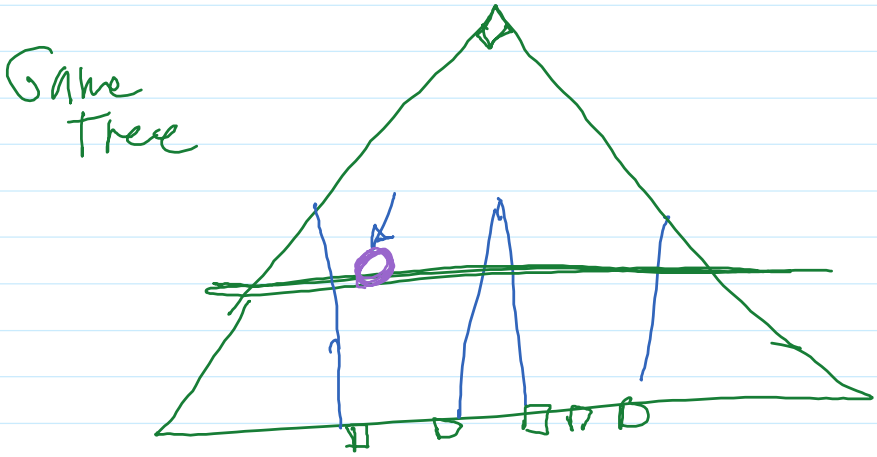
Depth-Limited Alpha-Beta Pruning

Thursday, April 9, 2020 11:47 PM



Heuristic Evaluation function, HEval(s) :- (s: state)

- an aproximation of the minimax value of s

HMinimax(s,d) {
- HEval(s) if Cutoff-Test(s,d)
- max a∈Actions(s) HMinimax(Result(s,a),d+1) if MAX = player(s)
- min a∈Actions(s) HMinimax(Result(s,a),d+1) if MIN = player(s)

}

- Qualities of a good HEval() function.-
  - should order the terminal nodes in the same way as the utility function
  - quick to compute
  - correlate non-terminal states with "chance of winning"

- Using features of state for HEval()

Chess - pawn = 1, rook = 5, bishop = 3, queen = 9
- "good pawn structure" +1/2
- "king safety" + 1/2

$$HEval(s) = w_1 f_1(s) + w_2 f_1(s) + w_3 f_1(s) \ldots \sum_i w_i f_i(s)$$

assumes independence among the features,

but ignores things like
- pawn position
- value difference at end game
- blocked pieces

Algorithm for :
Heuristic Depth-Limited Minimax with Alpha-Beta Pruning

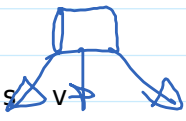
```
(* Heuristic Depth Limited Minimax with Alpha-Beta Pruning *)

PROCEDURE cuttoffTest( s : state, d )
RETURN ( term(s) OR d = depth_limit )


PROCEDURE AlphaBetaSearch( s0 : state )
  v := maxValue( s0, 0, -∞, +∞ )
  besta := action for which minimax value of result( s0, a ) equals v
RETURN besta;


PROCEDURE maxValue( s : state, d, α , β )
  IF cutoffTest( s, d ) THEN RETURN HEval( s )
  v := -∞
  FOREACH a in actions( s ) DO
    v := MAX( v, minValue( result( s, a), d+1, α , β )
    if v >= β THEN RETURN v
    α := MAX(α,v)
  RETURN v

PROCEDURE minValue( s : state, d, α , β )
  IF cutoffTest( s ) THEN RETURN HEval( s )
```



```
IF cutoffTest( s ) THEN RETURN HEval( s )
v := ∞
FOREACH a in actions( s ) DO
  v := MIN( v, maxValue( result( s, a), d+1, α , β )
  if v <= α THEN RETURN v
  β := MIN(β,v)
RETURN v
```