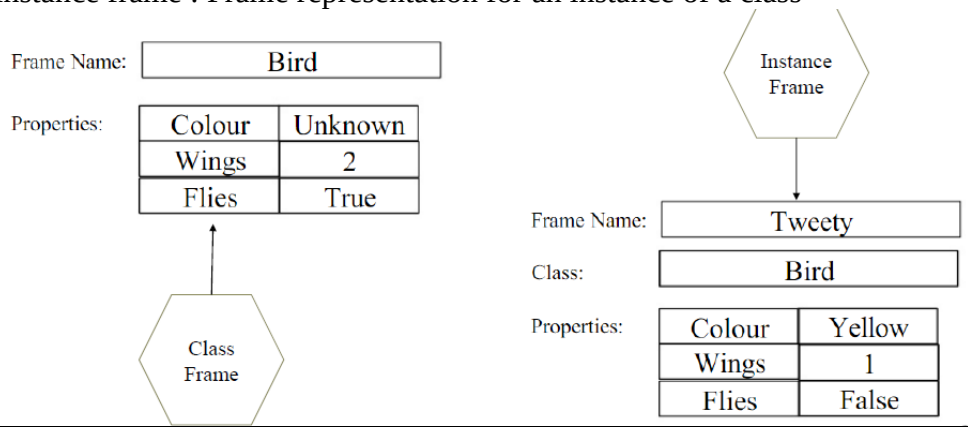
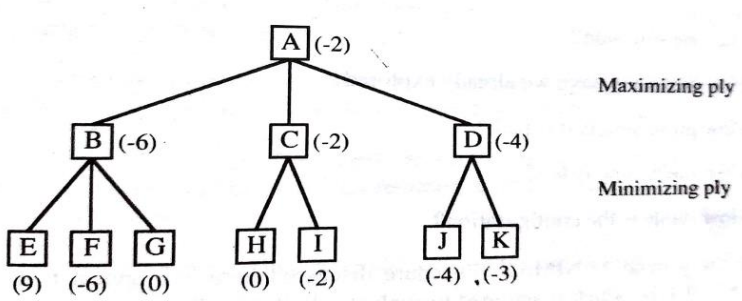
USN										

Internal Assessment Test 3 – November 2018

Sub:	Artificial Intelligence					Sub Code:	15CS562	Branch:	CSE	
Date:	22/11/2018 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 frame and differentiate class frame from instance frame with an example Frame is a Collection of attributes (called as slots) and associated values (called as fillers) (Including constraints) that describes an entity in real world. Class frame : Frame representation for a class Instance frame : Frame representation for an instance of a class 	4	CO2	L2
(b)	Explain minimax search procedure with an algorithm and example on both maximizing and minimizing levels. • Depth first and depth limited search. • Idea is to start at current position and use plausible move generator to generate the set of possible successor positions. • Now apply static evaluation function to choose the best position to move such that we maximize our possibility of win thereby minimizing the opponents' chance of win ▫ Player 1: Maximize the chance ▫ Player 2: Minimize the chance  • A straight forward recursive procedure that relies on 2 auxiliary procedures • MOVEGEN(Position, Player) –Plausible move generator which returns a list of nodes representing the moves that can be made by player in position. • STATIC(Position, Player) –The static evaluation function, which returns a number representing the goodness of position from the stand point of player	6	CO2	L2

Algorithm: MINIMAX(Position, Depth, Player

1. If DEEP-ENOUGH(*Position*, *Depth*), then return the structure
   VALUE = STATIC(*Position*, *Player*);
   PATH = nil

   This indicates that there is no path from this node and that its value is that determined by the static evaluation function.

2. Otherwise, generate one more ply of the tree by calling the function MOVE-GEN(Position Player) and setting SUCCESSORS to the list it returns.

3. If SUCCESSORS is empty, then there are no moves to be made, so return the same structure that would have been returned if DEEP-ENOUGH had returned true.

4. If SUCCESSORS is not empty, then examine each element in turn and keep track of the best one. This is done as follows.

   Initialize BEST-SCORE to the minimum value that STATIC can return. It will be updated to reflect the best score that can be achieved by an element of SUCCES-SORS.

For each element SUCC of SUCCESSORS, do the following:

(a) Set RESULT-SUCC to
   MINIMAX(SUCC, *Depth* + 1, OPPOSITE(*Player*))[3]
   This recursive call to MINIMAX will actually carry out the exploration of SUCC.

(b) Set NEW-VALUE to −VALUE(RESULT-SUCC). This will cause it to reflect the merits of the position from the opposite perspective from that of the next lower level.

(c) If NEW-VALUE > BEST-SCORE, then we have found a successor that is better than any that have been examined so far. Record this by doing the following:

   i. Set BEST-SCORE to NEW-VALUE.

   ii. The best known path is now from CURRENT to SUCC and then on to the appropriate path down from SUCC as determined by the recursive call to MINIMAX. So set BEST-PATH to the result of attaching SUCC to the front of PATH(RESULT-SUCC).

2 (a)	Explain how the property inheritance algorithm resolves tangled hierarchies with an example? • Hierarchies that are not simple trees are called tangled hierarchies. o These allow another type of inheritance conflict. • Inferential distance: "Node1 is closer to Node2 than Node3 if and only if Node1 has an inference path through Node2 to Node3, i.e. Node2 is in between Node1 and Node3."	6	CO2	L2

To retrieve a value *V* for slot *S* of an instance *F* do:

1. Set *CANDIDATES* to empty.

2. Do breadth-first or depth-first search up the *isa* hierarchy from *F*, following all *instance* and *isa* links. At each step, see if a value for *S* or one of its generalizations is stored.

   (a) If a value is found, add it to *CANDIDATES* and terminate that branch of the search.

   (b) If no value is found but there are *instance* or *isa* links upward, follow them.

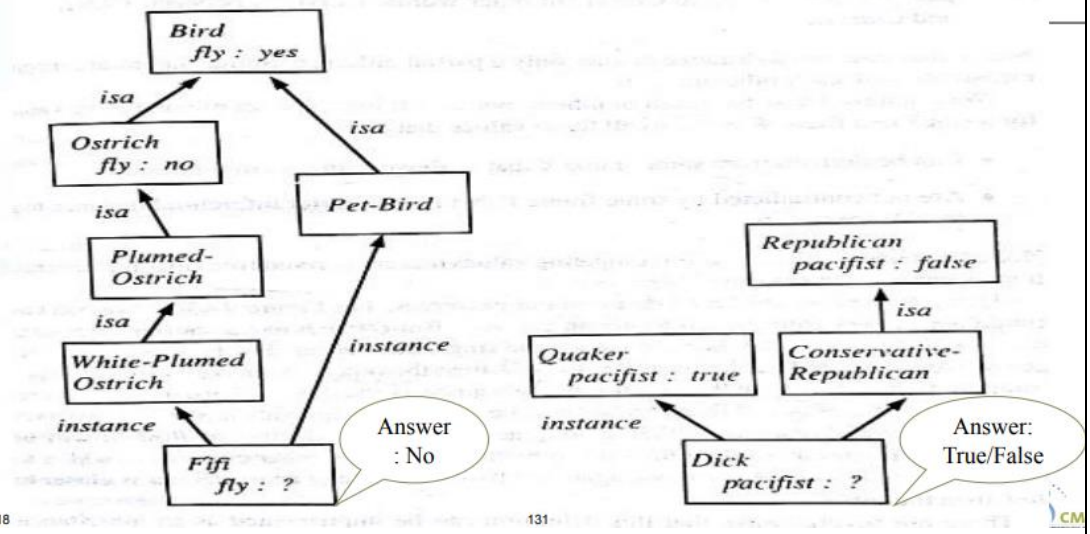
   (c) Otherwise, terminate the branch.

3. For each element *C* of *CANDIDATES* do:

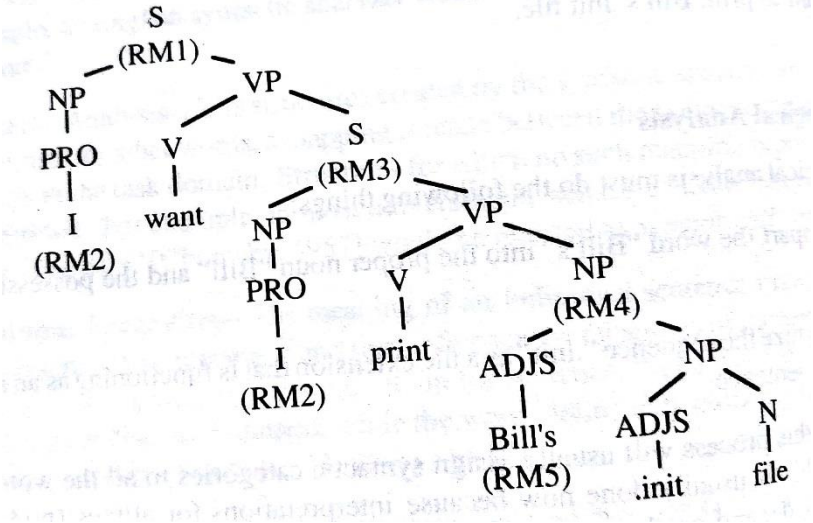
   (a) See if there is any other element of *CANDIDATES* that was derived from a class closer to *F* than the class from which *C* came.

(b) If there is, then, remove *C* from *CANDIDATES*.

4. Check the cardinality of *CANDIDATES*:
   (a) If it is 0, then report that no value was found.
   (b) If it is 1, then return the single element of *CANDIDATES* as *V*.
   (c) If it is greater than 1, report a contradiction.



(b)	Demonstrate any 4 dependency rules of Conceptual Dependency (CD). □ Rule1: Describes the relationship between an actor and event he/she causes. This is a two way dependency since neither actor nor event can be primary. □ PP ⟷ ACT □ E.g.: John $\overset{p}{\longleftrightarrow}$ PTRANS □ P indicates past tense □ English version : John ran Rule 2: Describes the relationship between a PP and a PA that is being asserted to describe it. Many state descriptions like height are represented in CD as numeric scales. PP ⟺ PA John ⟺ height (> average) John is tall Rule 3: describes the relationship between two PPs, one of which belongs to the set defined by other. Relationship between 2 objects PP ⟺ PA John ⟺ doctor John is a doctor	4	CO2	L2

	Rule 4 : describes the relationship between a PP and an attribute that has already been predicated of it. Direction of arrow is towards the PP being described. PA→PP E.g.: nice→boy English Sentence : A nice boy Difference from rule 2 : this is general case while rule 2 is about a particular person john			
3	Illustrate the steps involved in Natural Language Processing with an example. • **Morphological Analysis** – Individual words are analyzed into their components, and non-word tokens, such as punctuation, are separated from the words. o An example: I want to print Bill's .init file. o Bill's --> Bill (the proper noun) + 's (the possessive suffix) o Recognize the sequence ".init" as a file extension. • In addition, this process will usually assign syntactic categories to all the words in the sentence. E.g. verbs, Nouns, Singular, plural etc., • **Syntactic Analysis (Parsing)** – Linear sequences of words are transformed into structures that show how the words relate to each other • Syntactic analysis must exploit the results of morphological analysis to build a structural description of the sentence. • I.e. arrange words in a proper manner such that proper relationship is established between words • Parse tree is generated representing the relationship between words • Reference markers: Set of entities represented in parenthesis.  • **Semantic Analysis** – does two things o It must map individual words into appropriate objects in the knowledge base or database. o It must create the correct structures to correspond to the way the meaning of the individual words combine each other o Partial meaning constructed is,	10	CO2	L3

	RM1 {the whole sentence} *instance :* *Wanting* *agent :* *RM2* {I} *object :* *RM3* {a printing event} *RM2* {I} *RM3* {a printing event} *instance :* *Printing* *agent :* *RM2* {I} *object :* *RM4* {Bill's .init file} *RM4* {Bill's .init file} *instance :* *File-Struct* *extension :* .init *owner :* *RM5* {Bill} *RM5* {Bill} *instance :* *Person* *first-name :* Bill • **Discourse Integration** – The meaning of an individual sentence may depend on the sentences that proceed it and may influence the meanings of the sentences that follow it. • We do not know to whom the pronoun "I" or the proper noun "Bill" refers from the previous three steps. To pin down these references requires an appeal to a model of the current discourses context, from which we can learn that the current user is *User068* and that the only person named "Bill" is *User073.* • **Pragmatic Analysis** – The structure representing what was said is reinterpreted to determine what was actually meant (intended meaning). • The final step toward effective understanding is to decide what to do as a result. • In this case, a translation from the knowledge-based representation to a command to be executed by the system has to be done.			

4	Construct a script for bank robbery scene mentioned below. "There happened a robbery in a bank where criminal had hold up the bank using a gun and had shot policeman there and escaped with money and jewels through bike." Props: • Gun G, Loot L, Bag B, Get away car C Roles: • Robber S, Cashier M, Bank Manager O, Policeman P Entry Conditions: • S is poor Results: • S has more money • O is angry • M is in a state of shock • P is shot	10	CO2	L3

Script: ROBBERY — *Track: Successful Snatch*

Props:
G = Gun,
L = Loot,
B= Bag,
C = Get away car.

Roles:
R = Robber,
M = Cashier,
O = Bank Manager,
P = Policeman.

Entry Conditions:
R is poor.
R is destitute.

Results:
R has more money.
O is angry.
M is in a state of shock.
P is shot.

Scene 1: Getting a gun

R PTRANS R into Gun Shop
R MBUILD R choice of G
R MTRANS choice.
R ATRANS buys G

(go to scene 2)

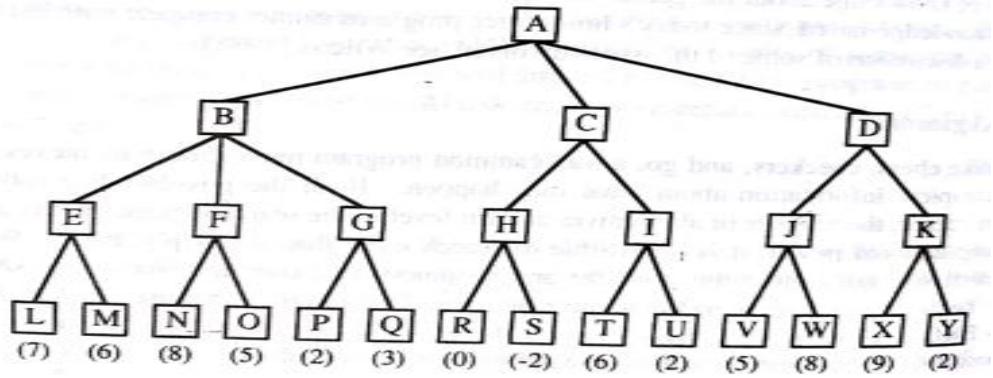
Scene 2 Holding up the bank

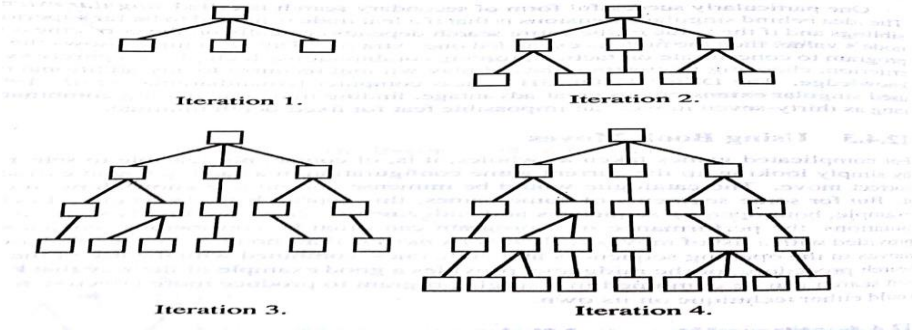
R PTRANS R into bank
R ATTEND eyes M, O and P
R MOVE R to M position
R GRASP G
R MOVE G to point to M
R MTRANS "Give me the money or ELSE" to M
P MTRANS "Hold it Hands Up" to R
R PROPEL shoots G
P INGEST bullet from G
M ATRANS L to M
M ATRANS L puts in bag, B
M PTRANS exit
O ATRANS raises the alarm

(go to scene 3)

Scene 3: The getaway

M PTRANS C

5 (a)	Consider the following game tree in which static scores are all from the first players' point of view:  i) Suppose the first player is the maximizing player, what move should be chosen? Move chosen would be D because of the value 8 from W ii) Apply alpha beta pruning procedure for the game tree given above and find the nodes that will not be examined in search.	7	CO2	L3

	I,O,T,U and Y			
(b)	Why the search in game playing programs always proceed forward from initial state rather than backward from a goal state? Infer the reason with an example of any game of your choice. Because games will have less number of initial state or even one compared to final states. The search has to be from less number of nodes to high because of which it has to be forward. E.g. Chess only one initial state and more final states	3	CO2	L2
6 (a)	Discuss about Iterative deepening and identify how it enhances Depth First Search and A* search. • Rather than searching to a fixed depth in a game tree , the same is searched as follows • Single ply first, apply static evaluation function , find possible moves • Initiate minimax again to the depth of two-ply, followed by three-ply, 4-ply and etc., • i.e. On each iteration the tree is searched one level deeper  **Algorithm: Depth-First Iterative Deepening** 1. Set SEARCH-DEPTH = 1. 2. Conduct a depth-first search to a depth of SEARCH-DEPTH. If a solution path is found, then return it. 3. Otherwise, increment SEARCH-DEPTH by 1 and go to step 2. • Advantages : DFID finds shortest solution path ***Algorithm: Iterative-Deepening-A**** 1. *Set THRESHOLD* = the heuristic evaluation of the start state. 2. *Conduct a depth-first search, pruning any branch when its total cost function* ($g + h'$) *exceeds THRESHOLD.*[4] *If a solution path is found during the search, return it.* 3. *Otherwise, increment THRESHOLD by the minimum amount it was exceeded during the previous step, and then go to Step 2.* • Guaranteed to find optimal solution provided that h' is an admissible heuristic • Efficient with respect to space	6	CO2	L2
(b)	Explain about the various learning techniques. 4 General category • Learning by practicing E.g.: Playing tennis, Riding etc., • Learning by experience	4	CO3	L2

	E.g.: Knowledge acquisition based on the previous tasks • Learning by taking advice • Learning from examples			