Lecture 4: Minimax, Alpha-Beta pruning

AI For Traditional Games
Prof. Nathan Sturtevant
Winter 2013

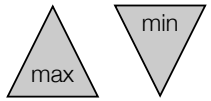
UNIVERSITY OF DENVER

"Solving a game"

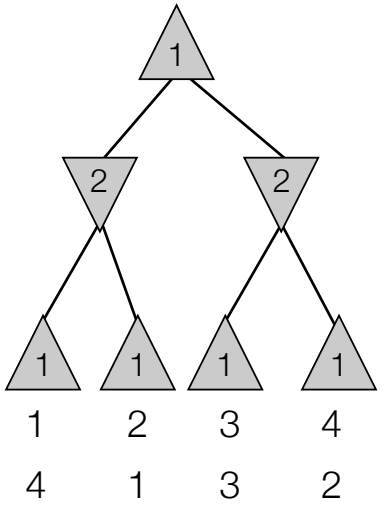
- Classify nodes as AND/OR
  - AND/OR tree
  - Max player needs any of the children to be a win (OR)
  - Min player needs all of the children to be a win (AND)
    - Win for the max player, that is
- Only works in small games
- Bottom up example from Nim
  - Start with 7 in the stack
  - time: $O(b^d)$ space: $O(b^d)$

Minimax

- Min player and Max player
  - Max player tries to find strategy for maximum score
  - Min player tries to find strategy for minimum score
- Depth-First Search
  - Find minimax value of each state recursively



Example



UNIVERSITY OF DENVER

Minimax Pseudo-Code

```
Minimax()
    GetMaxVal()

GetMaxVal()
    if (game over) return game value
    currVal ← -∞
    for each successor s in 1… # successors
        ApplyMove(s)
        currVal = max(currVal, GetMinVal())
        UndoMove(s)
    return currVal
```

UNIVERSITY OF DENVER

Minimax

- Analysis
  - O(d) memory
  - O(N) = O(b^d) time
- What if we don't have time to do the whole tree?

UNIVERSITY OF DENVER

Minimax Pseudo-Code (2)

```
Minimax(depth)
    GetMaxVal(depth)

GetMaxVal(depth)
    if (depth == 0) return CutoffEval();
    currVal ← -∞
    for each successor s in 1… # successors
        ApplyMove(s)
        currVal = max(currVal, GetMinVal(depth-1))
        UndoMove(s)
    return currVal
```

UNIVERSITY OF DENVER

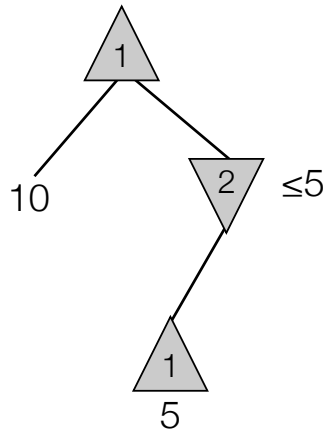
How should we choose our eval?

- Only need an ordering on the preference of leaf values
- But, often normalize values
  - Choose set of useful features & weights
  - $f_1 \cdot w_1 + f_2 \cdot w_2 + \ldots + f_n \cdot w_n$
  - In chess, just material value of pieces plus a good search will play somewhat reasonable chess

UNIVERSITY OF DENVER

Can we do better?

- Don't have to search a whole tree to know the value of the tree
  - Simple example:



UNIVERSITY OF DENVER

Alpha Beta Pruning

- alpha is the best score achieved by the max player
  - alpha starts at -∞
- beta is the best score achieved by the min player
  - beta starts at ∞
- If alpha >= beta, then we can perform a cutoff