Unit 1

1.Define Finite Automation.

A finite automata is a collection of 5 tuples (Q, Σ. δ, q0, F)
where Q is a finite set of states, which is non empty.
Σ is a input alphabet, indicates input set.
δ is a transition function or a function defined for going to next state.
q0 is an initial state (q0 in Q)
F is a set of final states.

2. *Define Deterministic Finite Automation.*

- The finite automata is called DFA if there is only one path for a specific input from current state to next state.
- A finite automata is a collection of 5 tuples (Q, Σ. δ, q0, F)

where Q is a finite set of states, which is non empty.
Σ is a input alphabet, indicates input set.
δ is a transition function or a function defined for going to next state.
q0 is an initial state (q0 in Q)
F is a set of final states.

3. **Define Non-Deterministic Finite Automation.**

The finite automata is called NFA when there exists many paths for a specific input from current state to next state.
A finite automata is a collection of 5 tuples (Q, Σ. δ, q0, F)
where Q is a finite set of states, which is non empty.
Σ is a input alphabet, indicates input set.
δ is a transition function or a function defined for going to next state.
q0 is an initial state (q0 in Q)
F is a set of final states.

4. State regular expression.

Let Σ be an alphabet. The regular expressions over Σ and the sets that they denote are defined recursively as follows
a. Ø is a regular expression and denotes the empty set.
b. □ is a regular expression and denotes the set {□}
c. For each 'a'□ Σ, 'a' is a regular expression and denotes the set {a}.
d. If 'r' and 's' are regular expressions denoting the languages L1 and L2 respectively then
r + s is equivalent to L1 U L2 i.e. union
rs is equivalent to L1L2 i.e. concatenation
r* is equivalent to L1
* i.e. closure

5. **How the kleen's closure or closure of L can be denoted?**

$$L^* = \bigcup_{i=0}^{n} L_i$$ (e.g. a* ={□,a,aa,aaa,……})

6. **How do you represent positive closure of L**?

$$L^+ = \bigcup_{i=1}^{n} L_i$$ (e.g. a+ ={a,aa,aaa,……})

Unit 2

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.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*.

3. **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.

4.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

5.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,…….}

Unit 3

1.Define Pushdown Automata.

A pushdown Automata M is a system (Q, Σ, Γ ,δ ,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.

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

2.Compare NFA and PDA.

NFA	PDA
1.The language accepted by NFA is the regular language.	The language accepted by PDA is Context free language.
2.NFA has no memory.	PDA is essentially an NFA with a 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.

3. 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.

4. 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.

5. 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->α where A is a variable and α is a string of symbols from (V U T)*.

6. 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.

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 α A γ => αβγ.

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.

Unit 4

1.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.

2. 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 most one move is possible from any ID.

3.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.

4. 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 Γ * 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.

5.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 recursively enumerable languages.

6. 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.

7. Define Turing machine.

A Turing machine is denoted as M=(Q, Σ, Γ ,δ ,q0, B,F) Q is a finite set of states.

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

Γ is the finite set of tape symbols. q0 in Q is called start state.

B in Γ is blank symbol.

F is the set of final states.

δ is a mapping from Q X Γ to Q X Γ X {L,R}.

Unit 5

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 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.

2. 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 acceptable languages). TMs that semi-decides languages are not algorithms.

3.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.

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

5. **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 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.

6. **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.

7. **Define MPCP or Modified PCP.**

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