****https://lh4.googleusercontent.com/4T9vR6okYDxkIfm9nwzmuWpgrZBspUWcA4RNELKNVTzbGF4RdfpmFx6Rz6hMi3BY9i40EcFhgHPY5M10p8xAJBduLVRZxgkRS-6xUlXwTgpleZRdCA0GmFEt5Ku6YrG0IO2009MZ **SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**RAMAPURAM CAMPUS**

**FACULTY OF ENGINEERING AND TECHNOLOGY**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**CONTINUOUS LEARNING ASSESSMENT-II**

**ANSWER KEY**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sub Code/Name** | **18CSC305J–Artificial Intelligence** | **Set** | **ODD** |
| **Year/Sem/Branch** | **III Year / VI / B.Tech-CSE and its Specialization** | **Date** | **20-03-23** |
| **Max.Marks** | **50** | **Duration** | **90 Mins** |

**PART A (10x1= 10)**

**ANSWER ALL THE QUESTIONS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Q.No.** | **Question** | **Marks** | **CO** | **BL** | **PI** |
| 1 | Choose the data structure used in standard implementation of Breadth First Search is? a) Stack **b) Queue** c) Linked List d) Tree | 1 | 2 | 1 | 1.3.1 |
| 2 | A person wants to visit some places. He starts from a vertex and then wants to visit every place connected to this vertex and so on. What algorithm he should use? a) Depth First Search **b) Breadth First Search** c) Trim’s algorithm d) Kruskal’s algorithm | 1 | 2 | 2 | 1.2.1 |
| 3 | What is the space complexity of Depth-first search? where b is the branching factor and m is the maximum depth of the search tree. a) O(b) b) O(bl) c) O(m) **d) O(bm)** | 1 | 2 | 1 | 1.2.1 |
| 4 | Which search implements stack operation for searching the states? a) Depth-limited search **b) Depth-first search** c) Breadth-first search d) Best-first search | 1 | 2 | 1 | 2.2.3 |
| 5 | Which is used to compute the truth of any sentence? **a) Semantics of propositional logic** b) Alpha-beta pruning c) First-order logic d) Sematic net | 1 | 2 | 1 | 2.2.3 |
| 6 | Translate the following statement into FOL. “For every a, if a is a philosopher, then a is a scholar” **a) ∀ a philosopher(a) scholar(a)** b) ∃ a philosopher(a) scholar(a) c) ∃ a philosopher(a) scholar(a) d) ∀ a philosopher(a) ∃ scholar(a) | 1 | 3 | 2 | 1.2.1 |
| 7 | Which system is used to demonstrate, on a purely syntactic basis, that one formula is a logical consequence of another formula.  **a) Deductive Systems** b) Inductive Systems c) Reasoning with Knowledge Based Systems d) Search Based Systems | 1 | 3 | 3 | 1.3.1 |
| 8 | The adjective “first-order” distinguishes first-order logic from \_\_\_\_\_\_\_\_\_\_\_ in which there are predicates having predicates or functions as arguments are permitted.  a) Representational Verification b) Representational Adequacy **c) Higher Order Logic** d) Inferential Efficiency | 1 | 3 | 2 | 1.2.1 |
| 9 | Show the representation of Fuzzy logic a) IF-THEN-ELSE rules **b) IF-THEN rules** c) Both IF-THEN-ELSE rules & IF-THEN rules d) None of the mentioned | 1 | 3 | 2 | 1.2.1 |
| 10 | Infer the following algorithm learns from more complex environments to generalize, approximate and simplify solution logic. a) Fuzzy Relational DB b) Ecorithms **c) Fuzzy Set** d) None of the mentioned | 1 | 3 | 3 | 1.3.1 |

**PART-B (4x4= 16)**

**ANSWER ANY FOUR QUESTIONS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Q.No.** | **Question** | **Marks** | **CO** | **BL** | **PI** |
| 11. | Illustrate the heuristic estimation function for A\* search?  Heuristics, sometimes called heuristic functions, are used to provide 'good enough' solutions to very complex problems where finding a perfect solution would take too much time. When you use heuristics, you trade accuracy, correctness, and exactness for speedy processing.  One of the drawbacks with Dijkstra's algorithm is that it can (and will) evaluate paths that will never provide the shortest option. Imagine tring to find the shortest route on a map between London and Edinburgh. Anyone with a reasonable grasp of UK geography, would not bother to evaluate a route that went via Plymouth. The trade off between speed and accuracy is important. In some applications, accuracy is less important than computational time. For example, in a SatNav application a route that is calculated in seconds and is "short enough" is preferable to having to wait 10 minutes for the perfect route.  The A\* algorithm uses a heuristic function to help decide which path to follow next. The heuristic function provides an **estimate** of the minimum cost between a given node and the target node. The algorithm will combine the **actual cost from the start node** - referred to as g(n) - with the **estimated cost to the target node** - referred to as h(n) - and uses the result to select the next node to evaluate. This is explained in more detail in the step-by-step method that follows. Choosing a heuristic function There is no single best heuristic to use in path finding, as every application is different. For example, if the cost relates to a distance, it could be estimated by calculating the "straight line" distance between the nodes, perhaps by using one of the following methods:   * [Euclidian distance](https://en.wikipedia.org/wiki/Euclidean_distance#:~:text=In%20mathematics%2C%20the%20Euclidean%20distance,metric%20as%20the%20Pythagorean%20metric.) * [Manhattan distance](https://en.wikipedia.org/wiki/Taxicab_geometry) * [Great circle distance](https://en.wikipedia.org/wiki/Great-circle_distance)   However, do remember that the weights on a graph do not always represent distance.  It is very important that the heuristic function does **not overestimate** costs. However, so long as the heuristic function provides an estimate that is less than or equal to the actual cost, A\* will always find an optimal path and will generally find it much faster than Dijkstra’s algorithm. | 4 | 2 | 2 | 1.3.1 |
| 12. | Identify the termination condition for the Hill-Climbing algorithm.   * **Local maximum:**It is a state which is better than its neighboring state however there exists a state which is better than it(global maximum). This state is better because here the value of the objective function is higher than its neighbors. * **Global maximum:**It is the best possible state in the state space diagram. This is because, at this stage, the objective function has the highest value. * **Plateau/flat local maximum:**It is a flat region of state space where neighboring states have the same value. * **Ridge:**It is a region that is higher than its neighbors but itself has a slope. It is a special kind of local maximum. * **Current state:**The region of the state space diagram where we are currently present during the search. * **Shoulder:**It is a plateau that has an uphill edge. | 4 | 2 | 2 | 1.3.1 |
| 13. | Develop the min-max algorithm for the given example and find the optimal path for MIN to win the game    C:\Users\HP\Downloads\WhatsApp Image 2023-04-03 at 2.37.41 PM.jpeg | 4 | 2 | 2 | 2.1.1 |
| 14. | Utilize the fact which you represent in propositional and predicate logic with an example.  **Proposition** logic  A **proposition** is basically a declarative sentence that has a truth value. Truth value can either be true or false, but it needs to be assigned any of the two values and not be ambiguous. The purpose of using propositional logic is to analyze a statement, individually or compositely.  Explanation must be provided with example propositional sentence  **Predicate Logic :**  Predicates are properties, additional information to better express the subject of the sentence. A quantified predicate is a proposition , that is, when you assign values to a predicate with variables it can be made a proposition.  Explanation must be provided with example predicate logic sentence | 4 | 3 | 2 | 2.1.1 |
| 15. | Construct first order logic for the following English statements:   1. Every boy or girl is a child   forall x ((boy(x) or girl(x)) -> child(x))   1. Every child gets a doll or a train or a lump of coal   forall y (child(y) -> (gets(y,doll) or gets(y,train) or gets(y,coal))) | 4 | 3 | 3 | 2.1.1 |
| 16. | Identify objects, properties, functions and relations for the given example.  “EVIL KING JOHN BROTHER OF RICHARD RULED ENGLAND IN 1200”  Objects : John, Richard, England, 1200 Relation : Ruled Properties : Evil, King Functions : BROTHER OF | 4 | 3 | 1 | 2.1.1 |

**PART-C (2x12= 24)**

**ANSWER ALL THE QUESTIONS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Q.No.** | **Question** | **Marks** | **CO** | **BL** | **PI** |
| 17. | Draw the State Space diagram for Hill Climbing search problem. Identify the problems in different regions in Hill climbing and give reason.    Hill climbing is a simple optimization algorithm used in Artificial Intelligence (AI) to find the best possible solution for a given problem. It belongs to the family of local search algorithms and is often used in optimization problems where the goal is to find the best solution from a set of possible solutions.  In Hill Climbing, the algorithm starts with an initial solution and then iteratively makes small changes to it in order to improve the solution. These changes are based on a heuristic function that evaluates the quality of the solution. The algorithm continues to make these small changes until it reaches a local maximum, meaning that no further improvement can be made with the current set of moves.  There are several variations of Hill Climbing, including steepest ascent Hill Climbing, first-choice Hill Climbing, and simulated annealing. In steepest ascent Hill Climbing, the algorithm evaluates all the possible moves from the current solution and selects the one that leads to the best improvement. In first-choice Hill Climbing, the algorithm randomly selects a move and accepts it if it leads to an improvement, regardless of whether it is the best move. Simulated annealing is a probabilistic variation of Hill Climbing that allows the algorithm to occasionally accept worse moves in order to avoid getting stuck in local maxima.  Hill Climbing can be useful in a variety of optimization problems, such as scheduling, route planning, and resource allocation. However, it has some limitations, such as the tendency to get stuck in local maxima and the lack of diversity in the search space. Therefore, it is often combined with other optimization techniques, such as genetic algorithms or simulated annealing, to overcome these limitations and improve the search results.  **Advantages of Hill Climbing algorithm:**  Hill Climbing is a simple and intuitive algorithm that is easy to understand and implement.  It can be used in a wide variety of optimization problems, including those with a large search space and complex constraints.  Hill Climbing is often very efficient in finding local optima, making it a good choice for problems where a good solution is needed quickly.  The algorithm can be easily modified and extended to include additional heuristics or constraints.  OR  Describe the Alpha-Beta pruning algorithm with example and how the Shortfalls of Min-max algorithm is eliminated by using Alpha -Beta pruning.    Alpha-beta pruning is a modified version of the minimax algorithm. It is an optimization technique for the minimax algorithm.  As we have seen in the minimax search algorithm that the number of game states it has to examine are exponential in depth of the tree. Since we cannot eliminate the exponent, but we can cut it to half. Hence there is a technique by which without checking each node of the game tree we can compute the correct minimax decision, and this technique is called **pruning**. This involves two threshold parameter Alpha and beta for future expansion, so it is called **alpha-beta pruning**. It is also called as **Alpha-Beta Algorithm**.  Alpha-beta pruning can be applied at any depth of a tree, and sometimes it not only prune the tree leaves but also entire sub-tree.  The two-parameter can be defined as:   * 1. **Alpha:** The best (highest-value) choice we have found so far at any point along the path of Maximizer. The initial value of alpha is **-∞**.   2. **Beta:** The best (lowest-value) choice we have found so far at any point along the path of Minimizer. The initial value of beta is **+∞**.   The Alpha-beta pruning to a standard minimax algorithm returns the same move as the standard algorithm does, but it removes all the nodes which are not really affecting the final decision but making algorithm slow. Hence by pruning these nodes, it makes the algorithm fast.  The Max player will only update the value of alpha.  The Min player will only update the value of beta.  While backtracking the tree, the node values will be passed to upper nodes instead of values of alpha and beta.  We will only pass the alpha, beta values to the child nodes. | 12 | 2 | 1 | 1.3.1 |
|  | 12 | 2 | 1.3.1 |
| 18. | Apply how the forward chaining algorithm work for the following problem:  The law says that it is a crime for an American to sell weapons to hostile nations. The country Nono, an enemy America, has some missiles, and all of its missiles were sold to it by Col. West, who is an American.  Prove that Col. West is a criminal  Facts Conversion into FOL:  It is a crime for an American to sell weapons to hostile nations. (Let's say p, q, and r are variables) **American (p) ∧ weapon(q) ∧ sells (p, q, r) ∧ hostile(r) → Criminal(p)       ...(1)**  Country A has some missiles. **?p Owns(A, p) ∧ Missile(p)**. It can be written in two definite clauses by using Existential Instantiation, introducing new Constant T1. **Owns(A, T1)             ......(2)** **Missile(T1)             .......(3)**  All of the missiles were sold to country A by Robert. **?p Missiles(p) ∧ Owns (A, p) → Sells (Robert, p, A)       ......(4)**  Missiles are weapons. **Missile(p) → Weapons (p)             .......(5)**  Enemy of America is known as hostile. **Enemy(p, America) →Hostile(p)             ........(6)**  Country A is an enemy of America. **Enemy (A, America)             .........(7)**  Robert is American **American(Robert).             ..........(8)**    OR | 12 | 3 | 3 | 2.1.1 |
| Explore the use of predicate logic as a way of representing knowledge by looking at a specific example and prove that Marcus hate Caesar by Consider the following set of sentence for converting in to CNF.  1. Marcus was a man.  2. Marcus was a Pompeian.  3. All Pompeians were Romans.  4. Caesar was a ruler.  5. All Pompeians were either loyal to Caesar or hated him.  6. Every one is loyal to someone.  7. People only try to assassinate rulers they are not loyal to.  8. Marcus tried to assassinate Caesar. | 12 | 3 | 1.2.1 |

Prepared by: HOD