Register No: | | | | | | | | | |

Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam – 603 110

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

B.E. / B.Tech. End Semester Theory Examinations, Nov / Dec 2021

Fifth Semester

Computer Science and Engineering

UCS1524 LOGIC PROGRAMMING

(Regulations 2018)

Time: **Three Hours** **Answer ALL Questions** Maximum:**100 Marks**

K1: Remembering K2: Understanding K3: Applying K4 :Analyzing K5: Evaluating

## PART – A (10 × 2 = 20 Marks)

| | | | |
|---|---|---|---|
| 01. | K3 | Using truth table, show that the formula ¬(A→B)^(B→A) is satisfiable but not valid. | CO1 |
| 02. | K3 | Consider the formula ¬P(k) ^ ∀x Q(x) ∧ ∀y P(y) ∧ ∃z ¬ Q(z). Identify whether the formula has the following properties (answer yes/no).<br>a. Rectified<br>b. Prenex<br>c. Skolem<br>d. Clausal form | CO1 |
| 03. | K2 | Compare procedural semantics and declarative semantics. | CO2 |
| 04. | K3 | Applying model theoretic semantics, find the Herbrand base for the following facts:<br>likes(adam, chocolates)<br>likes(adam, cake)<br>likes(vishnu, chocolates)<br>likes(vishnu, cake) | CO2 |
| 05. | K3 | Consider the following Prolog code.<br>fun1([], [] ).<br>fun1([First \| Rest], Result) :- fun1( Rest, TempResult), append(TempResult, [First], Result).<br>append( [], L, L).<br>append ( [X\| L1], L2, [X\| L3]) :- append ( L1, L2, L3).<br>Make use of the above code and print the output for the goal: ?- fun1([1,2,3],R). | CO3 |
| 06. | K1 | List the data objects supported by Prolog. Give examples. | CO3 |
| 07. | K3 | Applying the rules for concatenation given below, find the value of L using the goal:<br>?- conc( [_, _, _], L, [1,2,3,4,5,6]).<br>conc( [], L, L).<br>conc( [X\| L1], L2, [X\| L3]) :- conc( L1, L2, L3). | CO4 |
| 08. | K3 | Let a search tree be represented using the following facts where s(x,y) means x is the successor of y.<br>goal(j).<br>s(b,d).<br>s(b,e).<br>s(d,h).<br>s(e,i).<br>s(e,j).<br>Using the following Prolog program, predict the result for the goal: ?:-solve(b, S).<br>solve( N, [N] ) :- goal( N). | CO4 |

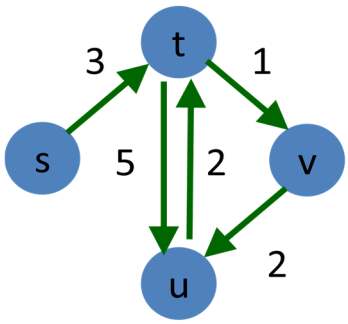
| | | | |
|---|---|---|---|
| | | solve( N, [N \| Sol1] ) :- s( N, N1), solve( N1, Sol1). | |
| 09. | K1 | List the important features to be considered when you define the rules to represent knowledge. | CO5 |
| 10. | K1 | Define shell in an expert system. | CO5 |

## PART – B (5 × 6 = 30 Marks)

| | | | |
|---|---|---|---|
| 11. | K3 | Construct the conjunctive and disjunctive normal forms of the formula:<br>¬ ((¬ A→ ¬ B ) ^ ¬ C) | CO1 |
| 12. | K2 | Explain answer generation in logic programming with an example. | CO2 |
| 13. | K3 | Consider a one way road connecting 6 places as below.<br>Tambaram --> Chrompet --> Pallavaram --> Meenambakkam --> Guindy --> Adayar<br>A person can travel from one place to another only in forward direction.<br><br>Develop a recursive Prolog program to check whether she can travel on that road.<br>Trace the output for the following goals:-<br>a. can_travel(Chrompet,Guindy)<br>b. can_travel(Adayar,Pallavaram)<br>c. can_travel(Tambaram,Chrompet)<br><br>Extend the above program with a suitable rule such that the person can travel also in reverse direction. Trace the output for the above goals. | CO3 |
| 14. | K2 | Explain problem reduction strategy using AND/OR graph with DFS search. | CO4 |
| 15. | K3 | Maintain a knowledge base for automatic hotel booking. Each hotel can be described using the following details: Hotel_Name, City, Star, Room_Type, Room_Rate.<br><br>Develop an expert system using Prolog with two Askables namely City and Room_Rate and perform the following.<br>a. Write the rules in Prolog to create the knowledge base with 5 instances.<br>b. Write the rules for Room_Search with required Askables and Star as an argument to display the Hotel_Name and Room_Type.<br>c. Test for any 2 goals. | CO5 |

## PART – C (5 × 10 = 50 Marks)

| | | | |
|---|---|---|---|
| 16. | K3 | Consider the following FOL formulas.<br>a. ∃x Puppy(x) ^ Has (Mini, x)<br>b. ∀x (∃y Puppy(y) ^ Has(x,y)) -> Zoophilist(x)<br>c. ∀x Zoophilist (x) -> ∀y Animal(y) -> ¬Hurts(x,y)<br>d. Hurts (Mini, Rini) V Hurts (Bunny, Rini)<br>e. Rabit(Rini)<br>f. ∀x Rabit (x) -> Animal (x)<br><br>Construct clausal forms for the above formulas and show that the inference of the clause, Hurts(Bunny, Rini) using the logical formulas follows N-resolution. | CO1 |
| OR | | | |

| | | | |
|---|---|---|---|
| 17. | K3 | Apply resolution to show "Philip is smart" using the following sentences. (Hint: convert the sentences into clausal forms)<br>a. Anyone passing data structures exam and wins the coding competition is smart.<br>b. Anyone who studies or is intelligent can pass all his exams.<br>c. Philip did not study but he is intelligent.<br>d. Anyone who is intelligent wins the coding competition. | CO1 |
| | | | |
| 18. | K2 | Explain procedural interpretation with an example. | CO2 |
| | | OR | |
| 19. | K2 | Explain Horn clause program logic with example. | CO2 |
| | | | |
| 20. | K2 | Explain any five list operations with examples. | CO3 |
| | | OR | |
| 21. | K2 | Explain the Prolog rules for any five polygons with examples. | CO3 |
| | | | |
| 22. | K3 | Let the set of students in Class A be {tom, john, mary, jim, brute} and in Class B {anu, preeth, banu, vimal}. Develop a Prolog program to combine these students into Class C and display the students in alphabetical order. | CO4 |
| | | OR | |
| 23. | K3 | Develop a Prolog program to find the shortest path between the nodes: 's' and 'u' in the following graph.<br>  | CO4 |
| | | | |
| 24. | K2 | Explain the structure of an expert system and elaborate the knowledge representation for medical domain. | CO5 |
| | | OR | |
| 25. | K2 | Show the rules and user interactions for an automatic music player using Prolog code and explain with appropriate goals. | CO5 |

CO1: Apply propositional logic and predicate logic for knowledge representation. (K3)
CO2: Apply different types of semantics for logic programming. (K3)
CO3: Develop programs in PROLOG. (K3)
CO4: Solve AI problems using search algorithms in PROLOG. (K3)
CO5: Develop a simple Expert system shell in PROLOG. (K3).