THEORY OF COMPUTATION – CS8501 (V SEMESTER)

UNIT I FINITE AUTOMATA

Introduction- Basic Mathematical Notation and techniques- Finite State systems – Basic Definitions –Finite Automaton – DFA & NDFA – Finite Automaton with €- moves – Regular Languages- Regular Expression – Equivalence of NFA and DFA – Equivalence of NDFA's with and without €-moves –Equivalence of finite Automaton and regular expressions –Minimization of DFA- - Pumping Lemma for Regular sets – Problems based on Pumping Lemma.

UNIT II GRAMMARS

Grammar Introduction– Types of Grammar - Context Free Grammars and Languages– Derivations and Languages – Ambiguity- Relationship between derivation and derivation trees – Simplification of CFG – Elimination of Useless symbols - Unit productions - Null productions – Greiback Normal form – Chomsky normal form – Problems related to CNF and GNF.

UNIT III PUSHDOWN AUTOMATA

Pushdown Automata- Definitions – Moves – Instantaneous descriptions – Deterministic pushdown automata – Equivalence of Pushdown automata and CFL - pumping lemma for CFL – problems based on pumping Lemma.

UNIT IV TURING MACHINES

Definitions of Turing machines – Models – Computable languages and functions –Techniques for Turing machine construction – Multi head and Multi tape Turing Machines - The Halting problem –Partial Solvability – Problems about Turing machine- Chomskian hierarchy of languages.

UNIT V UNSOLVABLE PROBLEMS AND COMPUTABLE FUNCTIONS

Unsolvable Problems and Computable Functions – Primitive recursive functions – Recursive and recursively enumerable languages – Universal Turing machine. MEASURING AND CLASSIFYING COMPLEXITY: Tractable and Intractable problems- Tractable and possibly intractable problems - P and NP completeness - Polynomial time reductions.

Total= 45 Periods

TEXT BOOKS:

1. Hopcroft J.E., Motwani R. and Ullman J.D, "Introduction to Automata Theory, Languages and Computations", Second Edition, Pearson Education, 2008. (UNIT 1,2,3)
2. John C Martin, "Introduction to Languages and the Theory of Computation", Third Edition, Tata McGraw Hill Publishing Company, New Delhi, 2007. (UNIT 4,5)

REFERENCES:

1. Mishra K L P and Chandrasekaran N, "Theory of Computer Science - Automata, Languages and Computation", Third Edition, Prentice Hall of India, 2004.
2. Harry R Lewis and Christos H Papadimitriou, "Elements of the Theory of Computation", Second Edition, Prentice Hall of India, Pearson Education, New Delhi, 2003.
3. Peter Linz, "An Introduction to Formal Language and Automata", Third Edition, Narosa Publishers, New Delhi, 2002.
4. Kamala Krithivasan and Rama. R, "Introduction to Formal Languages, Automata Theory and Computation", Pearson Education 2009

UNIT I

PART-A

1. **Define**
   - **(a) Finite Automata (FA)**
   - **(b) Transition Diagram** **NOV/DEC 2012**

   Finite Automata is a 5 tuples denoted by

   A = (Q, ∑, δ, q_0, F)

   where

   Q is a finite set of states
   ∑ is the finite set of input symbols
   δ is a transition function (Q X ∑ Q)
   q0 is the start state or initial state
   F is a set of final or accepting states

2. **State the Principle of induction.** **NOV/DEC 2012**

   Refer notes

3. **What is proof by contradiction?** **MAY/JUNE 2012**

   Refer notes

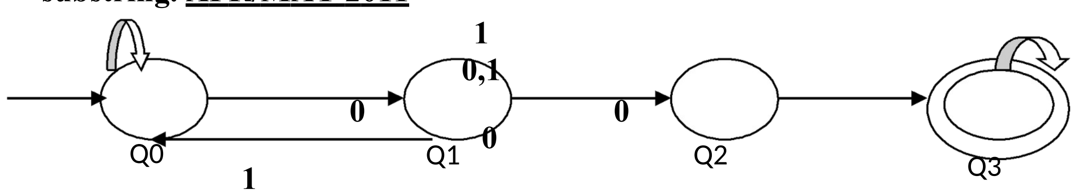
4. **Define ε-closure(q) with an example.** **MAY/JUNE 2012**

   Refer notes

5. **Differentiate between proof by contradiction and proof by contrapositive. APR/MAY 2011**

   If H then C will be proved by assuming ~H and then proving falsehood of falsehood of C. This is proof by contadiction. Proof by contrapositive is proved by assuming ~H and proving ~C.

6. **Construct a DFA for the language over {0, 1}* such that it contains "000" as a substring. APR/MAY 2011**



7. What is structural induction? **NOV/DEC 2011**

Let S(X) be a statement about the structures X that are defined by some particular recursive definition.

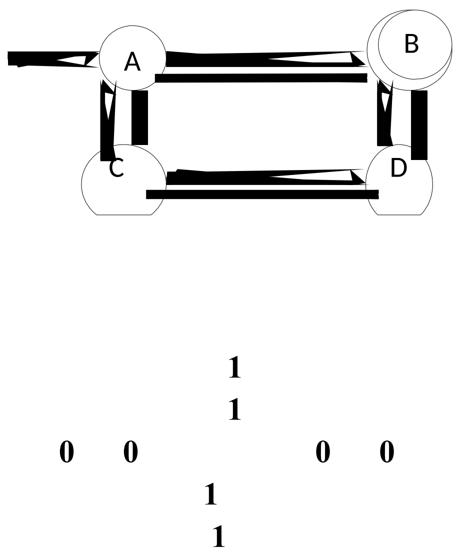
1. As a basis, Prove S(X) for the basis structure(s) X.
2. For inductive step, take a structure X that the recursive definition says is formed from Y_1, Y_2,....Y_k. Assume the statements $S(Y_1)$,...,$S(Y_k)$ and use these to prove S(X).

8. State the difference between NFA and DFA. **NOV/DEC 2011**

DFA must emit one and only one vertex/line/edge for each element of the alphabet. NFA do not have to obey this and can have multiple edges labeled with the same letter (repetition) and /or edges labeled with the empty string.

Both DFA and NFA recognize the same languages – the regular languages.

9. Construct deterministic finite automata to recognize odd number of 1's and even number of 0's? **APR/MAY 2010**



10. State the relations among regular expression, deterministic finite automata, non deterministic finite automaton and finite automaton with epsilon transition. **APR/MAY 2010**

Every Regular language defined by a regular expression ia also defined by the finite automata. If a Regular language 'L' is accepted by a NFA then there exists a DFA that accepts 'L'.

11. What is inductive proof? **NOV/DEC 2010**

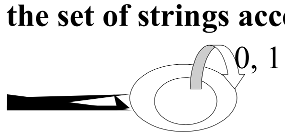
Statement P(n) follows from

(a) P(0) and

(b) P(n-1) implies P(n) for n>=1

Condition (a) is an inductive proof is the basis and Condition (b) is called the inductive step.

12. Find the set of strings accepted by the finite automata. **NOV/DEC 2010**



(0+1)* or L={ε, 0, 1, 00, 01, 10, 11,...........}

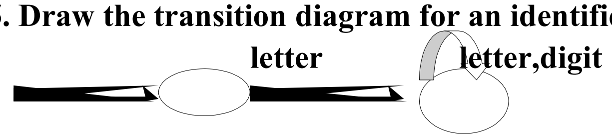
13. What is meant by DFA **MAY/JUNE 2013**

DFA—also known as **deterministic finite state machine**—is a finite state machine that accepts/rejects finite strings of symbols and only produces a unique computation (or run) of the automaton for each input string.

14. Define the term Epsilon transition **MAY/JUNE 2013**

In the automata theory, a **nondeterministic finite automaton with ε-moves (NFA-ε)**(also known as *NFA-λ*) is an extension of nondeterministic finite automaton(NFA), which allows a transformation to a new state without consuming any input symbols

15. Draw the transition diagram for an identifier **NOV/DEC 2013**



16. What is non deterministic finite automata? **NOV/DEC 2013**

In automata theory, a **nondeterministic finite automaton** (NFA), or nondeterministic finite state machine, is a finite state machine that ***(1)*** does not require input symbols for state transitions and ***(2)*** is capable of transitioning to zero or two or more states for a given start state and input symbol

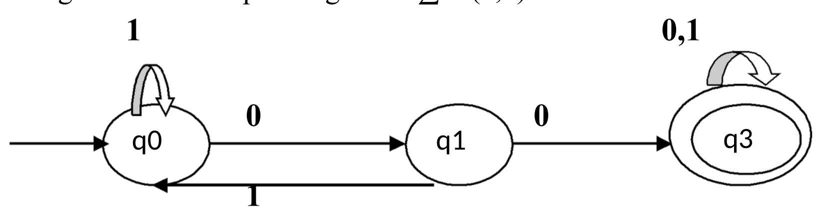
17. Define Deductive Proof. **NOV/DEC 2014**

A Deductive proof consists of a sequence of statements whose truth leads us from some initial statement, called the 'hypothesis' to a 'conclusion' statement.

"if H then C"

Ex: if $x >= 4$ then $2^x >= x^2$

18. Design DFA to accept strings over ∑ = (0,1) with two consecutive 0's. **NOV/DEC 2014**



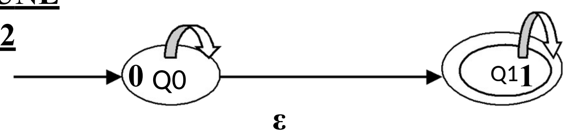
PART - B

1. **Explain the different forms of proof with examples. (8) NOV/DEC 2012**
2. **Prove that, if L is accepted by an NFA with ε-transitions, then L is accepted by an NFA without ε-transitions. (8) NOV/DEC 2012, NOV/DEC 2013**
3. **Prove that if n is a positive integer such that n mod 4 is 2 or 3 then n is not a perfect square. (6) NOV/DEC 2012**
4. **Construct a DFA that accepts the following**
   - **(i) L={ x € {a,b}:$|x|_a$ = odd and $|x|_b$ = even}. (10) NOV/DEC 2012**
   - **(ii) Binary strings such that the third symbol from the right end is 1. (10) MAY/JUNE 2012**
   - **(iii) All strings w over {0,1} such that the number of 1's in w is 3 mod 4. (8) NOV/DEC 2011**
   - **(iv) Set of all strings with three consecutive 0's.(10) NOV/DEC 2010**

5. **Prove by induction on n that $\sum_{i=0}^{n} i = n(n+1)/2$**

(6) MAY/JUNE 2012

6. **Construct an NFA without ε-transitions for the NFA give below. (8) MAY/JUNE 2012**



7. **Construct an NFA accepting binary strings with two consecutive 0's. (8) MAY/JUNE 2012**
8. **Show that a connected graph G with n vertices and n-1 edges (n>2) has at least one leaf. (6) APR/MAY 2011**

G has n vertices & (n-1) edges.

Therefore ∑ deg(V) = 2(n-1) which is impossible

Therefore deg(V) = 1 for at least one vertex and this vertex is a leaf.

9. **Convert the following ε-NFA to a DFA using the subset construction algorithm. (10)**

	ε	**a**	**b**	**c**
p	**Φ**	**{p}**	**{q}**	**{r}**
q	**{p}**	**{q}**	**{r}**	**Φ**
***r**	**{q}**	**{r}**	**Φ**	**{p}**

APR/MAY 2011

	a	b	c
[p]	[p]	[pq]	[pqr]
[pq]	[pq]	[pqr]	[pqr]
*[pqr]	[pqr]	[pqr]	[pqr]

10. **Prove that there exists a DFA for every ε-NFA. (8) APR/MAY 2011**

Refer Notes

11. **Show that the maximum edges in a graph (with no self-loops or parallel edges) is given by (n(n-1))/2 where 'n' is the number of nodes. (8) APR/MAY 2011**
12. **Prove that $\sqrt{2}$ is not rational. (8) NOV/DEC 2011**
13. **Describe the fundamental differences in the rules for forming DFA and NFA. Are these differences important in terms of the languages they can recognize? Give a reason for your answer. (8) APR/MAY 2010, NOV/DEC 2010**
14. **Construct an NFA for the following regular expression: (8) APR/MAY 2010**

(a+b)*a+b

15. **Consider the finite automata transition table shown below with F={q0}**

States **Inputs**

0 **1**

q0	q2	q1
q1	q3	q0
q2	q0	q3
q3	q1	q2

Find the language accepted by the finite automata. (10) NOV/DEC 2010

16. What is ε-closure (q)? Explain with an example. (6) NOV/DEC 2010

17. Construct DFA to accept the language
L={w | w is of even length and begins with 11 (10) MAY/JUNE 2013

18. (i) convert the following NFA to DFA

	a	b
p	{p}	{p,q}
q	{r}	{r}
r	{¢}	{¢}

(ii) Discuss on the relation between DFA and minimal DFA MAY/JUNE 2013

19. (i) Prove the equivalence of NFA and DFA using subset construction.
(ii) give DFA accepting the following language over the alphabet
Number of 1's is a multiples of 3
Number of 1's is not a multiples of 3 NOV/DEC 2013

20. Distinguish NFA and DFA with examples. MAY/JUNE 2013

21. (i) Prove that every tree has 'e' edges and 'e+1' nodes. NOV/DEC 2014
(ii) Prove that for every integer n>=0 the number $4^{2n+1} + 3^{n+2}$ is a multiple of 13. NOV/DEC 2014

22. (i) Let L be a set accepted by a NFA and then prove that there exists a DFA that accepts L. NOV/DEC 2014

(ii) Construct a DFA equivalent to the NFA M = ({a,b,c,d},{0,1},δ,a,{b,d}) where δ is a defined as

δ	0	1
a	{b,d}	{b}
b	c	{b,c
c	d	a
d	-	a

NOV/DEC 2014

UNIT – II

PART - A

1. **Give regular expressions for the following**

   L1=set of all strings of 0 and 1 ending in 00

   L2=set of all strings of 0 and 1 beginning with 0 and ending with 1. NOV/DEC 2012

   $RE1=(0+1)^{+}00$

   $RE2=0(0+1)^{+}1$

2. **Differentiate regular expression and regular language. NOV/DEC 2012**

   Refer notes

3. **Construct NFA for the regular expression a*b*. MAY/JUNE 2012**

   Refer notes

4. **Is regular set is closed under complementation?Justify. MAY/JUNE 2012**

   Closure under complement

   If L is a regular language, over alphabet ∑

   Complement of L=∑* - L

   Let L = L(A) for DFA A = (Q, ∑, δ, q0, F)

   Complement of L = L(B) where DFA B = (Q, ∑, δ, q0, Q-F)

   B is similar to A except accepting states of A have become non-accepting states of B and vice-versa.

   A string w is in L(B) iff δ' (q0,w) is in Q-F which occurs iff w is not in L(A).

5. **Prove that the complement of a regular language is also regular. APR/MAY 2011**

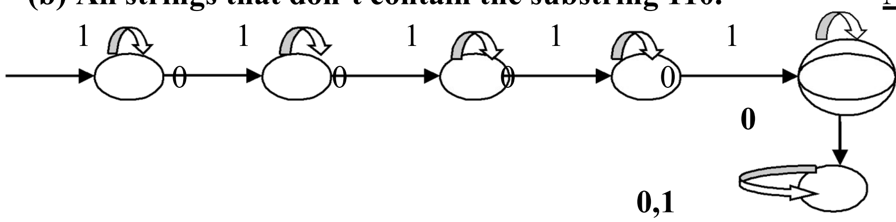
   Complement of L_1 is constructed from L_1 by reversing the states and the arrows in automata.

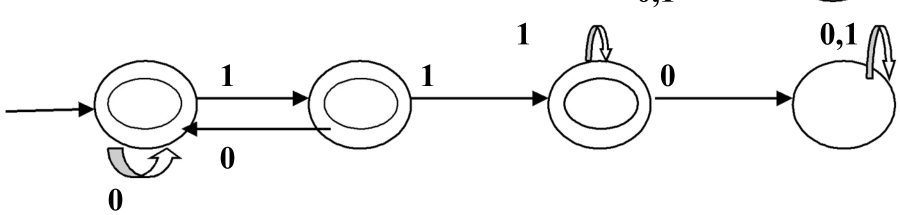
6. **Prove by pumping lemma, that the language 0^n1^n is not regular. APR/MAY 2011**

7. **Construct a DFA for the following:**

   (a) All strings that contain exactly 4 zeros.

   (b) All strings that don't contain the substring 110. NOV/DEC 2011





8. **Is the set of strings over the alphabet {0} of the form 0n where n is not a prime is regular? Prove or disprove. NOV/DEC 2011**

   To prove this language is not regular, examine the complement because the set of regular languages is closed under complement.

Assume that the set is regular. Let p be the pumping length of the language. Then, according to the pumping lemma, break the string $s=0^p$ into s=xyz where y has positive length.

Then, $s=xy^iz=0^{p+(i-1)|y|}$ must also be in the set for any i. In particular let i=p+1.Then $xy^{p+1}z=0^{p+p|y|}$ must be in the set so p+p|y| = p(1+|y|) must be prime.

Thus we have a contradiction and the set cannot be regular.

9. Let L = {w:w ε {0,1}* w does not contain 00 and is not empty}. Construct a regular expression that generates L. **APR/MAY 2010**

Regular Expression = (0+1)(1+10)*(101+1)*

10. Prove or disprove that the regular languages are closed under concatenation and complement. **APR/MAY 2010**

Closure under concatenation

Since L and M are regular languages,they have regular expressions

L=L(R) and M=L(S)

Then L.M = L(R.S), by definition of regular expression

Closure under complement

If L is a regular language, over alphabet ∑

Complement of L=∑* - L

Let L = L(A) for DFA A = (Q, ∑, δ, q0, F)

Complement of L = L(B) where DFA B = (Q, ∑, δ, q0, Q-F)

B is similar to A except accepting states of A have become non-accepting states of B and vice-versa.

A string w is in L(B) iff δ' (q0,w) is in Q-F which occurs iff w is not in L(A).

11. Give the regular expression for set of all strings ending in 00. **NOV/DEC 2010**

(0+1)*00

12. State pumping lemma for regular set. **NOV/DEC 2010, NOV/DEC 2013,NOV/DEC 2014**

Let L be a regular set. Then there is a constant n such that if Z is a string in L and |Z| >=n, Z can be written as Z=UVW such that |V|>=1 and |UV|<=n and for all i>=0 UV^iW is in L.

13. What is a regular expression ? **MAY/JUNE 2013**

A **regular expression** (abbreviated **regex** or **regexp**) is a sequence of characters that forms a search pattern, mainly for use in pattern matching with strings, or string matching, i.e. "find and replace"-like operations

14. Name any four closure properties of regular languages **MAY/JUNE 2013**

union, intersection, complement, difference

15. Construct NFA equivalent to the regular expression (0+1)01 **NOV/DEC 2013**

Refer notes

16. Prove or disprove that (r+s)* = r* + s* **NOV/DEC 2014**

Refer notes

PART – B

1. **Prove that there exists an NFA with ε-transitions that accepts the regular expression γ.** **(10) <u>MAY/JUNE 2012, NOV/DEC 2010</u>**

2. **Which of the following languages is regular? Justify.(Using Pumping Lemma)**

   (i) $L=\{a^nb^m \mid n,m>=1\}$

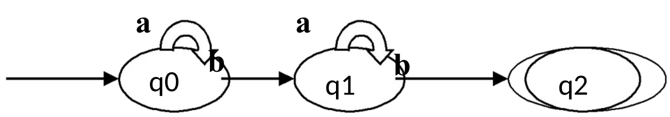
   (ii) $L=\{a^nb^n \mid n>=1\}$ **(8) <u>MAY/JUNE 2012</u>**

   (iii) $L=\{a^mb^n \mid m>n\}$ **(10) <u>NOV/DEC 2012</u>**

   (iv) $L=\{a^nb^n \mid n>=1\}$ **(6) <u>NOV/DEC 2010</u>**

   (v) $L=\{0^{n^2} \mid n \text{ is an integer}, n>=1\}$ **(6) <u>NOV/DEC 2014</u>**

3. **Obtain the regular expression for the finite automata.** **(8) <u>MAY/JUNE 2012</u>**



4. **Prove any two closure properties of regular languages.(8)<u>NOV/DEC 2012, NOV/DEC 2011, APRIL/MAY 2010</u>**

5. **Construct a minimized DFA from the regular expression**
   - **(i) (b/a)*baa** **(10) <u>MAY/JUNE 2012</u>**
   - **(ii) 0*(01)(0/111)*** **(16) <u>NOV/DEC 2012</u>**
   - **(iii) (x+y)x(x+y)*. Trace for a string w=xxyx.** **(16) <u>NOV/DEC 2011</u>**
   - **(iv) (a+b)(a+b)* and trace for a string baaaab.** **(16) <u>APR/MAY 2010</u>**
   - **(v) (b/a)*baa** **(16) <u>NOV/DEC 2010</u>**
   - **(vi) 10+(0+11)0*1** **(16) <u>NOV/DEC 2014</u>**

6. **Construct a regular expression for the following DFA using kleene's theorem. (10) <u>APR/MAY 2011</u>**

	0	1
*A	A	B
B	C	B
C	A	B

7. **Construct a ε-NFA for the following regular expression.** **(6) <u>APR/MAY 2011</u>**

   (i) **(0+1)*(00+11)(0+1)***

8. Construct minimized automata for the following automata to define the same language. (10) APR/MAY 2011

	a	b
q0	q1	q0 q1
	q0	q2
q2	q3	q1
*q3	q3	q0
q4	q3	q5
q5	q6	q4
q6	q5	q6
q7	q6	q3

9. Prove that "If two states are not distinguished by the table-filling algorithm then the states are equivalent. (6) APR/MAY 2011

10. State and explain the conversion of DFA into regular expression using Arden's theorem. Illustrate with an example. (16) NOV/DEC 2011

11. Define regular expression. Show that

(1+00*1)+(1+00*1)(0+10*1)*(0+10*1)=(0*1(0+10*1)* (8) NOV/DEC 2011

20. Consider the alphabet A = {a,b} and the language L = {bb,bab,baab,baaab,…..}

(i) Is A* finite or infinite? Give a brief reason for your answer.(2)
(ii) Write down a regular expression that represents the above language L.(4)
(iii) Write down a regular grammar which describes the above language L. (4)
(iv) Draw the DFA corresponding to the above language L. (6) APR/MAY 2010

21. Find an equalities for the following regular expression and prove for the same.(9) APR/MAY 2010

(i) b + ab* + aa*b + aa*ab*
(ii) a*(b + ab*)
(iii) a(a+b)* + aa(a+b)* + aaa(a+b)*

13. Prove that regular sets are closed under substitution. (6) NOV/DEC 2010

14. (i) Discuss on regular expressions
(ii) Discuss in detail about the closure properties of regular languages. MAY/JUNE 2013, NOV/DEC 2013

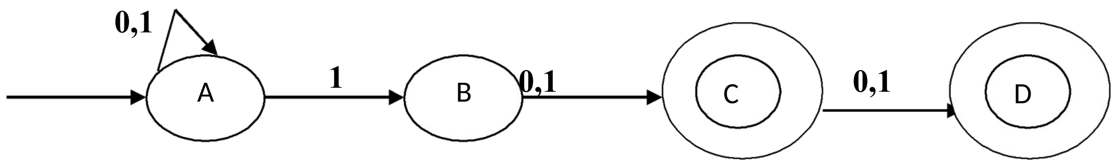
15. Prove that the following languages are not regular

(1) $\{0^{2n} \mid n>=1\}$

(2) $\{a^m b^n a^{m+n} \mid m>=1 \text{ and } n>=1\}$ **<u>MAY/JUNE 2013</u>, <u>NOV/DEC 2013</u>**

16. (i)Discuss the application of finite automata **<u>NOV/DEC 2013</u>**

17. Convert the following NFA into a regular expression **<u>NOV/DEC 2013</u>**



18. Explain the DFA minimization algorithm with an example. (10) **<u>NOV/DEC 2014</u>**

Refer Notes

<u>UNIT – III</u>

<u>PART - A</u>

1. **What is ambiguous grammar?** **<u>NOV/DEC 2012, MAY/JUNE 2013</u>**

   Refer notes

2. **What are the diiferent types of language accepted by a PDA and define them?** **<u>NOV/DEC 2012</u>**

   Accepted by null state

   Accepted by final state

3. **Specify the use of context free grammar.** **<u>MAY/JUNE 2012</u>**

   Refer notes

4. **Define parse tree with an example.** **<u>MAY/JUNE 2012</u>**

   Refer notes

5. **Construct a CFG over {a,b} generating a language consisting of equal number of a's and b's.** **<u>APR/MAY 2011</u>**

   S aSbS | bSaS | SS

6. **Is the language of Deterministic PDA and Non – deterministic PDA same?** **<u>APR/MAY 2011</u>**

   The language is not same. This language of NPDA is a superset of the language of DPDA.

7. **Is the grammar below ambiguous S SS | (S) | S(S)S | E?** **<u>NOV/DEC 2011</u>**

   It is ambiguous

The sentence such as E(E)E can have more than one LMD (or) RMD (or) Parse tree.

8. Convert the following grammar into an equivalent one with no unit productions and no useless symbols SABA AaAA|aBC|bB B A|bB|Cb CCC|Cc **NOV/DEC 2011**

Refer notes

9. Consider the following grammar G with productions **APR/MAY 2010**

S ABC | BaB A aA|BaC|aaa BbBb|a CCA|AC Give a CFG with no useless variables that generates the same language.

Symbol C is non-generative, after removing productions with C we have,

S BaB A

aA|aaa

BbBb|a

CFG with no usless variables

{S BaB A aA|aaa BbBb|a}

10. State the definition of Pushdown automata. **APR/MAY 2010**

A pushdown automaton consists of 7 tuples

P = (Q, ∑, Γ, δ, q0, Z0, F)

Where

Q – A finite non empty set of states

∑ - A finite set of input symbols

Γ – A finite non empty set of stack symbols

δ - The transition function is given by

δ:Q X (∑ U {ε}) X Γ Q X Γ*

q0 – q0 in Q is the start state

Z0 - Initial start symbol of the stack

F - F in Q ,set of accepting states or final states.

11. Write down the context free grammar for the language L = { $a^n b^n$ |n>=1} **NOV/DEC 2010, NOV/DEC 2013**

S aSb |ab

12. Is the grammar EE+E|id is ambigupus ? Justify. **NOV/DEC 2010**

It is ambiguous

The sentence such as id+id+id can have more than one LMD (or) RMD (or) Parse tree.

13. What is a CFG ? **MAY/JUNE 2013**

A **context-free grammar** (**CFG**) is a formal grammar in which every production rule is of the form *V* → *w* where *V* is a *single* nonterminal symbol, and *w* is a string of terminals and/or nonterminals (*w* can be empty). A formal grammar is considered "context free" when its production rules can be applied regardless of the context of a nonterminal. No matter which symbols surround it, the single nonterminal on the left hand side can always be replaced by the right hand side.

14. Compare NFA and PDA **NOV/DEC 2013**

Refer notes

15. Give the general forms of CNF. **NOV/DEC 2014**

ABC

Aa

16. Show that CFLs are closed under substitutions **NOV/DEC 2014**

Refer notes

PART – B

1. **Consider the following grammar for list structures:**

   Sa|^|(T)   TT,S|S

   Find left most derivation, rightmost derivation and parse tree for (((a,a),^(a)),a)(10) NOV/DEC 2012

2. **Construct the PDA accepting the language**

   (i) L={$(ab)^n$|n>=1} by empty stack. (6) NOV/DEC 2012

   (ii) L={$a^{2n}b^n$|n>=1} Trace your PDA for the input with n=3. (10) NOV/DEC 2012

   (iii)L={ww^R|w is in (a+b)*} (10) MAY/JUNE 2012

   (iv)L={0^n1^{2n}} by empty stack (8) APR/MAY 2011

   (v) L={ww^Rw|w is in {0+1}*} (10) NOV/DEC 2010

3. **Find the PDA equivalent to the given CFG with the following productions**

   (i) SA, ABC, Bba, Cac (6) NOV/DEC 2012

   (ii) SaSb|A, AbSa|S| ε (10) NOV/DEC 2011

4. **Is the following grammar is ambiguous? Justify your answer.**

   (i) E E+E |E*E | id (6) MAY/JUNE 2012

   (ii) E E+E|E*E|(E)|a (4) APRIL/MAY 2011

5. **Find the context free languages for the following grammars.**

   (i) SaSbS|bSaS| ε } **(10) MAY/JUNE 2012**

(ii) SaSb|ab

(iii)SaSb|aAb, AbAa, Aba **(6) NOV/DEC 2011**

6. **Discuss the equivalence between PDA and CFG. (6) MAY/JUNE 2012, MAY/JUNE 2013**
7. **Prove that if 'w' is a string of a language then there is a parse tree with yield 'w' and also prove that if Aw then it implies that 'w' is a string of the language L defined by a CFG. (6)APR/MAY 2011**
8. **Construct a CFG for the set $\{a^i b^j c^k | i \neq j \text{ or } j \neq k\}$ (6) APR/MAY 2011**
9. **Prove that if there exists a PDA that accepts by final state then there exists an equivalent PDA that accepts by Null state. (8) APRIL/MAY 2011**
10. **Is NPDA (Nondeterministic PDA) and DPDA (Deterministic PDA) equivalent? Illustrate with an example. (8) NOV/DEC 2011**
11. **What are the different types of language acceptances by a PDA and define them. Is it true that the language accepted by a PDA by these different types provides different languages? (8) NOV/DEC 2011**
12. **Consider the grammar**
    (i) **SiCtS** (ii)
    SiCtSeS (iii)
    Sa
    (iv) **Cb**

where i, t, and e stand for if, then, and else, and C and S for "conditional" and "statement" respectively.

(i) Construct a leftmost derivation for the sentence w=ibtibtaea.
(ii) Show the corresponding parse tree for the above sentence.
(iii)Is the above grammar ambiguous? If so, prove it.
(iv)Remove ambiguity if any and prove that both the grammar produces the same language. (16)APR/MAY2010

13. **Consider the GNF CFG G=({S,T,C,D},{a,b,c,d}, S, P}where P is:**

ScCD|dTC|€

CaTD|c

TcDC|cST|a

DdC|d

Present a PDA that accepts the language generated by this grammar. Your PDA must accept by empty store, it must start with S on its stack, and it must be based on the above grammar. (16) APR/MAY2010

14. Let G be the grammar

SaB|bA Aa|

aS|bAA Bb|

bS|aBB

for the string baaabbabba. Find leftmost derivation, rightmost derivation and parse tree. (9)NOV/DEC 2010,MAY/JUNE 2013

15. What is deterministic PDA? Explain with an example. (7) NOV/DEC 2010

16. Let L is a context free language. Prove that there exists a PDA that accepts L. (6) NOV/DEC 2010

17. Construct PDA for the language

18. L = {ww^R | W in (a+b)*)} MAY/JUNE 2013,NOV/DEC 2013

19. Convert the following grammar into GNF

S -> XY1/0

X -> 00X/Y

Y -> 1X1 NOV/DEC 2013

20. Show that the following grammars are ambiguous

{ S->aSbS/bSaS}

{S->AB/aaB, A->a/Aa, B->b} NOV/DEC 2013

21. (i) Write a grammar G to recognize all prefix expressions involving all binary arithmetic operators. Construct a parse tree for the sentence '-*+abc/de' using G? (6)

(ii) Show that the following grammar G is ambiguous S SbS |a (6)

(iii)Construct a context free grammar for {0^m1^n | 1<=m<=n} (4) NOV/DEC 2014

22. (i) If L is context free language prove that there exists a PDA M, such that L=N(M). (8)

(ii) Prove that if L is N(M1) (the language accepted by empty stack) for some PDA M1, then L is N(M2) (the language accepted by final state) for some PDA M2. (8) <u>NOV/DEC 2014</u>

<u>UNIT – IV</u>

<u>PART - A</u>

1. **State the pumping lemma for CFLs.** **<u>NOV/DEC 2012</u>**
   Refer notes
2. **What are the applications of Turing Machine?** **<u>NOV/DEC 2012</u>**
   Refer notes
3. **State pumping lemma for CFL.** **<u>MAY/JUNE 2012</u>**
   Refer notes
4. **What is chomsky normal form?** **<u>MAY/JUNE 2012</u>**
   Refer notes
5. **What is the height of the parse tree to represent a string of length 'n' using Chomsky normal form?** **<u>APR/MAY 2011</u>**
   n+1
6. **Construct a Turing machine to compute 'n mod 2' where n is represented in the tape in unary form consisting of only 0's.** **<u>APR/MAY 2011</u>**

	0	B	
q0	(q1,B,R)	(q2,B,R)	
q1	(q0,B,R)	(q3,B,R)	
q2	-	-	representing even no
q3	-	-	representing odd no

7. **Design a TM that accepts the language of odd integers written in binary.** **<u>NOV/DEC 2011</u>**
   To accept odd valued binary strings, we only have to look at the last bit. The TM moves right until it reads a blank, moves left one space and acepts if and only if there is a 1 on the tape.
8. **State the two normal forms and give an example.** **<u>NOV/DEC 2011</u>**
   1. Chomsky normal form ABC |a
   2. Greibach normal form XbYXZ|a
9. **Convert the following grammar G in greibach normal form.** **<u>APR/MAY 2010</u>**

SABb|a AaaA|B BbAb

No ε production in the given grammar

Eliminating unit production AB we have

SABb|a AaaA|bAb BbAb

Eliminating useless variables A & B (non generating)

Sa

10. Design a Turing machine with no more than three states that accepts the language a(a+b)*. Assume ∑ = {a,b} **APR/MAY 2010**

TM M=(Q, ∑, Γ, δ, q0, B,{q2})

Q – {q0,q1,q2}

∑ - {a,b}

Γ – (a,b,B}

q0 – Initial state

q2 - Final state

δ - Transition function given as follows

δ(q0,a) = (q1,a,R)

δ(q1,a) = (q1,a,R)

δ(q1,b) = (q1,b,R)

δ(q1,B) = (q2,B,R)

11. What is Turing machine? **NOV/DEC 2010**

TM is denoted by

M=(Q, ∑, Γ, δ, q0, B,F)

where

Q – A finite non empty set of states

∑ - A finite set of input symbols

Γ – A finite non empty set of tape symbols

δ - The transition function is given by

δ:Q X Γ Q X Γ X {L,R,S}

q0 – Initial state

B ε Γ – Blank Symbols

F - Final state

12. Is the language L={$a^n b^n c^n$ | n>=1} is context free?Justify. **NOV/DEC 2010**

It is not context free.

Pumping Lemma for CFL is not satisfied.

13. What is meant by Greibach Normal Form ? **MAY/JUNE 2013**

A context-free grammar is in **Greibach normal form** (GNF) if the right-hand sides of all production rules start with a terminal symbol, optionally followed by some variables. A nonstrict form allows one exception to this format restriction for allowing the empty word (epsilon, ε) to be a member of the described language

14. List the closure properties of Context Free Languages **MAY/JUNE 2013, NOV/DEC 2013**

Union, intersection, kleene closure, substitution, homomorphism

15. List out the different techniques for turing machine construction.NOV/DEC 2013

Refer notes

16. Let G be the grammar SaB|bA Aa|aS|bAA Bb|bS|aBB. For the string aaabbabbba, Find (a) LMD (b) RMD **NOV/DEC 2014**

Refer Notes

17. Define Diagonalization (Ld) Language. **NOV/DEC 2014**

Ld={wi | wi ∑ L(Mi)}

PART – B

1. Convert the following grammar into CNF

- **(i) ScBA, SA, AcB, AAbbS, Baaa** **(6) NOV/DEC 2012**
- **(ii) Sa|AAB, Aab|aB| ε, Baba| ε** **(8) APR/MAY 2011**
- **(iii) SA|CB, AC|D, B1B|1, C0C|0, D2D|2 (16) APR/MAY 2010**
- **(iv) SaAD AaB|bAB B b D d** **(6) NOV/DEC 2014**

2. State and prove the pumping lemma for CFL. What is its main application? Give two examples. **(10) NOV/DEC 2012, NOV/DEC 2011, MAY/JUNE 2013**

3. Design a Turing machine for the following

- **(i) Reverses the given string {abb}.** **(8) NOV/DEC 2012**
- **(ii) L={$1^n0^n1^n$|n>=1}** **(10) MAY/JUNE 2012**
- **(iii) L={$a^nb^nc^n$}** **(8) APR/MAY 2011**
- **(iv) To perform proper subtraction** **(8) APR/MAY 2011**
- **(v) To move an input string over the alphabet A ={a} to the right one cell. Assume that the tape head starts somewhere on a blank cell to the left of the input string. All other cells are blank, labeled by ^. The machine must move the entire string to the right one cell, leaving all remaining cells blank.** **(10) APR/MAY 2010**

(i) $L=\{1^n0^n|n>=1\}$ (8) NOV/DEC 2010

(vi) $L=\{ww^R|$ w is in $(0+1)^*\}$ (8) NOV/DEC 2010

(vii) Implement the function "MULTIPLICATION" using the subroutine "COPY". (12) NOV/DEC 2014

4. Write briefly about the programming techniques for TM. (8) NOV/DEC 2012,MAY/JUNE 2013

5. Find Greibach normal form for the following grammar

(ii) SAA | 1, ASS |0 (10) MAY/JUNE 2012

(iii) Sa|AB, Aa|BC, Bb, Cb (4) APR/MAY 2011

(iv) SAA|0, ASS|1 (8) NOV/DEC 2010

(v) $A_1A_2A_3$, $A_2A_3A_1|b$, $A_3A_1A_2|a$ (10) NOV/DEC 2014

6. Explain any two higher level techniques for Turing machine construction. (6) MAY/JUNE 2012, NOV/DEC 2011

7. Discuss the closure properties of CFLS. (6) MAY/JUNE 2012, NOV/DEC 2011,NOV/DEC 2010, MAY/JUNE 2013

8. Prove that every grammar with ε productions can be converted to an equivalent grammar without ε productions. (4) APR/MAY 2011, NOV/DEC 2013

9. Explain the different models of Turing machines. (10) NOV/DEC 2011

10. Define Pumping Lemma for Context Free Languages. Show that $L=\{a^i b^j c^k : i<j<k\}$ is not context free grammar (6) APR/MAY 2010

11. construct the following grammar in CNF
A->BCD/b
B->Yc/d
C->gA/c
D->dB/a
y->f MAY/JUNE 2013

12. Explain turing machine as a computer of integer functions with an example. NOV/DEC 2013

13. Write short notes on the following
(i) Two-way infinite tape TM
(ii) Multiple tracks TM NOV/DEC 2013

14. Show that langyage $\{0^n1^n2^n \mid n>=1\}$ is not context free language.(4) **NOV/DEC 2014**

UNIT – V

PART - A

1. **When we say a problem is decidable? Give an example of undecidable problem. NOV/DEC 2012**

   Refer notes

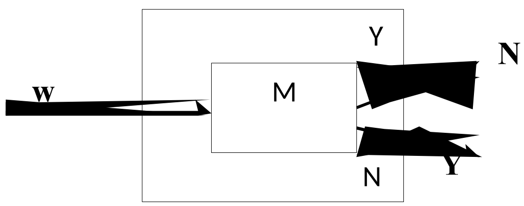
2. **What is recursively enumerable language? NOV/DEC 2012, MAY/JUNE 2012, NOV/DEC 2010, MAY/JUNE 2013, NOV/DEC 2013**

   REL is the language accepted by a Turing Machine.

3. **Mention the difference between P and NP problems. MAY/JUNE 2012**

   Refer notes

4. **Prove that the complement of recursive languages is recursive. APR/MAY 2011**



5. **Define the classes P and NP. APR/MAY 2011, MAY/JUNE 2013**

   P Solvable in polynomial time.

   NP Verifiable in Polynomial time.

6. **How to prove that the Post Correspondence problem is Undecidable. NOV/DEC 2011**

   Introduce a modified PCP and reduce the same to the original PCP.

   Again reduce L_u to the modified PCP.

   The chain of reduction infers if original L_u is known to be undecidable then conclude that PCP is undecidable.

7. **Show that any PSPACE-hard language is also NP-hard. NOV/DEC 2011**

   First we must show that the language is not in NP. This is trivial since NP is a subset of PSPACE and therefore, anything outside of PSPACE is also outside of NP.

   Then we must show that any problem in NP can be reduced to any PSPACE-hard language. Thus, any PSPACE-hard problem is also NP-hard.

8. **State Rice's theorem. APR/MAY 2010**

   Every non-trivial property of the RE language is undecidable.

   A property is trivial if it is either empty such that it is satisfied by no language or is all RE languages, or else it is non-trivial.

9. **Show that the collection of all Turing machines is countable. APR/MAY 2010**

   If for a set there is an enumerator, then the set is countable.

   Any Turing Machine can be encoded with a binary string of 0's and 1's.

   An enumeration procedure for the set of Turing Machine strings:

   Repeat

Generate the next binary string of 0's and 1's in proper order
Check if the string describes a Turing Machine
If Yes: Print string on output tape
If No: Ignore string

10. **Mention the difference between decidable and undecidable problems. NOV/DEC 2010**

Decidable Problem – Existence of an algorithm
Undecidable Problem – No algorithm for solving it.

11. **What is universal turing machine NOV/DEC 2013**

universal Turing machine (**UTM**) is a Turing machine that can simulate an arbitrary Turing machine on arbitrary input. The universal machine essentially achieves this by reading both the description of the machine to be simulated as well as the input thereof from its own tape

12. **Define multiple turing machine. NOV/DEC 2014**

An extended TM model has more number of tapes. A move is based on the state and on the vector of symbols scanned by the hand on each of the tapes.

13. **Give example for NP-complete problems. NOV/DEC 2014**

Traveling Salesman Problem

PART – B

1. **If L_1 and L_2 are recursive language then $L_1 U L_2$ is a recursive language.(6)NOV/DEC 2012**
2. **Prove that the halting problem is undecidable. (10)NOV/DEC 2012, NOV/DEC 2010**
3. **State and prove the Post's correspondence problem. (10)NOV/DEC 2012, NOV/DEC 2010**
4. **Write a note on NP problems. (6) NOV/DEC 2012**
5. **Explain undecidability with respect to post correspondence problem. (8) MAY/JUNE 2012**
6. **Discuss the properties of recursive languages. (8) MAY/JUNE 2012**
7. **Explain any two undecidable problems with respect to Turing machine. (8) MAY/JUNE 2012**
8. **Discuss the difference between NP-complete and NP-hard problems. (8) MAY/JUNE 2012, NOV/DEC 2011**
9. **Prove that the universal language L_u is recursively enumerable but not recursive. Also prove that L_d is not recursive or recursively enumerable. (16) APR/MAY 2011**

10. Prove that PCP problem is undecidable and explain with an example. (16) APR/MAY 2011, NOV/DEC 2013

11. State the halting problem of TMs. Prove that the halting problem of Turing Machine over {0,1}* as unsolvable. (8) NOV/DEC 2011

12. Let ∑={a,b}*. Let A and B the lists of three strings as given below:

A={b,bab3,ba} B={b3,ba,a}

Does this instance of PCP have a solution? Justify your answer.(8) NOV/DEC 2011

13. Write short notes on recursive and recursively enumerable languages. (8) NOV/DEC 2011

14. Consider the language of all TMs that given no input eventually write a non-blank symbol on their tapes. Explain why this set is decidable. Why does this not conflict with the halting problem? (8) APR/MAY 2010

15. Prove that the Post Correspondence Problem is decidable for strings over the alphabet (0). (8) APR/MAY 2010

16. Prove that the problem of determining if the languages generated by two CFGs are equal is undecidable. (8) APR/MAY 2010

17. Prove that the Punchcard Puzzle is NP-compete. (8) APR/MAY 2010

18. Explain the difference between tractable and intractable problems with examples. (10) NOV/DEC 2010

19. Explain any four NP-Complete Problems. (8)NOV/DEC 2010,NOV/DEC 2013

20. (i) Explain about "A language that is not recursively enumerable"

(ii)Prove L_{ne} is recursively enumerable. MAY/JUNE 2013

21. (i)Discuss on undecidable problems about turing machine.

(ii)Explain about the PCP MAY/JUNE 2013

22. Prove that for two recursive languages L_1 and L_2 their union and intersection is recursive. NOV/DEC 2013

23. Prove that if a language is recursive if and only if it and its complement are both recursively enumerable. NOV/DEC 2013

24. **(i) Show that the union of two recursive language is recursive & union of two recursively enumerable languages is recursive. (12) NOV/DEC 2014**
(ii)Define the language Lu and show that Lu is RE language. (4) NOV/DEC 2014
25. **State and prove Post Correspondence Problem and Give example. (16) NOV/DEC 2014**