

### Pseudocode $\alpha$ - $\beta$ -pruning

AlphaBeta(*node*,  $\alpha$ ,  $\beta$ )

**Input:** *node* ist der aktuell zu behandelnde Knoten / aktueller Zustand  
die Werte  $\alpha$ ,  $\beta$  geben das Fenster für den Knoten an

**Output:** Wert  $\alpha$  für Knoten *node*, falls dieser ein Max-Knoten ist und  $\beta$  sonst

**IF** (*node* ist ein Blatt) **THEN**

    RETURN Auswertung zum Knoten *node*

**ENDIF**

**IF** (*node* ist ein Max-Knoten) **THEN**

    erzeuge alle gültigen Nachfolgezüge

**WHILE** (es gibt noch einen unbehandelten gültigen Nachfolgezug *move*) **DO**

        erzeuge Knoten *child* durch Anwendung von *move* auf *node*

$a := \text{AlphaBeta}(\text{child}, \alpha, \beta)$

        nimm Anwendung von *move* auf *node* zurück

**IF** ( $a \geq \beta$ ) **THEN**

**RETURN**  $\beta$

**ENDIF**

**IF** ( $a > \alpha$ ) **THEN**

$\alpha := a$

**ENDIF**

**ENDWHILE**

**RETURN**  $\alpha$

**ELSE** //Min-Knoten

    erzeuge alle gültigen Nachfolgezüge

**WHILE** (es gibt noch einen unbehandelten gültigen Nachfolgezug *move*) **DO**

        erzeuge Knoten *child* durch Anwendung von *move* auf *node*

$b := \text{AlphaBeta}(\text{child}, \alpha, \beta)$

        nimm Anwendung von *move* auf *node* zurück

**IF** ( $b \leq \alpha$ ) **THEN**

**RETURN**  $\alpha$

**ENDIF**

**IF** ( $b < \beta$ ) **THEN**

$\beta := b$

**ENDIF**

**ENDWHILE**

**RETURN**  $\beta$

**ENDIF**