AI algorithms: Games

MINI-MAX
Alpha-Beta

Input:
A game *board* and a rule set R defining legal moves
A *depthbound*, i.e. the number of turns you wish to think ahead
Ancestor α and β: *parent*α and *parent*β (init: lowest and highest board evaluation)
Output:
The next best move m

```
Algorithm (board,depth,parentα,parentβ):
  $\alpha \leftarrow 0$ (lowest board evaluation)
  $\beta \leftarrow$ infinity (highest board evaluation)
  if $depth = depthbound$ then
    return an evaluation of board
  else
    if is_max_level($depth$) then
      for all legal move $m$ on board given $R$ do
        $newboard \leftarrow$ do move $m$ on board
        $value_m \leftarrow$ recurse($newboard$,$depth + 1$,$parent\alpha$,$parent\beta$)
        $\alpha \leftarrow \max(\alpha, value_m)$
        $parent\alpha \leftarrow \max(\alpha, parent\alpha)$
        if $parent\beta \leq \alpha$ then
          return $m$ with $min(value_m)$ (stop loop)
        end if
      end for
      return $m$ with $max(value_m)$
    end if
    if is_min_level($depth$) then
      for all legal move $m$ on board do
        $newboard \leftarrow$ do move $m$ on board
        $value_m \leftarrow$ recurse($newboard$,$depth + 1$,$parent\alpha$,$parent\beta$)
        $\beta \leftarrow \min(\beta, value_m)$
        $parent\beta \leftarrow \min(\beta, parent\beta)$
        if $parent\alpha \geq \beta$ then
          return $m$ with $min(value_m)$ (stop loop)
        end if
      end for
      return $m$ with $min(value_m)$
    end if
  end if
```