Artificial Intelligence
Game Playing

Nilsson - Chapter 12

Russell and Norvig - Chapter 6

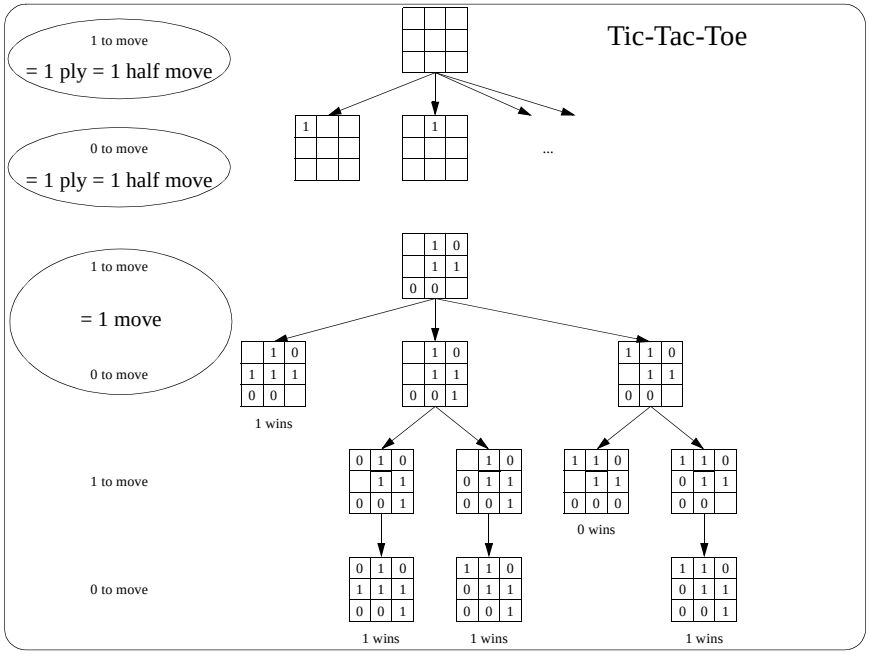
May 1997
Deep Blue - Garry Kasparov
3.5 - 2.5

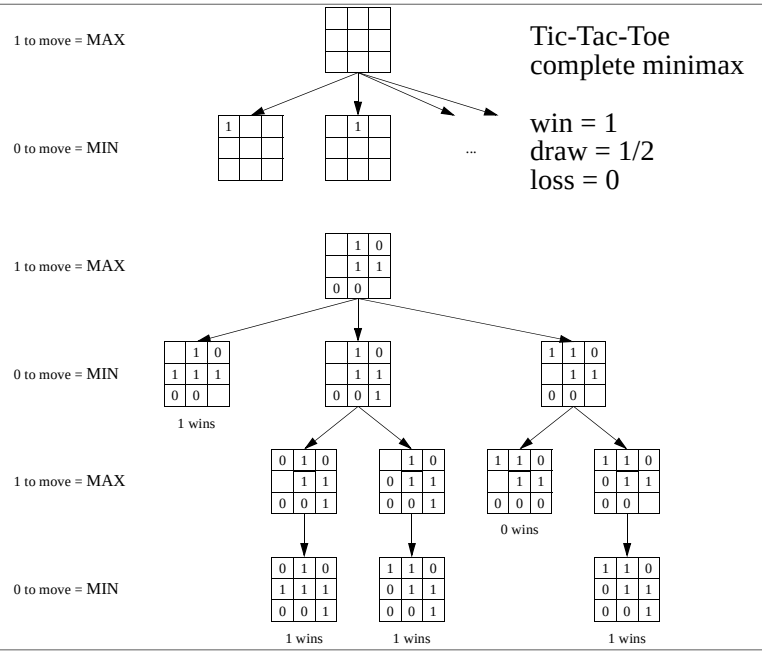
machines are better than humans in:
othello

machines and humans are about equally good in:
checkers (draughts), backgammon, scrabble, chess, bridge?

humans are better than machines in:
go

here: perfect information zero-sum games





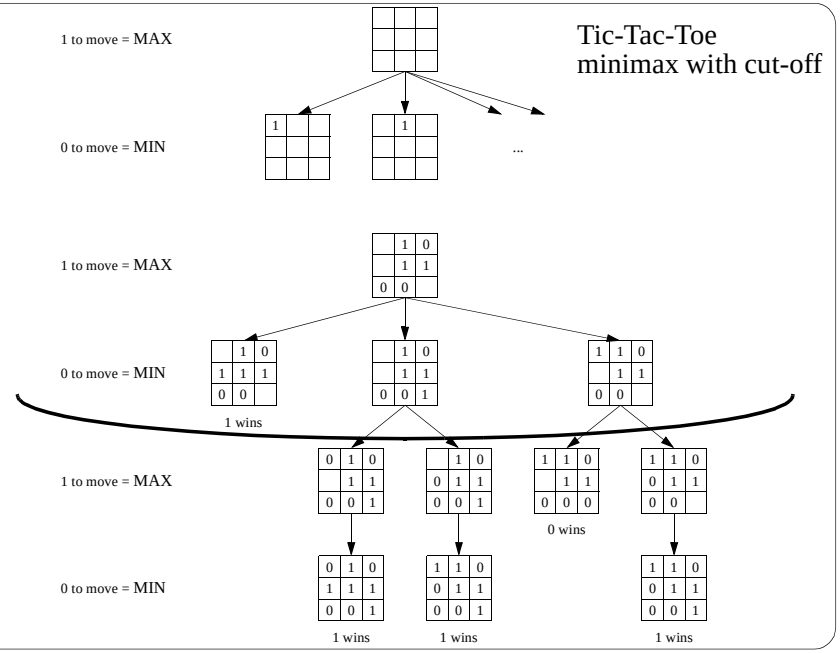
minimax algorithm

at terminal node:
the minimax value of the terminal node is
given by the evaluation function (0 = loss, 1/2 = draw, 1 = win)

at a non-terminal MAX node:
calculate the minimax values of its successor nodes
the minimax value of the MAX node is
the maximum of the minimax values of its successor nodes

at a non-terminal MIN node:
calculate the minimax values of its successor nodes
the minimax value of the MIN node is
the minimum of the minimax values of its successor nodes

chess:
10^{40} different legal positions
35^{100} nodes in average game tree



issues

evaluation function
- roughly corresponds to the “likelihood of winning”
- must be between the value of a winning terminal state and a losing one
- must be easy to calculate (= fast)

often a weighted linear function
with handselected features and learned coefficients

features for chess
-
-
-
-
-

reach quiescence (horizon effect)

alpha-beta

= same result as minimax but more efficient

insight:

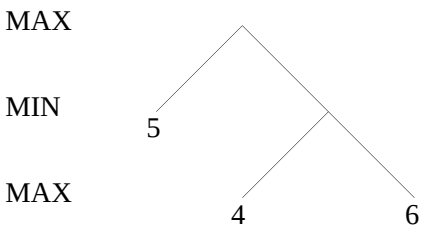
one does not need to look at all nodes to determine
the minimax value of the root of the game tree

alpha-beta algorithm

perform a depth-first minimax search

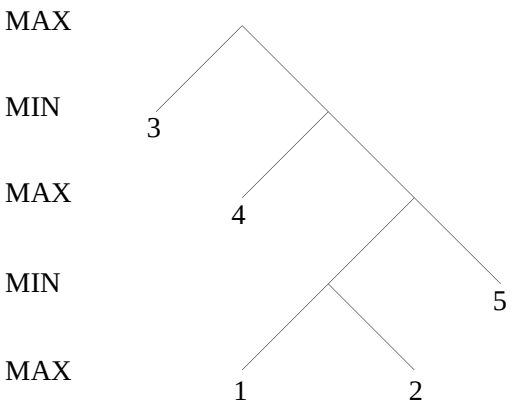
if the minimax value of the root of the game tree is the same
not matter what the minimax value of a node is
then we can prune this node and all of its descendants

an easy(?) example

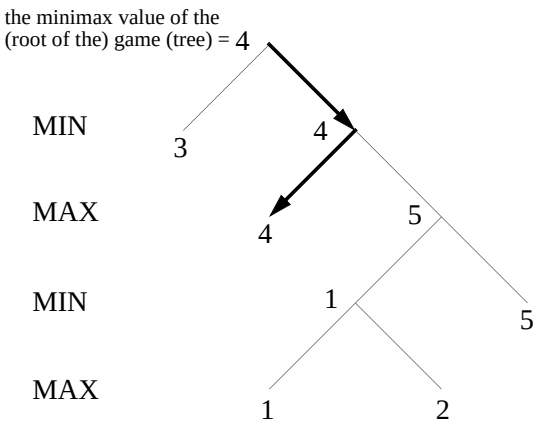


often, alpha-beta is able to cut huge subtrees
ideally, it can search a game tree twice as deep as minimax
in the same amount of time

a harder(?) example



a harder(?) example: minimax

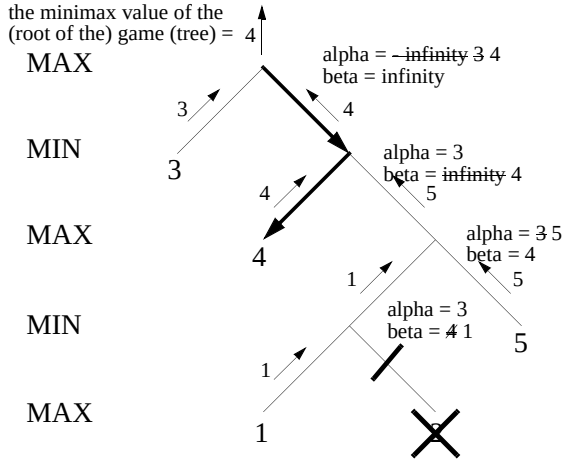


alpha-beta algorithm

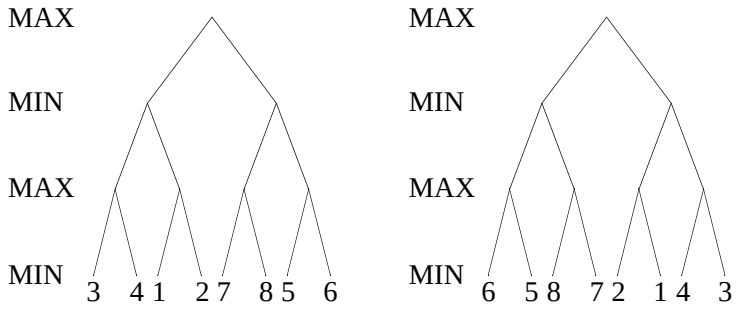
alpha = best (largest) minimax value MAX is guaranteed to reach
beta = best (smallest) minimax value MIN is guaranteed to reach

```
call MAX-VALUE(state = root, alpha = loss, beta = win)

MAX-VALUE(state, alpha, beta)
if node is a terminal node (or to be treated like one) then
    return the value of the evaluation function for that node
else
    for each succesor state s of state do
        set alpha := MAX(alpha, MIN-VALUE(s, alpha, beta))
        if alpha >= beta then return alpha
return alpha

MIN-VALUE(state, alpha, beta)
if node is a terminal node (or to be treated like one) then
    return the value of the evaluation function for that node
else
    for each succesor state s of state do
        set beta := MIN(beta, MAX-VALUE(s, alpha, beta))
        if alpha >= beta then return beta
return beta
```

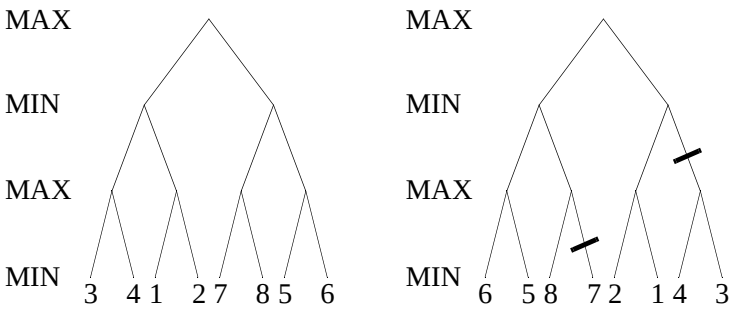
a harder(?) example: alpha-beta (a “deep” cut)



another example



another example



try “killer moves” early