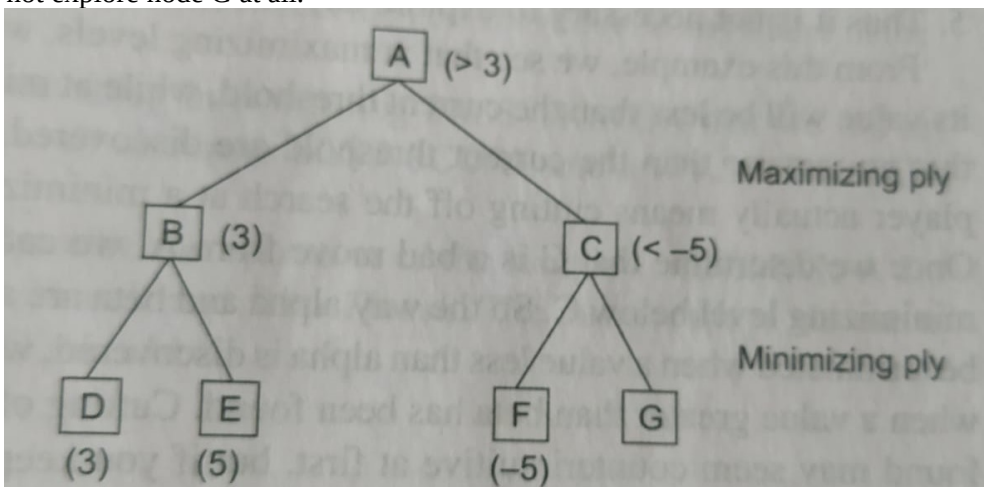
USN | | | | | | | | | | |

Internal Assessment Test 5 – Feb 2022

Sub:	Introduction to Artificial Intelligence					Sub Code:	18CS753	Branch:	ECE/EEE/ME/CIV
Date:	/02/22	Duration:	90 mins	Max Marks:	50	Sem / Sec:	VII	OBE	

Answer any FIVE FULL Questions

		MARKS	CO	RBT
1.	Explain the MINIMAX(position, depth, players) algorithm. • MINIMAX needs to return 2 results: 1. The backed-up value of the path it chooses. 2. The path itself. • MINIMAX returns a structure containing both results & the two functions, VALUE and PATH extract separate components. • Initially it takes 3 parameters, a board position, the current depth of search and the player to move. • The initial call to compute best move from CURRENT position is MINIMAX(CURRENT, 0, PLAYER-ONE) if PLAYER-ONE is to move, or MINIMAX(CURRENT, 0, PLAYER-TWO) if PLAYER-TWO is to move. ***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 SUCCESSORS. For each element SUCC of SUCCESSORS, do the following: (a) Set RESULT-SUCC to MINIMAX(SUCC, *Depth* + 1, OPPOSITE(*Player*)) } recursive call. 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). 5. Now that all the successors have been examined, we know the value of Position as well as which path to take from it. So return the structure VALUE = BEST-SCORE PATH = BEST-PATH When the initial call to MINIMAX returns, the best move from CURRENT is the first element on PATH	[10]	CO2	L2
2.	Explain alpha – beta pruning with an example. Minimax procedure follows depth first. Alpha-beta pruning is a branch and bound technique using which the efficiency of minimax procedure can be improved in which partial solutions that are worse than the new solutions can be abandoned early.	[10]	CO2	L2

	With reference to the figure, after examining node F, the opponent is guaranteed a score of -5 or less at node C. We also know that we are guarnateed a score of 3 or more at node A, which can be achieved on moving to B, After examining only F, we are sure that a move to C is worse regardless of the score of G. Thus we need not explore node G at all.  			
3.	Explain 5 phases of natural language understanding process. Morphological Analysis Syntactic Analysis Semantic Analysis Discourse Integration Pragmatic Analysis	[10]	CO2	L1
4a.	What are the causes and types of spelling errors? Typographic Errors Orthographic Errors Phonetic Errors	[05]	CO2	L1
4b.	Explain the techniques applied for spell checking. Non-word Error Detection Isolated-word error Detection – Minimum Edit Distance Context dependent error detection and correction Soundex Algorithm	[05]	CO3	L1
5	Differentiate between transformational analogy and derivational analogy. Transform a solution to previous problem as a solution to a current problem. Transormational analogy do not look at how the problem was solved. It only looks at the final soluion. Often, the twists and turns that are involved in solving an old problem is relevant to solving a new problem. The history of problem solving is called derivation. The analytical reasoning that takes into account these derivations is called derivational analogy. Add examples and shematic diagram.	[05]	CO3	L2
6	How is knowledge acquisition performed to build an expert system? To develop a knowledge acquisition system, a knowledge engineer interviews a domain expert knowledge, which is then translated into rules. After the initial system is built, it must be iteratively refined until it approximates expert-level performance. Since this process is expensive and time consuming, automatic	[05]	CO4	L2

	knowledge acquisition systems were attempted. Such a program is expected to perform following activities: 1) Entering knowledge 2) Maintaining the knowledge base consistency. 3) Ensuring knowledge base completeness. The proble to be solved using the expert systes can be either diagnosis or design. In the case of diagnosis, an expert system accepts input data, comes out with a set of candidate diagnosis, then uses differtiating knowledge to determine which one is the best – cover and differentiate technique. In the case of design task, it is impossible to pre-enumerate all solutions. Values are assigned to all the parameters in such a way that they are consistent with each other and the design satisfies the external constraints such as cost. Propose and revise is a technique used to buld expert systems to solve design tasks. Mention on MOLE and SALT.			