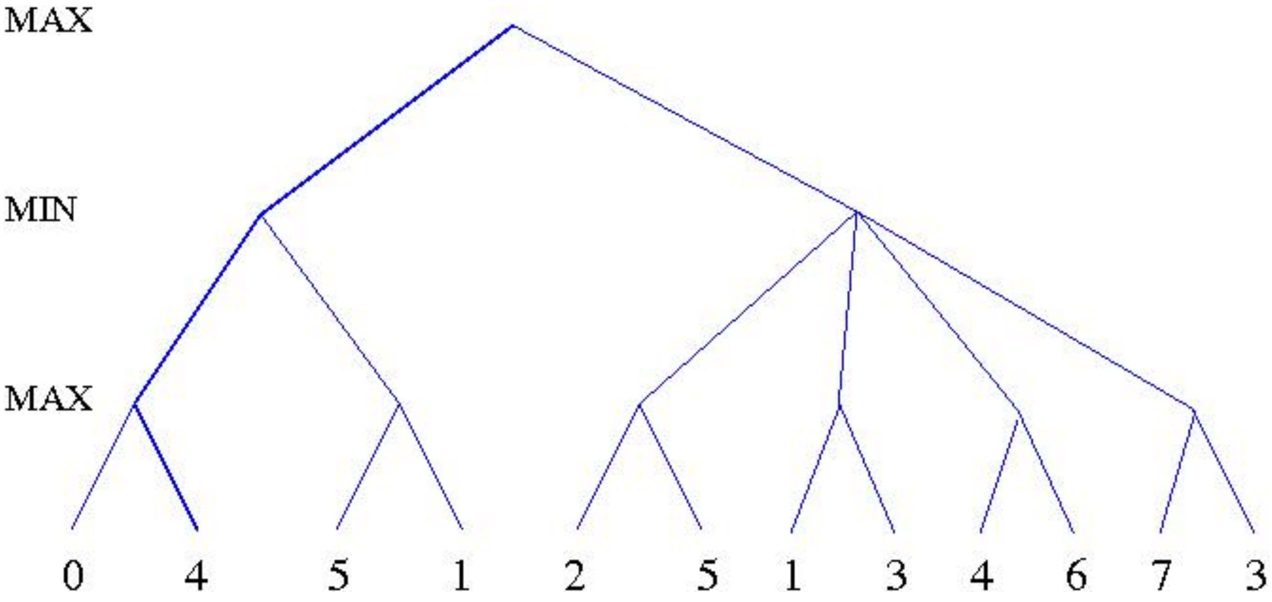
COMP3411/9814 Artificial Intelligence Term 1, 2024

Tutorial - Week 4

This page was last updated: 02/29/2024 19:26:06

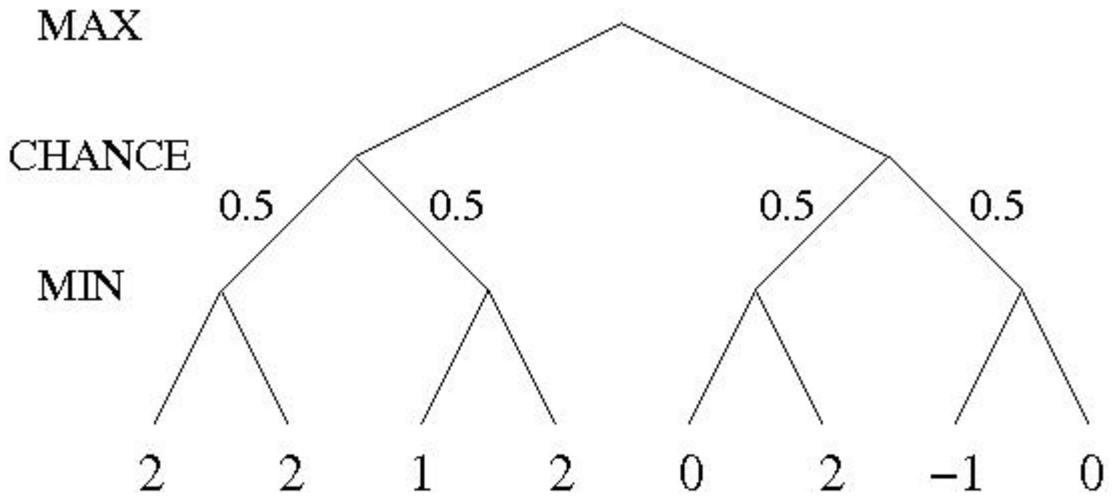
1. **Alpha-Beta Search**

   Apply the alpha-beta search algorithm to the following game tree, indicating clearly the values of alpha and beta at each node as the algorithm progresses, and circling the nodes that would not be evaluated.



2. **Pruning in Games with Chance Nodes**

   This question considers pruning in games with chance nodes.
   This figure shows the complete game tree for a very simple such game.
   Assume that the leaf nodes are to be evaluated in left-to-right order, and that before a leaf node is evaluated, we know nothing about its value – the range of possible values is -∞ to ∞.



a. Copy the figure, mark the value of all internal nodes, and indicate the best move at the root with an arrow.

b. Given the values of the first six leaves, do we need to evaluate the seventh and eighth leaf? Given the values of the first seven leaves, do we need to evaluate the eighth leaf? Explain your answers.

c. Suppose the leaf node values are known to lie between -2 and 2 inclusive. After the first two leaves are evaluated, what is the value range for the left-hand chance node?

d. How many (and which) leaves need to be evaluated under the assumption in Part (c)?

3. **Reactive Agents**

   Check that you understand the Identifying Braitenberg Vehicles activity on the Reactive Agents page in OpenLearning (Module 2).

Additional Questions and Discussion (if time permits)

4. **Nim**

   Describe an optimal strategy for a simple version of the game Nim, which uses only a single heap of n stones. Players take turns to remove either 1, 2 or 3 stones from the heap. The player who takes the last stone (or, all the remaining stones) wins. For what values of n can the first player force a win? For what values can the second player force a win?

5. **Other Simple Games**

   Discuss what you found out from the Tree Search for Simple Games activity on the Alpha-Beta Pruning page, concerning one or more of these games:

   - Hexapawn on a 3x3 board
   - Connect-4, or a simplifed version with four columns, where you only need to

get three in a row in order to win

- Sprouts, with two initial dots
- another simple game of your choosing

6. **Endgame Databases**

The Chinook checkers program makes extensive use of endgame databases, which provide exact values for every position with eight or fewer pieces.
How might such databases be efficiently generated, stored and accessed?