(An Autonomous Institution, Affiliated to Anna University, Chennai)

DEPARTMENT: Computer Science and Engineering /Information Technology

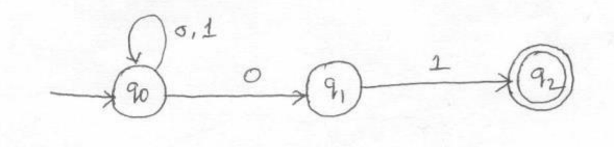
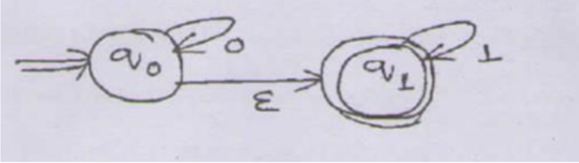
IV - Semester

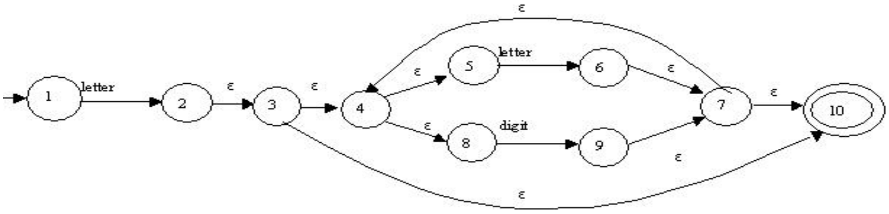
CS3452- THEORY OF COMPUTATION

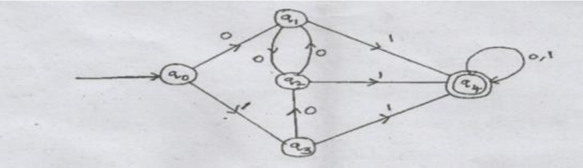
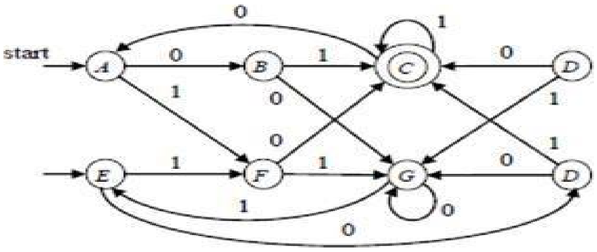
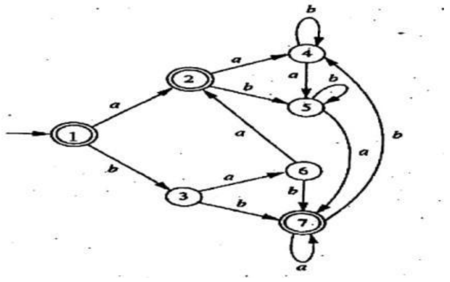
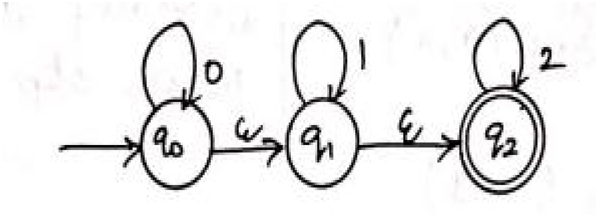
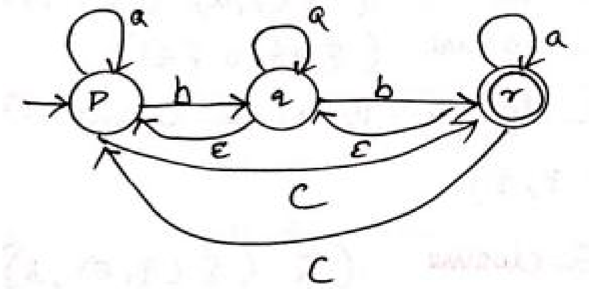
(B.E/B.Tech – CSE/IT)

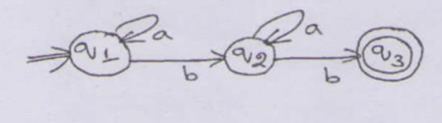
(Regulations 2021)

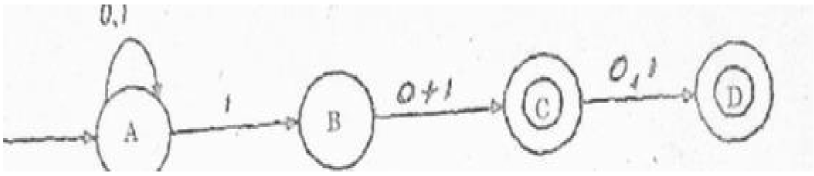
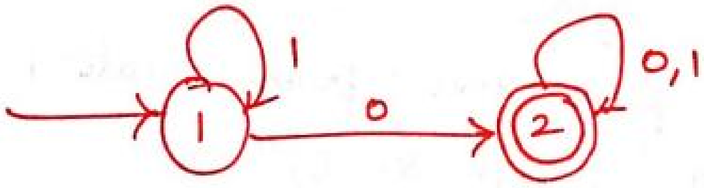
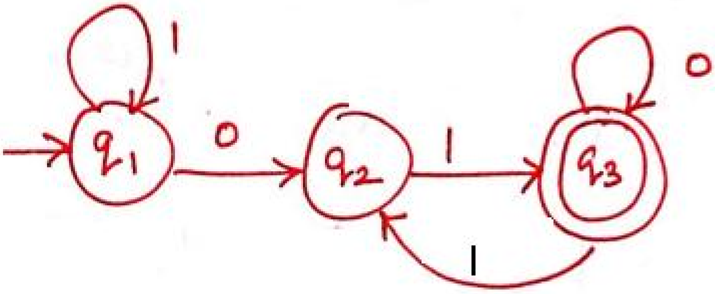
	UNIT I **AUTOMATA AND REGULAR EXPRESSIONS** Need for automata theory - Introduction to formal proof – Finite Automata (FA) – Deterministic Finite Automata (DFA) – Non-deterministic Finite Automata (NFA) – Equivalence between NFA and DFA – Finite Automata with Epsilon transitions – Equivalence of NFA and DFA- Equivalence of NFAs with and without ε-moves- Conversion of NFA into DFA – Minimization of DFAs
	PART A
1.	Define computation and theory of computation.
2	What is the need for automata?
3.	Design FA which accepts odd number of 1's and any number of 0's.
4.	Design FA to check whether given binary number is divisible by three.
5.	State the difference between NFA & DFA.
6.	Define the term ϵ transition
7.	Draw a NFA to accept strings containing the substring 0101
8.	Define ϵ –Closure (q) with an example
9.	Define Non-Deterministic Finite Automata
10.	Define Deterministic Finite Automata
11	What is structural induction?
12	What is proof by contradiction?
13	Prove 1+2+3+…+n= n(n+1)/2 using induction method
14	Write any three applications of Automata Theory
15	Define (i) Finite automaton? (ii)Transition diagram
16	Define deductive proof

17	Give some examples for additional forms of proof.	
18	Define hypothesis.	
19	Define inductive proof.	
20	Draw non-deterministic automata to accept strings containing the substring 0101.	
	Unit I - Part B	
1.	Prove the following by induction for all n≥0 (i) $1^2+2^2+3^2+4^2+\ldots\ldots.+n^2=(n(n+1)(2n+1))/6$ (ii) $1^3+2^3+\ldots.+n^3=(n^2(n+1)^2)/4$	13
2.	Prove the following by mathematical induction method (i) $2^n > n$ for all n≥0 (ii) $x \geq 4$, $2^x \geq x^2$	13
3.	Construct DFA equivalent to the NFA given below 	13
4.	Design DFA to accept the Language L={w/w has both even number of 0‘s and even number of 1‘s}	13
5.	i) Construct an NFA for the set of strings with {0,1} ending with 01 draw the transitiontable for the same and check whether the input string 00101 is accepted by above NFA. ii) Construct NFA for set of all strings {0,1} that ends with three consecutive 1‘sat its end.	13
6.	i) If a Regular language L is accepted by a non – deterministic finite automata then there exists a Deterministic Finite Automata that accepts L. ii) A Language _L‘ is accepted by some ε – NFA if and only if L is accepted by NFA without ε transition	13
7.	Construct NFA without ε transitions for the NFA given below 	13

<table>
<tr><td>8.</td><td>i) Construct a DFA that accepts all strings on {0, 1} except those containing the substring 101.
ii) Construct a NFA accepting the set of strings over {a,b} ending in aba. Use it to constructa DFA accepting the same set of strings.</td><td>13</td></tr>
<tr><td>9.</td><td>i) Convert the following NFA-with ε, to a NFA- without ε
<table>
<tr><th></th><th>0</th><th>1</th><th>2</th><th>ε</th></tr>
<tr><td>q_0 (start)</td><td>{q_0}</td><td>{φ}</td><td>{φ}</td><td>{q_1}</td></tr>
<tr><td>q_1</td><td>{φ}</td><td>{q_1}</td><td>{φ}</td><td>{q_2}</td></tr>
<tr><td>* q_2</td><td>{φ}</td><td>{φ}</td><td>{q2}</td><td>{φ}</td></tr>
</table>
ii) Convert to a DFA, the following NFA
<table>
<tr><th></th><th>a</th><th>b</th></tr>
<tr><td>p(start)</td><td>{p}</td><td>{p,q}</td></tr>
<tr><td>q</td><td>{r}</td><td>{r}</td></tr>
<tr><td>r</td><td>{ Φ }</td><td>{ Φ }</td></tr>
</table></td><td>13</td></tr>
<tr><td>10.</td><td>Define ε-NFA. Consider the following ε-closure of each state and find it‘s equivalent DFA.

</td><td>13</td></tr>
<tr><td>11.</td><td>Design a NFA accept the following strings over the alphabets {0,1} that begins with 01 and ends with 11. Check for the validity of 01111 and 0110 strings.</td><td>13</td></tr>
<tr><td>12.</td><td>Convert to a DFA, the following NFA.
<table>
<tr><th></th><th>0</th><th>1</th></tr>
<tr><td>P(start)</td><td>{p,q}</td><td>{p}</td></tr>
<tr><td>Q</td><td>{r}</td><td>{r}</td></tr>
<tr><td>R</td><td>{ s }</td><td>-</td></tr>
<tr><td>S</td><td>{s}</td><td>{s}</td></tr>
</table></td><td>13</td></tr>
<tr><td>13.</td><td>Construct the DFA which accepts a language of all string not starting with ‘a’ or not ending with ‘b’</td><td>13</td></tr>
<tr><td>14.</td><td>Give NFA to accept the following languages over {0,1}
(i) L = {String that contains either 101 or 110 as a substring}
(ii) L = {Strings such that every 1 is following immediately by 00}</td><td>13</td></tr>
<tr><td>15.</td><td>Design a DFA to accept the language L = {w/w has both an even number of 0’s and even number of 1’s} and illustrate from its transition function to check the string w= 110101.</td><td>13</td></tr>
<tr><td></td><td><b>Unit I - PART – C</b></td><td></td></tr>
</table>

1.	Minimize the following DFA  	15
2.	Minimize the following DFA.  	15
3.	Minimise the following DFA.  	15
4.	Convert ∈ - NFA to NFA  	15
5.	Convert ∈ - NFA to NFA  	15

	UNIT II REGULAR EXPRESSIONS AND LANGUAGES 9 Regular expression – Regular Languages- Equivalence of Finite Automata and regular expressions – Proving languages to be not regular (Pumping Lemma) – Closure properties of regular languages.	
	UNIT II - PART A	
1.	What is regular expression?	
2.	Give the regular expression for set of all strings ending in 00	
3.	Is regular set is closed under complement? Justify	
4.	Construct NFA for the regular expression (0+1)01	
5.	Show whether a language $L=(0^n1^{2n}/n>0\}$ is regular or not using pumping Lemma	
6.	Prove or disprove that (r+s)*=r*+s*.	
7.	Give the description for the following language (0+10)*1.	
8.	State pumping lemma for regular language.	
9.	Give the regular expression for the following L1= set of all strings of 0 and 1 ending in 00 L2= set of all string 0 and 1 beginning with 0 and ending with 1	
10.	Construct NFA for the regular expression (0+1)01	
	UNIT II – Part B	
1	Construct an ϵ- NFA for the regular expression b+ba*.	13
2	Which of the following language is regular? Justify. i. $L=\{ a^nb^m/n,m>0\}$ ii. $L=\{ a^nb^n/n,>0\}$	13
3	 Obtain the regular expression for the finite automata.	13
4	i) Using pumping lemma for the regular sets, prove that the language $L=\{a^mb^n/m>n\}$ is notregular. ii) Prove any two closure properties of regular languages.	13

5	Convert the following NFA into a R.E  	13
6	Construct a finite automata for the RE 10+(0+11)0*1	13
7	Prove that the following languages are not regular: L= {w€{a,b}*/w = w^r}	13
8	Show that the regular language are closed under : Union ,Intersection, Kleene closure, Complement, Difference	13
9	State pumping lemma and hence find whether the given language is regular or not L={a^{n}b^n/m>n}	13
10	Prove the following statement with justification. “the language $L = \{a^i b^j c^i \| i,j \ > 0\}$ is not regular”	13
	UNIT II – Part C	
1	Convert the following to regular expression  	15
2	Convert the following to regular expression  	15
3	For every DFA A=$\{Q, \sum \delta, q_0\, , F\}$ there is a regular expression r such that L(r) = L(A).	15
4	Prove that there exists an NFA with €- transition that accepts the regular expression r.	15

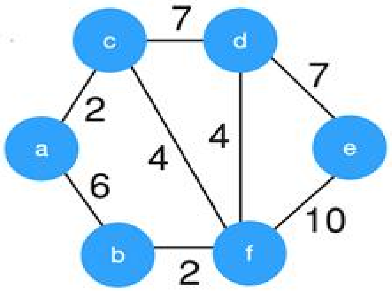
	UNIT III CONTEXT FREE GRAMMAR AND PUSH DOWN AUTOMATA 9 Types of Grammar - Chomsky's hierarchy of languages -Context-Free Grammar (CFG) and Languages – Derivations and Parse trees – Ambiguity in grammars and languages – Push Down Automata (PDA): Definition – Moves - Instantaneous descriptions -Languages of pushdown automata – Equivalence of pushdown automata and CFG-CFG to PDA-PDA to CFG – Deterministic Pushdown Automata.	
	UNIT-III PART A	
1	Define CFG with an example.	
2	Construct a CFG for the language of palindrome string over {a, b}.	
3	What is ambiguity?	
4	Define sentential form.	
5	Define parse tree.	
6	What is a derivation?	
7	When do you say grammar is ambiguous?	
8	Define DPDA and NPDA.	
9	What are the two normal forms of CFG?	
10	What are the conventional notations of PDA?	
11	What are the different ways of language acceptance by a PDA and define them?	
12	What are the 2 types of derivation?	
13	Define Deterministic PDA.	
14	Let G = ({S,A}, {a,b}, P,S} where P consists of S aAa, A aAa/b , Find L(G))?	
15	What are the properties of the CFL generated by a CFG?	
16	Construct a parse tree of (a+b)*c for the grammar E→E+E/E*E/(E)/id	
17	Construct a CFG for the set of strings that contain equal number of a's and b's over ∑={a,b}.	
18	Define PDA.	
19	Let the production of the grammar be S→0B | 1A, A→ 0 | 0S | 1AA, B→ 1|1S | 0BB.Find the right most derivation for the string 0110.	
20	Construct a PDA that accepts the language generated by the grammar S → aSbb, S→ aab.	
	UNIT-III PART B	
1	Define Deterministic PDA.Is NPDA and DPDA equivalent? Illustrate with an eg.	13

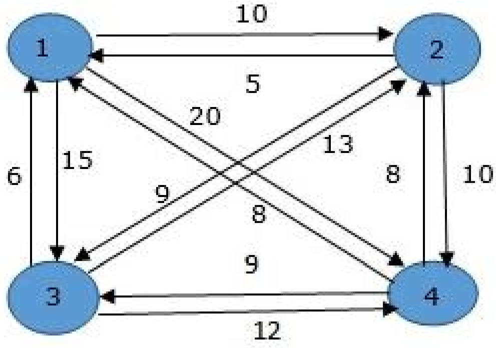
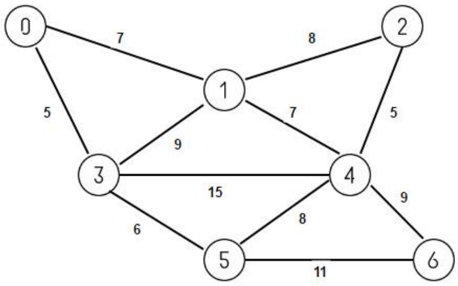
2	i) Let L is a context free language. Prove that there exists a PDA that accepts L. ii) Construct PDA for the Language L= $\{WW^R \mid W$ is in $(a+b)^*\}$.	13
3	i)Construct a transition table for PDA which accepts the language L={ $(a^{2n}b^n/n>0\}$ Trace your PDA for the input with n=3. ii)Find the PDA equivalent to the give CFG with the following productions.S→A, A→BC, B→ba, C→ac	13
4	Construct the PDA accepting the language L= { $a^nb^n/n>0\}$ by empty stack and final state.	13
5	Convert the grammar S→0S1/A: A→1A0/S/ ε into PDA that accepts the same language by empty stack. Check whether 0101 belongs to N(M).	13
6	Convert the grammar S→aSb/A, A→bSa/S/ε to PDA that accepts the same language byempty stack.	13
7	Construct a PDA for the language L={x€{a,b}*/ $n_a(x)>n_b(x)\}$	13
8	Design a PDA to accept $\{0^{2n}1^n\|n>1\}$. Draw the transition diagram for the PDA. Show byinstantaneous description that the PDA accepts the strings'0011'.	13
9	Construct the PDA accepting the language L= { $a^nb^nc^md^m$/m,n>0}	13
10	Construct the PDA accepting the language L= { $a^nb^mc^md^n$/m,n>0}	13
11	Construct the PDA accepting the language L= { $a^ib^jc^{i+j}$/ i,j>0}	13
12	Construct the PDA accepting the language L= { $a^ib^pc^i$/ i >0}	13
13	Construct a PDA for the grammar S→ aB/bA, A → a/aS/bAA, B→b/bS/aBB.	13
14	Let G be the grammar given by S →0BB,B→0S/1S/0. Construct a PDA and test whether it satisfies 010000 the language.	13
15	Construct a PDA equivalent to the following productions: S →aAA, A→aS/bS/a and hence whether the PDA accepts the string abaaaa.	13
	UNIT-III PART C	
1	Give CFG for the PDA, $M = (\{q_0,q_1\},\{0,1\},\{z_0,x\},\delta,q_0,z_0,\emptyset)$Where δ is given by $\delta(q_0,1,z_0) = (q_0,xz_0)$ $\delta(q_0,1,x) = (q_0,xx)$ $\delta(q_0,0,x) = (q_1,x)$ $\delta(q_0,\epsilon,z_0) = (q_0,\epsilon)$ $\delta(q_1,1,x) = (q_1,\epsilon)$ $\delta(q_1,0,z_0) = (q_0,z_0)$	15
2	A language is context free if and only if some PDA recognizes it.	15
3	Convert the grammar S→0S1/A, A→1A0/S/ϵ into an equivalent PDA and check whether 0101 belongs to it.	15

	UNIT IV NORMAL FORMS AND TURING MACHINES Normal forms for CFG – Simplification of CFG- Chomsky Normal Form (CNF) and Greibach Normal Form (GNF) – Pumping lemma for CFL – Closure properties of Context Free Languages –Turing Machine : Basic model – definition and representation – Instantaneous Description – Language acceptance by TM – TM as Computer of Integer functions – Programming techniques for Turing machines (subroutines).	
	UNIT IV - PART A	
1	What are the three ways to simplify a context free grammar?	
2	What are the closure properties of CFG?	
3	State the pumping lemma for CFL	
4	Give the steps to eliminate useless symbols.	
5	What is null production and unit production?	
6	Show that L={ap/ P is prime } is not context free.	
7	List all the closure properties of CFL.	
8	Define Turing Machine.	
9	What are the required fields of an instantaneous description or configuration of aTM?	
10	What is multiple tracks Turing machine?	
11	Explain the multi tape Turing Machine mode. Is it more power than the basic turing machine? Justify your answer.	
12	List out the different techniques for TM construction.	
13	What is Universal Turing machine?	
14	What are the differences between a Finite automata and a Turing machine?	
15	What is halting problem?	
16	Write short note on Chomskian hierarchy of languages.	
17	Define the language recognized by the TM.	
18	When do you say a TM is an algorithm?	
19	Define instantaneous description of a Turing Machine.	
20	Using Pumping lemma, show that the language L={ $a^n b^n c^n$|n≥1} is not a CFL.	
	UNIT IV - PART B	
1	Is the language L = {$a^n b^n c^n$ | n>=1} is context free? Justify.	13

2	Convert the following CFG into CNF for the following productions: S →ASA/aB, A → B/S,B →b/ ϵ	13
3	Construct a equivalent grammar G in CNF for the grammar G1=({S,A,B},{a,b},{S→ASB| ε, A→aAS|a, B→SbS|A|bb},S).	13
4	Begin with the grammar S→ 0A0 | 1B1 | BB, A → C,B →S/A, C→ S/ ϵ and simplify using safe order Eliminate ϵ production, unit production and useless symbol and put the grammar in Chomsky normal form.	13
5	Find an equivalent grammar in CNF for the grammar (Or) How a CFGf or L is converted into CNF accepting the same language? Convert the following CFG into CFG in CNF. S→bA|aB, A→bAA|aS|a, B→aBB|bS|b	13
6	Convert the following grammar G into Greibach Normal Form (GNF) S→XA|BB B→b|SB X→b A→a	13
7	Construct a TM for the language L={a^nb^n/n≥0}.Draw the transition diagram.(Also specify the instantaneous description to trace the string0011)	13
8	Construct a TM for the language L= {$a^nb^nc^n$ /n ≥0}.	13
9	Design a Turing machine which recognizes palindrome over alphabet {a,b}.Trace the strings “abab” and “baab”.	13
10	i) Design a Turing machine to compute multiplication of two positive integers. ii) Design a Turing machine to recognize the language{$0^n1^n0^n for\ n \geq 1$}	13
	UNIT IV - PART C	
11	Explain the variations of Turing machines. . Write about Multi Tape Turing machines.ExplainMultitapeandMultiheadTuringmachinewithsuitableexample.	15
12	What is the purpose of normalization? Construct the CNF and GNF for the following grammar and Explain the steps. S→ aAa | bBb | ε, A→ C |a B→C |b C→CDE |ε D→A |B | ab	15

13	Explain the various (programming) techniques for Turing machine construction.	15
14	Design a Turing machine to compute proper subtractions.	15
15	Construct a Turing machine for multiplying two non negative integers using subroutine.	15
	UNIT V UNDECIDABILITY Unsolvable Problems and Computable Functions –PCP-MPCP- Recursive and recursively enumerable languages – Properties - Universal Turing machine -Tractable and Intractable problems - P and NP completeness – Kruskal's algorithm – Travelling Salesman Problem-3-CNF SAT problems.	
	Unit V - Part A	
1	List the properties of recursive and recursively enumerable languages?	
2	Define diagnalization language Ld?	
3	What is primitive recursive function?	
4	Define recursive language.	
5	What is meant by RE languages? (Or) Define regular languages.	
6	When is a recursively enumerable language said to be recursive?	
7	Show that union of recursive language is recursive	
8	State when a problem is said to be decidable problem and give example.	
9	State when a problem is said to be undecidable problem and give example.	
10	Define intractable problem. Give example. (Or) Identify whether 'Tower of Hanoi' problem is tractable or intractable. Justify your answer.	
11	What is Universal language L_u?	
12	Define PCP.	
13	State RICE theorem.	
14	What is Universal Turing Machine?	
15	Define class P.	
16	Define tractable problem. Give example.	
17	Define class NP.	
18	State halting problem. Is halting problem decidable or undecidable problem	
19	Define Polynomial time reduction.	
20	Define NP-Hard. Define NP-Complete.	

	Unit V - Part B	
1	Describe about Recursive and RE languages in detail. Also discuss the properties of recursive and RE in detail.	13
2	Show that the union of two recursive language is recursive and union of two RE language is recursive.	13
3	State and prove halting problem	13
4	What is PCP? Explain with the help of an example.	13
5	Explain the universal Turing machine with its significance. Also explain the construction of Universal Turing machine with example..	13
6	Explain decidable and undecidable problem with example.	13
7	Explain tractable problem and intractable problem with example.	13
8	Explain class P, class NP, NP-complete and NP-hard problem with examples in detail.	13
9	Outline the concept of polynomial-time reductions.	13
10	Prove that Post Correspondence problem is undecidable.	13
11	Prove that Universal Language L_U is recursively enumerable but not recursive. (Or) Prove that Universal language is recursively enumerable but not recursive.	13
12	Define PCP and prove that PCP is undecidable.	13
13	Explain how to measure and classify complexity.	13
14	Consider two-tape Turing machine and determine whether the Turing machine always writes a nonblank symbol on its second tape during the computation on any input string. Formulate this problem as s language and show it is undecidable.	13
	UNIT –V PART C	
1	Find the minimal spanning tree using Kruskal's algorithm with all necessary steps  c 7 d 7 2 a 4 4 e 6 10 b 2 f	8

<table>
<tr><td>2</td><td>Solve the following travelling salesman problem with all necessary steps and explanations

</td><td>8</td></tr>
<tr><td>3</td><td>Find the minimal spanning tree using Kruskal’s algorithm with all necessary steps

</td><td>8</td></tr>
<tr><td>4</td><td>State whether the instances of the post correspondence problem (PCP) have a solution. The following are the instances with ∑ = {0,1}. In case the PCP has a solution, describe the post – correspondence solution with justification.
<table><tr><td>Index</td><td>List A</td><td>List B</td></tr><tr><td>1</td><td>B</td><td>CA</td></tr><tr><td>2</td><td>A</td><td>AB</td></tr><tr><td>3</td><td>CA</td><td>A</td></tr><tr><td>4</td><td>ABC</td><td>C</td></tr></table></td><td>8</td></tr>
<tr><td>5</td><td>State whether the instances of the post correspondence problem (PCP) have a solution. The following are the instances with ∑ = {0,1}. In case the PCP has a solution, describe the post – correspondence solution with justification.
<table><tr><td>Index</td><td>List A</td><td>List B</td></tr><tr><td>1</td><td>10</td><td>01</td></tr><tr><td>2</td><td>110</td><td>011</td></tr><tr><td>3</td><td>110</td><td>01</td></tr><tr><td>4</td><td>000</td><td>00</td></tr><tr><td>5</td><td>10</td><td>010</td></tr></table></td><td>8</td></tr>
</table>