|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | USN |  |  |  |  |  |  |  |  |  |  | | | | | | |  | | | | | | |
| Internal Assessment Test 1 – September 2018 | | | | | | | | | | | | |
| Sub: | Artificial Intelligence | | | | | Sub Code: | 15CS562 | Branch: | | CSE | | |
| Date: | 10/09/2018 AN | Duration: | 90 min’s | Max Marks: | 50 | Sem/Sec: | 5/ A,B And C | | | | OBE | |
| Answer any FIVE FULL Questions | | | | | | | | | MARKS | | CO | RBT |
| 1 (a) | Define Artificial Intelligence. List two real world examples.   * The study of how to make computers do things at which at the moment, people are better. * A field of study that seeks to explain and emulate intelligent behaviorin terms of computational processes. * Eg: Automatic door opening systems, Driverless cars | | | | | | | | [04] | | CO1 | L1 |
| (b) | Explain any two algorithms to solve the TIC-TAC-TOE problem. Interpret your results by tracing the algorithm step-by-step.  Data Structure:Nine element vector,   * 0-Blank * 1-X * 2-O * Move table: 3^9 entries   Algorithm 1:   1. View the vector board as ternary number (base 3). Convert to decimal. 2. Use the number from previous step as index into move table. 3. Set the board equal to that vector.   Algorithm 2:   1. See if it is a win, if so, call it the best by giving it the highest possible rating 2. Otherwise, consider all the moves the opponent could make next. See which is worst for us. Assume the opponent will make that move. Whatever rating that move has, assign it to the node we are considering. 3. The best node is then the ones with the highest rating. | | | | | | | | [06] | | CO1 | L2 |
| 2 (a) | Identify a good state space representation for the Tower of Hanoi problem.  State space: x – 0, 1, 2, …, n for n disks  Start state: (n, 0, 0)  Goal state: (0, 0, n)  Assumptions:  1. Only one disk can be moved at a time  2. At any state, on all pegs, the disks should be arranged in ascending order of their size from top to bottom.  3. Only the top disk can be removed.  Moves:  1. Move n-1 disks from source to aux  2. Move nth disk from source to dest  3. Move n-1 disks from aux to dest | | | | | | | | [04] | | CO1 | L2 |
| (b) | Construct the production rules and solve the Tower of Hanoi problem.  Production Rules:  1. Move n-1 disks from source to aux (n,0,0) → (1, n-1,0)  2. Move nth disk from source to dest (1, n-1, 0) → (0, n-1, 1)  3. Move n-1 disks from aux to dest (0, n-1, 1) → (0, 0, n)  Solution:  Let’s take a situation where n=3  1. (3, 0, 0) → (2, 0, 1) move 1 disk from source to dest  2. (2, 0, 1) → (1, 1, 1) move 1 disk from source to aux  3. (1, 1, 1) → (1, 2, 0) move 1 disk from dest to aux  4. (1, 2, 0) → (0, 2, 1) move 1 disk from source to dest  5. (0, 2, 1) → (1, 1, 1) move 1 disk from source to aux  6. (1, 1, 1) → (1, 0, 2) move 1 disk from aux to dest  7. (1, 0, 2) → (0, 0, 3) move 1 disk from source to dest | | | | | | | | [06] | | CO2 | L2 |
| 3 (a) | Identify the situation where the concept of heuristics is applied and explain about Constraint Satisfaction Problem with an example.  Why Heuristics?  Heuristic search is done,   * When a solution is required which might not be the best solution. * Need of a satisfier than an optimizer. * Compromise the requirement of mobility and systematicity. * Control structure that no longer guaranteed to find the best answer.   Constraint Satisfaction Problem:   * Constraint Satisfaction problems in AI have goal of discovering some problem state that satisfies a given set of constraints. * Design tasks can be viewed as constraint satisfaction problems in which a design must be created within fixed limits on time, cost, and materials.   Algorithm:   1. Propagate available constraints. To do this first set OPEN to set of all objects that must have values assigned to them in a complete solution. Then do until an inconsistency is detected or until OPEN is empty:    1. Select an object OB from OPEN. Strengthen as much as possible the set of constraints that apply to OB.    2. If this set is different from the set that was assigned the last time OB was examined or if this is the first time OB has been examined, then add to OPEN all objects that share any constraints with OB.    3. Remove OB from OPEN. 2. If the union of the constraints discovered above defines a solution, then quit and report the solution. 3. If the union of the constraints discovered above defines a contradiction, then return the failure. 4. If neither of the above occurs, then it is necessary to make a guess at something in order to proceed. To do this loop until a solution is found or all possible solutions have been eliminated:   a. Select an object whose value is not yet determined and select a way of strengthening the constraints on that object.  b. Recursively invoke constraint satisfaction with the current set of constraints augmented by strengthening constraint just selected.  Eg :   1. Graph Coloring 2. Cryptanalysis | | | | | | | | [06] | | CO1 | L2 |
| (b) | Explain the Breadth-First Search and demonstrate how the state space tree for the water-jug problem is parsed.   1. Create a variable called NODE-LIST and set it to the initial state. 2. Until a goal state is found or NODE\_LIST is empty.    1. Remove the first element from NODE-LIST and call it E. If NODE-LIST was empty, quit.    2. For each way that each rule can match the state described in E do:       1. Apply the rule to generate a new state.       2. If the new state is a goal state, quit and return this state.       3. Otherwise, add the new state to the end of NODE-LIST.   New Doc 2018-09-14 10.22.17.jpg | | | | | | | | [04] | | CO2 | L2 |
| 4 (a) | Interpret a heuristic function for travelling salesman problem.  TSP using nearest neighbor heuristic:   1. Arbitrarily select a starting city 2. Look at all the cities not yet visited, and select the one that is closest to the current city. 3. Repeat step 2 until all cities are visited. | | | | | | | | [04] | | CO2 | L2 |
| (b) | Explain the working Depth-First Search with an example and demonstrate how it is different from best first and breadth first search.   1. If the initial state is a goal state, quit and return success 2. Otherwise, do the following until success or failure is signaled:    1. Generate a successor, E, of the initial state. If there are no more successors, signal failure.    2. Call Depth-First Search with E as the initial state    3. If success is returned, signal success. Otherwise continue in this loop.   BFS-vs.-DFS.jpgmaxresdefault.jpg | | | | | | | | [06] | | CO2 | L2 |
| 5 (a) | Differentiate between OR and AND-OR graphs with an example.   |  |  | | --- | --- | | OR | AND-OR | | * Heuristic Function : f’=g+h’ * Maintains two lists OPEN and CLOSED * Any example graph to be solved with the give heuristic function. | * Heuristic Function : f’=h’ * Uses a single structure G * Any example graph to be solved with the give heuristic function. | | | | | | | | | [04] | | CO2 | L2 |
| (b) | Construct the production rules and solve the Water-Jug problem.  Production rule:      Solution : | | | | | | | | [06] | | CO2 | L2 |
| 6 (a) | Explain about hill climbing algorithm.   * Is a variant of generate-and test in which feedback from the test procedure is used to help the generator decide which direction to move in searchspace. * The test function is augmented with a heuristic function that provides an estimate of how close a given state is to the goalstate. * Computation of heuristic function can be done with negligible amount of computation. * Hill climbing is often used when a good heuristic function is available for evaluating states but when no other useful knowledge isavailable. * O/P of test function : Heuristic Function-distance from current state to desired state. | | | | | | | | [04] | | CO2 | L2 |
| (b) | Summarize the variations of hill climbing algorithm.  Simple Hill Climbing Algorithm:  1.Evaluatetheinitialstate.Ifitisalsogoalstate,thenreturnitandquit.Otherwisecontinuewiththeinitialstateasthecurrentstate.  2.Loop until a solution is found or until there are no new operators left to be applied in the currentstate:  a.Select an operator that has not yet been applied to the current state and apply it to produce a newstate  b.Evaluate the newstate  i.If it is the goal state, then return it andquit.  ii.If it is not a goal state but it is better than the current state, then make it the current state.  iii.If it is not better than the current state, then continue in theloop.  Steepest Ascent Hill Climbing:   1. Evaluate the initial state. If it is also a goal state, then return it and quit. Otherwise, continue with initial state as current state. 2. Loop until a solution is found or until a complete iteration produces no change to current state:  * Let SUCC be a state such that any possible successor of the current state will be better than SUCC * For each operator that applies to current state do: * Apply the operator and generate a new state * Evaluate the new state. If it is a goal state, then return it and quit. If not compare it to SUCC. If it is better, then set SUCC to this state. If it is not better, leave SUCC alone. * If the SUCC is better than current state then set current state to SUCC | | | | | | | | [06] | | CO2 | L2 |
| 7 (a) | Describe about the four knowledge representation approaches.  Four Knowledge Representation Approaches:   * Simple relational knowledge: Simplest way to represent declarative facts is as set of relations of the same sort used in database systems. ο Each fact about a set of objects is set out systematically in columns * Inheritable knowledge: Knowledge is made up of objects consisting of ν attributes and ν corresponding associated values. Boxed nodes: objects and values of attributes of objects. Values can be objects with attributes and so on.Arrows: point from object to its value. ο This structure is known as a slot and filler structure, semantic network or a collection of frames. ο Property inheritance : Elements of specific class inherits attributes and values of general class * Inferential knowledge: Represent knowledge as formal logic. * Advantages: A set of strict rules, Can be used to derive more facts, Truths of new statements can be verified, guaranteed correctness, E.g.: 1. “Marcus is a man” 2. “All men are mortal” ν Implies: 3. “Marcus is mortal” * Procedural knowledge: Representation of “how to make it” rather than “what it is”. May have inferential efficiency, but no inferential adequacy and acquisitional efficiency. Ex. Writing LISP programs i.e. artificial intelligence list processing. | | | | | | | | [06] | | CO2 | L2 |
| (b) | Illustrate the mapping between facts and representations. | | | | | | | | [04] | | CO2 | L2 |