Business Intelligence WS 04/05

Assignment #2 - Search in Games & Knowledge Representation

Part 1: Two-Player Game

a) The game tree, complete with annotations of all minimax values, is shown in Figure 1.

b) The "?" values are handled by assuming that an agent with a choice between winning the game and entering a "?" state will always choose the win. That is, min(-1, ?) is -1 and max(+1, ?) is +1. If all successors are "?", the backed-up value is "?".

c) Standard minimax is depth-first and would go into an infinite loop. It can be fixed by comparing the current state against the stack; and if the state is repeated, then return a "?" value. Propagation of "?" values is handled as above. Although it works in this case, it does not always work because it is not clear how to compare "?" with a drawn position; nor is it clear how to handle the comparison when there are wins of different degrees (as in backgammon). Finally, in games with chance nodes, it is unclear how to computer the average of a number and a "?". Note that it is not correct to treat repeated states automatically as drawn positions; in this example, both (1, 6) and (2, 6) repeat in the tree but they are won positions. What is really happening is that each state has a well-defined but initially unknown value. These unknown values are related by the minimax equation at the bottom of page 163. If the game tree is acyclic, then the minimax algorithm solves these equations by propagating from the leaves. If the game tree has cycles, then a dynamic programming method must be used, as explained in Chapter 17. (Exercise 17.8 studies this problem in particular.) These algorithms can determine weather each node has a well-determined value (as in this example) or is really an infinite loop in that both players prefer to stay in the loop (or have no choice). In such a case, the rules of the game will need to define the value (otherwise the game will never end). In chess, for example, a state that occurs 3 times (and hence is assumed to be desirable for both players) is a draw.

d) This question is a little tricky. One approach is a proof by induction on the size of the game. Clearly, the base case n = 3 is a loss for A and the base case n = 4 is a win for A. For any N > 4, the initial moves are the same: A and B both move one step towards each other. Now, we can see that they are engaged in a subgame of size n-2 on the squares {2, …, n-1}, expect that there is an extra choice of moves on squares 2 and n-1. Ignoring this for a moment, it is clear that if the "n-2" is won for A, then A gets to the square n-1 before B gets to 1, hence the "n" game is won for A. By the same line of reasoning, if "n-2" is won for B then "n" is won for B. Now, the presence of the extra moves complicates the issue, but not too much. First, the player who is slated to win the subgame {2, …, n-1} never moves back to his home square. If the player slated to lose the subgame does so, then it is easy to show that he is bound to lose the game itself – the other player simply moves forward and a subgame of size n-2k is played one step closer to the loser's home square.

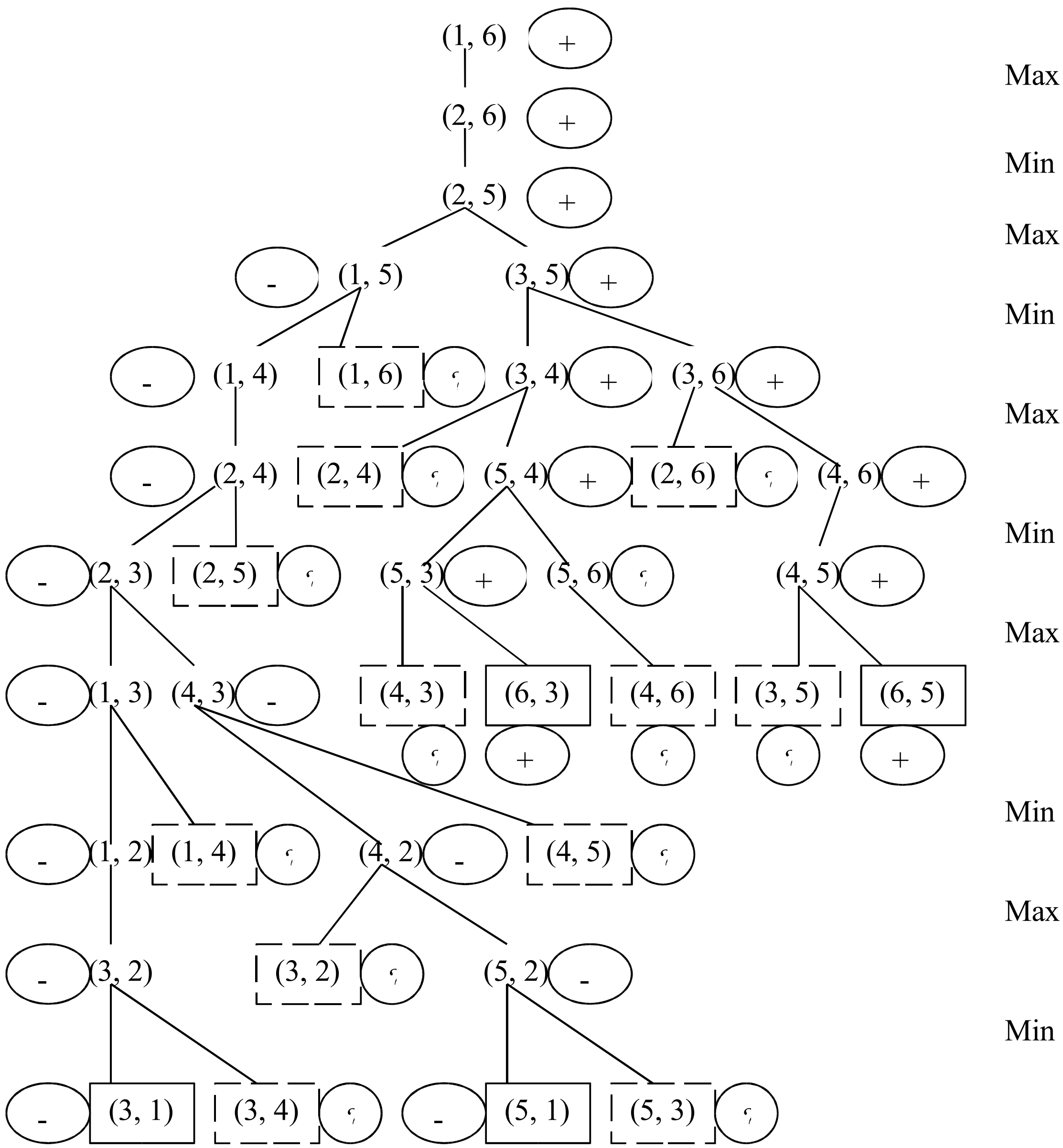

Figure 1