DHANALAKSHMI SRINIVASAN ENGINEERING COLLEGE,

PERAMBALUR – 621 212

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

THEORY OF COMPUTATION

QUESTION BANK

UNIT-I AUTOMATA

Part-A

1. Define Deductive Proof.

2. Describe the language accepted by the DFA.

3. Obtain the DFA equivalent to the following NFA.

	0	1
--> q0	q0	q0
q1	q1	-
* q2	-	q2

4. Obtain an NFA without ε-transition to the following NFA with ε-transition.

	0	1	2	ε
--> q0	q0	-	-	q1
q1	-	q1	-	q2
* q2	-	-	q2	-

5. Is it true that language accepted by any NFA is different from the regular language? Justify your answer.

6. Find the language accepted by a DFA given below.

	0	1
--> q0	q0	q1
* q1	q1	-

7. Find the ε-closure of the states 1, 2 and 4 in the following transition table.

	a	b	ε

→	1	-	-	2
	2	-	-	3
	3	-	-	6
	4	5	-	-
	5	-	6	7
	6	-	-	-
*	7	-	-	-

8. Construct a finite automaton that accepts $\{0, 1\}^+$.

9. The transition table for the NFA (without ε-transitions) is given below. Find the transition function δ(q,a).

		0	1	2
-->*	q0	{q0,q1,q2}	{q1,q2}	q2
*	q1	φ	{q1,q2}	q2
*	q2	φ	φ	q2

10. What is a finite automaton? Give examples.

11. Enumerate the difference between DFA and NFA.

12. List any four ways of theorem proving.

13. Define NFA with ε-transition. Is the NFA's with ε-transitions are more powerful than the NFA's without ε-transitions? (in the sense of language acceptance).

14. Define Mutual Induction.

15. Determine an NFA accepting the language L={a* U b*}.

16. Construct finite automata for the language $(0^n \backslash n \bmod 3 = 2, n \geq 0)$.

17. Define transition diagram and transition table.

18. Define the language of NFA.

19. List the uses of Finite Automata.

20. Design a DFA which accepts the string ends with 11.

21. Differentiate L* and L+.

Part-B.

1. Prove the theorem "Let S be a finite subset of some infinite set U. Let T be the complement of S with respect to U. Then T is infinite" using reduction to definitions. (16)

2. Consider the following ε-NFA.

		ε	a	b	c
-->	p	{q,r}	φ	{q}	{r}
	q	φ	{p}	{r}	{p,q}
*	r	φ	φ	φ	φ

Compute the ε-closure and convert the automaton to a DFA. (12)

3. Construct the DFA accepting the language over the alphabet {0,1}. L = The set of all strings ending in 00. (4)

4. Prove the theorem: If x>=4, then $2^x >= x^2$, using deductive proof. (6)

5. Design a DFA that eliminate the ε-transitions from the ε-NFA. (10)

		ε	0	1	2
-->	q0	q1	q0	-	-
	q1	q2	-	q1	-
*	q2	-	-	-	q2

6. Let L be a set accepted by an NFA. Then prove that there exists a deterministic finite automaton that accepts L. Is the converse true? Justify your answer. (10)

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

8. Consider the following ε-NFA. Compute the ε- Closure of each state and find its equivalent DFA. (8)

9. Convert the following NFA to its equivalent DFA. (8)

		0	1
-->	p	{p,q}	{p}
	q	{r}	{r}

	r	{s}	φ
*	s	{s}	{s}

10. Construct a DFA equivalent to the NFA M=({p,q,r},{0,1},δ,p,{q,s}), where δ is defined in the following table. (10)

	δ	0	1
-->	p	{q,s}	{q}
	q	{r}	{q,r}
	r	{s}	{p}
*	s	φ	{p}

11. If L is accepted by an NFA with ε-transition then show that L is accepted by an NFA without ε-transition. (8)

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

13. Consider the following ε-NFA. Compute the ε-Closure of each state and find it's equivalent DFA. (8)

		ε	A	b	c
-->	p	{q}	{p}	φ	φ
	q	{r}	φ	{q}	φ
*	r	φ	φ	φ	{r}

14. Draw the NFA to accept the following languages.

i. Set of strings over alphabet {0,1,…9} such that the final digit has appeared before. (8)

ii. Set of strings of 0's and 1's such that there are two 0'a separated by a number of positions that is a multiple of 4. (8)

15. Prove the following by the principle of induction (6)

$$\sum_{K=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}$$

16. Construct a DFA that accepts all the strings on (0,1) except those containing the substring 101. (6)

17. For the finite state machine M given in the following table, test whether the strings 101101, 11111 are accepted by M. (4)

		0	1
-->	q0	q0	q0
	q1	q3	q0
	q2	q0	q3
*	q3	q1	q2

18. Construct a non-deterministic finite automation accepting the same set of strings over {a,b} ending in aba. Use it to construct a DFA accepting the same set of strings. (10)

19. Draw transition diagram for recognizing the set of all operators in C language. (10)

20. Explain the extended transition function for NFA, DFA and ε-NFA. (6)

UNIT-II REGULAR EXPRESSIONS AND LANGUAGES

Part-A

1. Show that (r*)*=r* for a regular expression r.
2. Describe the following sets by regular expressions:
   a. L_1 = the set of all strings of 0's and 1'a ending in 00.
   b. L_2 = the set of all strings of 0's and 1's beginning with 0 and ending with 1.
3. Explain the construction of NFA with ε-transition from any given regular expression.

4. Verify whether L={a^{2n}/n≥1} is regular.
5. Mention the closure properties of regular languages.
6. Show that the complement of a regular language is also regular.
7. Write the regular expression for the following language: the set of strings of 0's and 1's whose sixth symbol from the right end is 1.
8. How does u test the emptiness of regular languages?
9. Let R be any set of regular languages. Is U R_i regular? Prove it.
10. What are the applications of Regular expressions?
11. What is relationship between FA and Regular Expression?

Part-B.

1. Let r be a regular expression. Then prove that there exists an NFA with ε-transition that accepts L(r). (6)
2. Obtain the Regular Expression R for the following DFA A such that L(A)=L(r) (10)
3. Construct an NFA equivalent to the regular expression (0+1)*(00+11)(0+1)*. (8)
4. Obtain the regular expression that denotes the language accepted by the following DFA. (8)

	0	1
--> 1	2	2
* 2	1	2

5. Construct an NFA equivalent to (0+1)* (00+11). (8)
6. Construct a regular expression corresponding to the state diagram give in the following figure. (10)

	0	1
-->* q1	q1	q2
q2	q2	q3
q3	q1	q2

7. Explain the construction of NFA with ε-transition from any given regular expression. (8)
8. Find the regular expression for the set of all strings denoted by R^2_{13} from the deterministic finite automata given below. (8)

	0	1

-->	1	2	3
	2	1	3
*	3	2	2

9. Show that the language L={0^{n^2}/n≥1, n is an integer} is not regular. (4)

10. Show that every set accepted by a DFA is denoted by a regular expression. (8)

11. Construct an NFA equivalent to the following regular expression 01*+1. (8)

12. Construct an NFA equivalent to the following regular expression ((10) + (0+1))*01. (10)

13. Prove that if L=L(A) for some DFA A, then there is a regular expression R such that L= L(R). (10)

14. Construct transition diagram of a finite automaton corresponding to the regular expression (ab+c*)b. (8)

15. Find the regular expression corresponding to the finite automaton given below. (8)

		0	1
--> *	q1	q1	q2
*	q2	q3	q2
	q3	q3	q3

16. Show that the language {0^p, p is prime} is not regular. (6)

17. Find whether the languages {ww,w is in (1+0)*, and {1^k/k=n^2, n≥1} are regular or not. (8)

18. Show that the regular languages are closed under intersection and reversal. (8)

19. Construct an NFA equivalent to the regular expression 10+ (0+11)0*1. (6)

20. Find the equivalence and minimal DFA for the below DFA. (6)

	a	b

--> q0	q1	q3
q1	q2	q4
q2	q1	q4
q3	q2	q4
* q4	q4	q4

21. Obtain the regular expression that denotes the language accepted by (12)

	0	1
--> q1	q2	q3
* q2	q1	q3
* q3	q2	q2

22. Prove: If L is a regular language over alphabet Σ and h is a homomorphism on Σ, then h (L) is also regular. (10)

23. Construct an NFA equivalent to the regular expression (0+1)*(00+11)(0+1)* (6)

24. Construct a minimum state automaton equivalent to a given automaton M whose transition table is given below. (8)

	a	b
--> q0	q0	q3
q1	q2	q5
q2	q3	q4
q3	q0	q5
q4	q0	q6
q5	q1	q4
* q6	q1	q3

UNIT-III CONTEXT FREE GRAMMAR AND LANGUAGES

Part-A

1. Let G=({S,C},{a,b},P,S) where P consists of S->aCa,C->aCa/b. Find L(G).
2. Consider G whose productions are S->aAS/a, A->SbA/SS/ba. Show that S=>aabbaa and construct a derivation tree whose yield in aabbaa.
3. Construct a CFG for generating the language L={$a^n b^n$/n≥1}
4. Let G be the grammar S->aB/bA, A->a/aS/bAA, B->b/bS/aBB. For the string aaabbabbba find a leftmost derivation.
5. What is the additional feature PDA has when compared with NFA? Is PDA superior over NFA in the sense of language acceptance? Justify your answer.
6. Explain what actions take place in the PDA by the transitions (moves) δ(q,a,Z) = {(p1,γ1), (p2,γ2),..., (pm,γm)} and δ(q,ε,Z) = {(p1,γ1), (p2,γ2),..., (pm,γm)}.
7. Find L (G)Ψ where G=({S}, {0,1}, {S->cS1, S->ε},S).
8. Define derivation tree for a CFG.
9. Give an example of a PDA.
10. Define the acceptance of a PDA by empty stack. Is it true that the language accepted by a PDA by empty stack or by that of final states is different languages?

11. Consider the alphabet ∑ = {a, b, (,), +, *, ., ε}. Construct a CFG that generates all strings in ∑* that are regular expressions over the alphabet {a,b}.

12. Write a CFG to generate the set $\{a^m b^n c^p / m+n = p \text{ and } p \geq 1\}$.

13. Define the languages generated by a PDA using final state of the PDA and empty stack of that PDA.

14. Fine the language generated by a CFG G=({S},{0,1},{S->0/1/ε,S->0S0/1S1},S).

15. Obtain a derivation tree for the grammar G=({S.A}, a, b}, P, S) where P consists of S->aAS/a, A->SbA/SS/ba

16. Give the formal definition of Pushdown Automation.

17. Define the languages generated by a PDA using final state of the PDA and empty stack of that PDA.

18. Is it true that deterministic pushdown automata and non-deterministic pushdown automata are equivalent in the sense of language of acceptance? Justify your answer.

19. Construct CFG to generate $\{a^n b^n | n€Z^+\}$.

20. Define Instantaneous description of a PDA.

21. Give an example for a context-free grammar.

22. Let the productions of the grammar be S->0B|1A, A->0|0S|1AA, B->1|1S|0BB. For the string 0110 find a rightmost derivation.

23. Is it true that nondeterministic PDA is more powerful than that of deterministic PDA? Justify your answer.

Part-B

1. State the pumping lemma for regular sets. Show that the set L= $\{O^{i2}/I$ is an integer, i≥1} is not regular. (6)

*

2. Let G=(V,T,P,S) be a context-free grammar. Then prove that S=>α if and only if there is a derivation tree in grammar G with yield α. (16)
3. If L is N (M_1) (the language accepted by empty stack) for same PDA M_1, then L is L (M_2) (language accepted by final state) for some PDA M_2. (6)
4. Let M=({q0,q1},{0,1},{X_1,Z_0}, δ is given by
   δ(q0,0,Z0)={(q0,XZ0)}

   δ(q0,0,X) ={(q0,XX)}

   δ(q0,1,X) ={(q1,ε)}

   δ(q1,1,X) ={(q1,ε)}

   δ(q1,ε,X) ={(q1,ε)}

$\delta(q1,\varepsilon,Z0)=\{(q1,\varepsilon)\}$

Construct a CFG G=(V,T,P,S) generating N(M). (10)

*

5. Let G=(V,T,P,S) be a CFG. Show that if S=>α, then there is a derivation tree in a grammar G with yield α. (6)

6. Construct a PDA accepting $\{a^n b^m a^n / m,n>=1\}$ by empty stack. Also construct the corresponding context-free grammar accepting the same set. (16)

7. Prove that L is L (M_2) for some PDA M_2 if and only if L is N (M_1) for some PDA M_1. (10)

*

8. Let G be a CFG and let A=>w in G. Then show that there is a leftmost derivation of w. (6)

9. Let G be the grammar S->0B|1A, A->0|0S|1AA, B->1|1S|0BB. For the string 00110101 find its leftmost derivation and derivation tree. (4)

10. If G is the grammar S->Sbs|a, show that G is ambiguous. (6)

11. If L is L (M_2) for some PDA M_2, then show that L is N (M_1) for some PDA M. (6)

12. Construct a context-free grammar G which accepts N (M), where M=({q0,q1},{a,b},{z0,z},δ,q0,z0,φ) and where δ is given by

$\delta(q0,b,z0)=\{(q0,zz0)\}$

$\delta(q0,\varepsilon,z0)=\{(q0,\varepsilon)\}$

$\delta(q0,b,z)=\{(q0,zz)\}$

$\delta(q0,a,z)=\{(q,z)\}$

$\delta(q,b,z)=\{(q,\varepsilon)\}$

$\delta(q1,a,z0)=\{(q0,z0)\}$ (10)

13. If L is CFL then prove that there exists a PDA M such that L=N (M). (10)

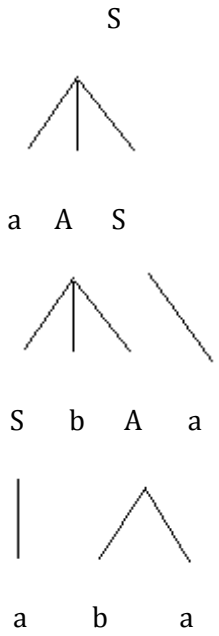
14. Prove that if L=N (P_N) for some PDA $P_N=(Q,\sum,\Gamma,\delta_N,q0,Z0)$, then there is a PDA P_F such that L=L (P_F). (8)

15. Construct a PDA for $\{a^n b^m a^{2(m+n)} / n,m\geq 0\}$. (8)

16. Show that the grammar S->aSbS|bSaS|ε is ambiguous and what is the language generated by this grammar? (6)

17. Write a grammar to recognize all prefix expressions involving all binary arithmetic operators. Construct parse tree for the sentence "-*+abc/de" using your grammar. (6)

18. Find the left most and right most derivation corresponding to the parse tree. (4)



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

20. Show that if L is a CFL then there exists a PDA M such that L=N (M). (16)

21. G denotes the context-free grammar defined by the following rules. (4x4=16)

    S->ASB|ab|SS, A->aA|ε, B->bB|ε

    i. Give a leftmost derivation of aaabb in G. Draw the associated parse tree.
    ii. Give a rightmost derivation of aaabb in G. Draw the associated parse tree.
    iii. Show that G is ambiguous. Explain with steps.
    iv. Construct an unambiguous grammar equivalent to G. Explain.

22. Construct a pushdown automaton that recognizes the language $\{a^i b^j c^k | I,j,k > 0$ and i=j or i=k$\}$. (8)

23. Discuss about PDA acceptance (8)

    i. From empty stack to final state.

ii. From final state to empty state.

24. Prove that if L is a context-free language then there exists a PDA on such that L =N (M). (12)

25. Explain different types of acceptance of a PDA. Are they equivalent in sense of language acceptance? Justify your answer. (4)

26. Give a detailed description of ambiguity in context-free grammar. (6)

27. Find a derivation tree of a*b+a*b given that a*b+a*b is in L (G) where G is given by S->S+S |S*S, S->a/b. (6)

28. Show that the grammar S->a|abSb|aAb, A->bS|aAAb is ambiguous. (6)

29. Construct a PDA for the language $\{a^n b^{2n} | n>=0\}$ (8)

30. Construct a PDA for the grammar:

S->aB|bA, A->a|aS|bAA, B->b|bS|aBB (8)

UNIT-IV PROPERTIES OF CONTEXT-FREE LANGUAGES

Part A.

1. Define Turing Machine.
2. Define Instantaneous description and move of a Turing machine.
3. Explain the basic Turing Machine model and explain in one move. What are the actions take place in a Turing Machine?
4. Explain how a Turing machine can be regarded as a computing device to compute integer functions.
5. Define multitape Turing machine. Is it more power than the basic turing machine? Justify your answer.
6. 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?
7. Can you say the language generated by a CFG in CNF id finite or infinite? If so, how? If not, why?
8. What is the class of language for which the TM has both accepting and rejecting configuration? Can this be called a context free language?
9. The binary equivalent of a positive integer is stored in a tape. Write the necessary transitions to multiply that integer by 2.
10. Using pumping lemma show that the language $L=\{a^n b^n c^n / n \geq 1\}$ is not a context-free language.
11. What is meant by a Turing Machine with two-way infinite tape?
12. State the pumping lemma for CFL.
13. Give the formal definition of a Turing Machine.
14. Mention any two problems which can be solved by Turing Machines.
15. What are the features of Turing Machine?

Part B

1. State the pumping lemma for regular sets. Show that the set L={O^{i^2}/i is an integer, i≥1} is not a regular. (6)
2. Find a grammar in Chomsky normal form equivalent to S->aAbB, A->aA/a, B->bB/b. (6)
3. Construct a grammar in Greibach normal form equivalent to the grammar S->AA/a, A->SS/b. (10)
4. Design a Turing Machine to compute f(m+n)=m+n, for all m, n≥0 and simulate their action on the input 0100. (10)
5. Describe the following Turing Machine and their working. Are they more powerful than the Basic Turing Machine?
    1) Multi-tape Turing Machine
    2) Multi-dimensional Turing Machine
    3) Non-deterministic Turing Machine (6)
6. Design a Turing machine M to implement the function "multiplication" using the subroutine 'Copy'. (12)
7. Explain how a turing Machine with the multiple tracks of the tape can be used to determine the given number is prime or not? (4)
8. Find a grammar in Chomsky Normal form equivalent to S->aAD, A->aB|bAB, B->b, D->d. (6)
9. convert to Grecbach normal form the grammar G=({A_1,A_2,A_3},{a,b},P,A_1}, where P consists of the following:
   A_1->A_2A_3, A_2->A_3A_1|b, A_3->A_1A_2|c (10)

10. Show that the language {$0^n1^n2^2$/n≥1} is not a context-free language. (6)
11. Suppose G is a CFG and w, of length l, is in L (G). How long is a derivation of w in G if G is in CNF and if G is in GNF? (4)
12. Show that every CFL without ε can be generated by a CFG in CNF. (4)
13. Simplify the following grammar and find it's equivalent on CNF. S->bA|aB, A->bAA|aS|a, B->aBB|bS|b (8)
14. Find the GNF equivalent of the grammar S->AA|0, A->SS|1. (4)
15. Design a Turing Machine M for f(x,y,z)=2(x+y)-z, z<2(x+y) and x,y,z are stored in the tape in the form $0^x10^y10^z1$. (12)
16. Show that if L is accepted by a multi tape Turing Machine, it is accepted by single tape Turing Machine also (4)
17. State and prove pumping lemma for context-free languages. (16)
18. Explain in detail: "The Turing Machine as a computer of integer functions". (8)
19. Design a Turing Machine to accept the language L={0^n1^n/n≥1} (8)
20. What is the rule of checking off symbols in a Turing Machine. (4)
21. Design a Turing Machine to implement "Proper Subtraction". (12)
22. Design a Turing Machine M that decides A={0^k|n>0 and k=2^n}, the language consisting of all strings of 0s whose length is a power of 2. (16)
23. Demonstrate the working of your TM with an example. (4)
24. Explain with an example how the finite control of a TM can be used to hold a finite amount of information. (6)
25. Explain how a Turing machine can be viewed as a computing device on functions involving integers. (4)
26. Construct a Turing Machine that recognizes the language {wcw/w in {a+b}*}. (16)
27. Convert the grammar S->AB, A->BS|b, B->SA|a into Greibach normal form. (8)
28. Simplify the following grammar and find it's equivalent in CNF. (8)
    S->AB|CA, B->BC|AB, A->a, C->aB|b

29. Find the GNF equivalent of the grammar S->AA|0, A->SS|1 (8)
30. State Pumping lemma for context free language. Show that $\{a^n b^n c^n / n$ is an integer, $n \geq 1\}$ is not context free language. (6)
31. Explain how the 'finite control' of a Turing Machine can be used to hold a finite amount of information with an example. (4)

UNIT V UNDECIDABILITY

Part-A

1. When a problem is said to be decidable and give an example of an undecidable problem?
2. Expalin the modified Post's Correspondence Problem.
3. Give two examples of Undecidable Problem.
4. Is it true that complement of a recursive language id recursive? Justify your answer.
5. When we say a problem is decidable? Give an example of undecidable problem?
6. Give two properties of recursively enumerable sets which are undecidable.
7. Show that the following problem is undecidable. "Given two CFGs G_1 and G_2, is L (G_1) L (G2)=φ?"
8. Define L_d.

9. Show that the union of two recursively enumerable languages is recursively enumerable.
10. What is undecidability problem?
11. Define Recursively Enumerable language.
12. Give an example for a non-recursively enumerable language.
13. Difference between recursive and recursively enumerable languages.
14. Mention any two undecidability properties for recursively enumerable languages.
15. Define Diagonal Language.

Part B

1. Define Universal language L_u. Show that L_u is recursively enumerable but not recursive. (16)
2. Show that the complement of a recursive language is recursive. (6)

3. If a language L and its complement L are both recursively enumerable then show that L and hence L complement is recursive. (6)
4. Obtain the code for <M,1011>, where M=({q1,q2,q3},{0,1},{0,1,B},δ,q1,B{q2}) have moves:
   $\delta(q1,1) = (q3,0,R)$

   $\delta(q3,0) = (q1,1,R)$

   $\delta(q3,1) = (q2,0,R)$

   $\delta(q3,B) = (q3,1,L)$ (4)

5. Show that L_u is recursively enumerable but not recursive (16)
6. Define L_d and show that L_d is not recursively enumerable (12)
7. Whether the problem of determining given recursively enumerable language is empty or not? Is decidable? Justify your answer. (4)
8. Show that union of recursive language is recursive. (4)
9. Define the language L_d and show that L_d is not recursively enumerable language. (8)
10. Explain the Halting problem. Is it decidable or undecidable problem? (4)
11. Find whether the following languages are recursive or recursively enumerable.
    i. Union of two recursive languages. (4)
    ii. Union of two recursive enumerable languages. (4)
    iii. L if L and complement of L are recursively enumerable. (4)
    iv. L_u.
12. Show that "Finding whether the given CFG is ambiguous or not" is undecidable by reduction technique. (4)
13. Consider the Turing Machine M and w=01, where M=({q1,q2,q3},{0,1},{0,1,B},δ,q1,B,{q3}) and δ is given by

qi	$\delta(q_i,0)$	$\delta(q_i,1)$	$\delta(q_i,B)$
q1	(q2,1,R)	(q2,0,L)	(q2,1,L)

q2	(q3,0,L)	(q1,0,R)	(q2,0,R)
q3	-	-	-

Reduce the above problem to Post's Correspondence Problem and find whether that PCP has a solution has a solution or not. (12)

14. Show that any non-trivial property J of the recursively enumerable languages is undecidable. (8)
15. Show that the language is not undecidable. L={<M>|M is a TM that accepts the string aaab}. (8)
16. Discuss the properties of Recursive and Recursive enumerable languages. (8)
17. Define Post Correspondence problem with an example. (8)
18. Prove that the function $f(n)=2^n$ does not grow at a polynomial rate, in other words, it does not satisfy $f(n) = O(n^p)$ for any finite exponent p. (8)
19. Define the language L_d. Show that L_d is neither recursive nor recursively enumerable. (12)
20. State and prove Rice's Theorem for Recursive index sets. (16)
21. Does PCP with two lists x=(b,bab^3,ba) and y=(b^3,ba,a) have a solution? (5)
22. Show that the characteristic function of the set of all even numbers is recursive. (6)