32002: AI (Adversarial Search) **Spring 2014**

Lecture 33: April 03-10, 2014

Instructor: K.R. Chowdhary *: Professor of CS*

33.1 Search and Game Playing

The *combinatorial Game theory* (CGT) studies strategies and mathematics of two-palyer games of *perfect knowledge* such as *chess* or *go*. But generally concentrates on simpler games like *nim* or solving *end-games* or their special cases. An important difference between this subject and *classical game theory* (a branch of economics) is that game players are assumed to move in sequence rather than simultaneously, so their is no point in randomization or information hiding strategies.

The CGT does not study with importance in computer knowledge, called games of chance, like poker (card games). The combinatorial games include, games like chess, chekers, Go, Arimaa, Hex, and Connect6. They also include one player combinatorial puzzles, and even no-player automata.

In CGT, moves are represented as game-tree, and the trees can be searched while making moves in the game. A game-trees is a special kind of semantic tree where nodes represent board configurations and branches indicate how one board configuration can be transformed into another configuration by a single move. The decisions regarding the next move are made by two *adversaries*, who play the game.

In game-playing computer programs, improved results can be obtained by searching the tree only to some severely limited depth, using a static evaluation function to estimate the utility values of the nodes at that depth; and then proceeding in the usual manner to compute utility values for the shallower nodes in the tree as if the estimated utility values were in fact correct. This technique, which we call heuristic game tree searching, is used implicitly in "real life" decision situations, although it does not seem to have been studied by decision analysts.

33.2 Game Playing Strategy

Let there be a facility by which every unique board configuration can be converted into a unique single, overall quality value in the form of an integer number. Further, consider that positive number indicate the move in favour of one player, and negative - for the favour of his/her opponent. The degree of favouredness is absolute value of that number. Let us call ourself as *maximizer* player, and the opponent as *minimizer*. Also, let us imagine that, there is a number (call it static *score*, accumulated so far), the maximizer will make a move such that the score increases to maximum, while the opponent (adversary) makes a move so that by addition of quality number, the score number minimizes.

The difference between the games and search problems is that the games playing are highly unpredictable, as one does not know what move the opponent is going to make. And, only based on the opponent's move the other player has to make a move. The other difference is that the time limits for a move are in realistic times, hence one is unlikely to find goal in that time, therefore approximations are necessary for both the

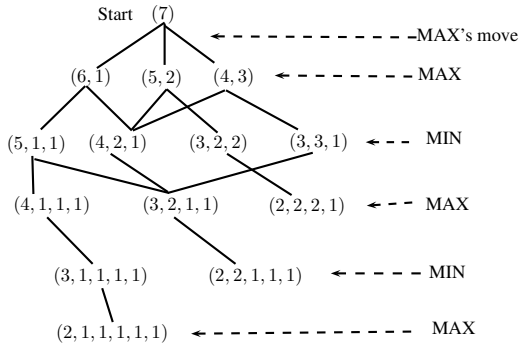

Figure 33.1: MINIMAX Game-Tree.

players. The formal system behind these games is called the *Game-Theory.*

Example 33.1 *Grundy's game.*

Consider a typical player game, called *Grundy's* game, where there is stack of seven coins of equal size, indicated by initial configuration: (7). The staring player MAX makes a move so that the stack is broken into two stacks of unequal size. Figure 33.1 shows the alternate moves as : $(6,1),(5,2),(4,3)$. The other player, MIN, has next move, which breaks the sub-stacks created further into unequal parts. This process goes on for alternate players. The player who first cannot play is game loser.

We note that for the node $(2,2,1,1,1)$ MAX is not able to make next move, hence MAX loses. The player that makes first move is generally MAX player. Thus after this move, the node positions occupied is called MAX nodes, and from these nodes the move is due to MIN player. This leads the transition to MIN-MAX nodes.

At every configuration, there are choices available either for MAX or MIN player for the next move. For example, MAX will try to foresee the possible next moves available to MIN after he has made a move, and accordingly chooses the next move so the MIN cannot have a good move, and he is closer to a winning strategy. The MIN also keeps the similar strategy. Finally, the game ends at a node where no further move is possible by any of the player. □

33.2.0.1 Complexities

The network of configurations and move (links) comes up as a search-tree, called *game-tree.* The start node is at level 0, and other nodes levels from top are at levels $1, 2, ..., d$ for a game-tree of depth d. If b is maximum branching factor, the worst case time and space complexities for breadth-first search (BFS) are both $O(b^d)$. For depth-first search (DFS), time and space complexities are $O(b^d), O(b * d)$, respectively.

References

[1] Chowdhary K.R. (2020) Logic and Reasoning Patterns. In: Fundamentals of Artificial Intelligence. Springer, New Delhi. https://doi.org/10.1007/978-81-322-3972-7_11