b) Construct a DFA accepting the language (5M)
   { W ∈ {a,b}* | W has neither aa nor bb as substring}
   c) Convert the following NFA-ε to NFA (7M)



6. a) Obtain a grammar to generate the language L = {$a^i b^j c^k$ | i+2j=k, i>=0, j>=0} (8M)
   b) Simplify the following CFG and Convert it into CNF (8M)
   S -> AaB | aaB
   A -> ε
   B -> bbA | ε

7. a) Design a Turing Machine "Parantheses Checker" that outputs 1 or 0 depending on whether the sequence is properly formed or not? (8M)
   b) What is Halting Problem of Turing Machine? Is it decidable or not? Explain? (8M)

Code No: R1622055 **R16** SET - 3

II B. Tech II Semester Model Examinations, March 2018

Formal Languages and Automata Thoery

Time: 3 hours Max. Marks: 70

Note: 1. Question Paper consists of two parts (**Part-A** and **Part-B)**
2. Answer **ALL** the question in **Part-A**
3. Answer any **THREE** Questions from **Part-B**

~~~~~~

PART –A

1. a) What is a state diagram? (3M)
   b) What is a formal language? Write the ways in which formal language can be specified? (4M)
   c) Write the design strategy for NFA? (4M)
   d) Write about indistinguishable and distinguishable states with illustration? (4M)
   e) Differentiate ambiguous and unambiguous grammar with example? (4M)
   f) Explain Church Turing Thesis? (3M)

PART –B

2. a) What is Automata? Explain classification of Automata? (8M)
   b) Write in detail about Models of Computation? (8M)

3. a) Write in detail the Chomsky hierarchy of formal languages? (8M)
   b) Show that the language L ={$a^n b^n c^n$ |n>=0} is not context free. (8M)

4. a) Construct a DLA accepting the language ; $\{w \in \{a,b\}^* = w\}$ has neither aa nor bb as subming (8M)
   b) Construct an NFA for r = $(a+bb)^* ba^*$ (8M)

5. a) Discuss the properties of Regular Expressions and Regular Languages. (8M)
   b) State and prove Arden's theorem. (8M)

6. a) Design a mealy machine to print out 1's complement of an input bit string? (8M)
   b) Write the general procedure to transform a grammar to Greibach Normal Form? (8M)

7. a) Design a Turing Machine to compute Max(n_1, n_2)? (8M)
   b) Explain about Universal Turing Machine? (8M)

|''|'''||''||'''||||
~~~~~~

Code No: R1622055 **R16** SET - 4

II B. Tech II Semester Model Examinations, March 2018

Formal Languages and Automata Thoery

Time: 3 hours Max. Marks: 70

Note: 1. Question Paper consists of two parts (**Part-A** and **Part-B)**
2. Answer **ALL** the question in **Part-A**
3. Answer any **THREE** Questions from **Part-B**

~~~~~~

PART –A

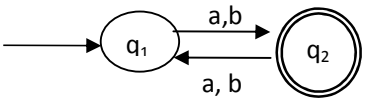
1. a) What is a state transition table? (3M)
   b) Consider a language L^* where L={ab, cd} with Σ = {a, b}. What is the shortest string in Σ^* that is not in the language L^*? (4M)
   c) Write the design strategy for DFA? (4M)
   d) Write the procedure to detect indistinguishable state? (4M)
   e) Write the general procedure to transform a grammar to Chomsky Normal Form? (4M)
   f) Explain about Multi Head Turing Machine? (3M)

PART -B

2. a) What are the components of Finite state Automata? Give examples of Finite state machine? (8M)
   b) Explain the disadvantages of Finite State Machine? (8M)

3. a) What are formal languages? Write about the different types of formal languages? (8M)
   b) Show that L = { a^p | p is prime} is generated with context sensitive grammar? (8M)

4. a) What is minimal DFA? Write the minimization Algorithm for DFA? (4M)
   b) Construct an NFA for the regular expression $(a+b)^*$ (aa+bb) $(a+b)^*$ (6M)
   c) Construct a regular expression for the given transition diagram (6M)



5. a) Construct a NFA equivalent to the regular expression (10+11)*00. (8M)
   b) Check wither the following time DFA's are equal or not (8M)

	0	1
q_1	q_1	q_2
q_2	q_3	q_1
q_3	q_2	q_3
	0	1
q_4	q_4	q_5
q_5	q_5	q_4
q_6	q_7	q_6
q_7	q_6	q_4

6. a) Design a Mealy machine to add two binary numbers of the form $x_1x_{2...}x_k$, $y_1y_{2...}y_k$? (8M)
   b) Prove that S -> aSbS | bSaS | ε is ambiguous. (8M)

7. a) Design a Turing Machine to accept the language L = {W W^R | W∈(a+b)*} (10M)
   b) Differentiate Turing Machines and Real Machines? (6M)

|''|'''||''||'''||||
~~~~~~