Recursive Minimax

Minimax is used to dynamically evaluate a position in the game. The procedure is described here as a depth first search. The nodes in the search tree are positions. A node is a "**max**" node if it describes a position where the "maximizer" is to make a move. It is a "**min**" node if it describes a position where the "minimizer" is to make a move. The children of a node are all the possible positions that can be reached after one move. With each position x we associate the value V_x. The function **static**(x) gives a static evaluation of the position. The following two recursive procedures evaluate "dynamically" the value V_x of the node x:

if x is a max node: $V_x = \text{MaxMin}(x)$, where:

$$\text{MaxMin}(x) = \begin{cases} \text{static}(x) & \text{if } x \text{ is a leaf} \\ \max\limits_{\text{children } y \text{ of } x} \text{MinMax}(y) & \text{otherwise} \end{cases}$$

if x is a min node: $V_x = \text{MinMax}(x)$, where:

$$\text{MinMax}(x) = \begin{cases} \text{static}(x) & \text{if } x \text{ is a leaf} \\ \min\limits_{\text{children } y \text{ of } x} \text{MaxMin}(y) & \text{otherwise} \end{cases}$$

These routines can also be written as:

```
MaxMin(x):
if x is a leaf return static(x).
else {
    set v = −∞
    for each child y of x:
    v = max(v, MinMax(y))
    return v
}
```

```
MinMax(x):
if x is a leaf return static(x).
else {
    set v = +∞
    for each child y of x:
        v = min(v, MaxMin(y))
    return v
}
```

Recursive Alpha-Beta pruning

For each node x retain the range of values of its parents so that: $\alpha \leq V_{\text{parents of x}} \leq \beta$. If the root of the tree is a MAX node it is evaluated by: MaxMin(root, $-\infty, +\infty$).

MaxMin(x, α, β)**:**

1. if x is a leaf return static(x).

2. else do steps 2.1, 2.2, 2.3

2.1. set $v = -\infty$

2.2 for each child y of x:

2.2.1 $v = \max(v, \text{MinMax}(y, \alpha, \beta))$

2.2.2 if$(v \geq \beta)$ return v

2.2.3 else $\alpha = \max(v, \alpha)$

2.3. return v

MinMax(x, α, β)**:**

3. if x is a leaf return static(x).

4. else do steps 2.1, 2.2, 2.3

4.1 set $v = +\infty$

4.2 for each child y of x:

4.2.1 $v = \min(v, \text{MaxMin}(y, \alpha, \beta))$

4.2.2 if$(v \leq \alpha)$ return v

4.2.3 else $\beta = \min(v, \beta)$

4.3 return v

Notes:

. Step 2.2.2 is a β cut. The value returned at 2.2.2 will not affect the outcome since it will never be the minimum at 4.2.1.

. Same observation holds for the α cut at Step 4.2.2.