

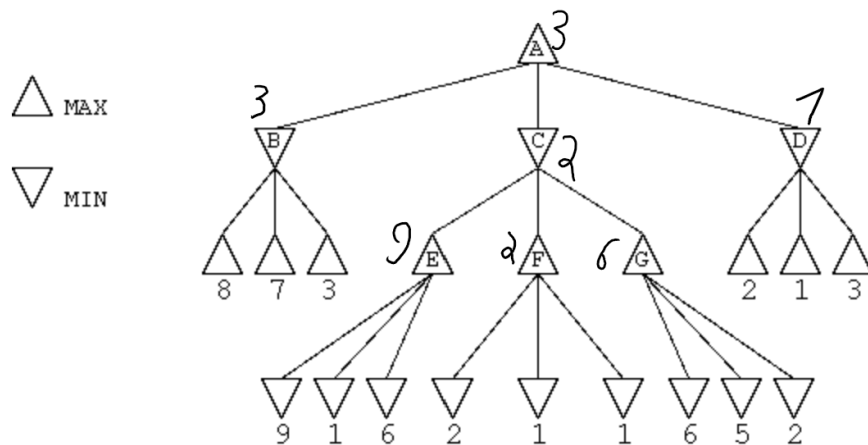
Blatt 03: Games

Michel Bünger

Games.01: Handsimulation: Minimax und alpha-beta-Pruning

Rückgabewerte

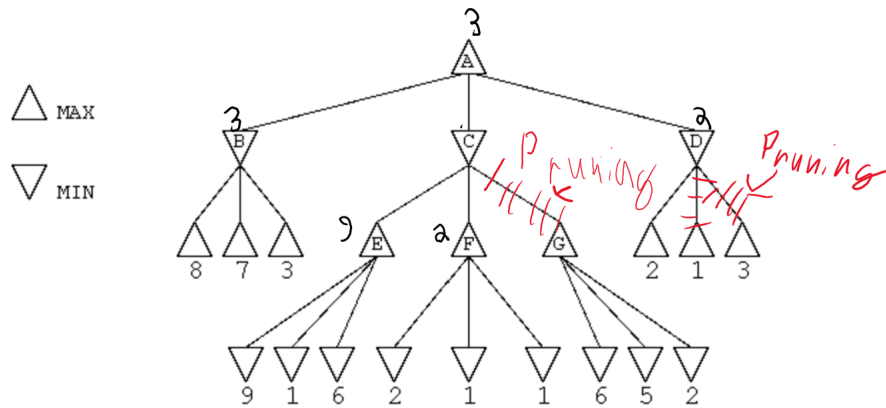
Die Minimax bewertungen für den Spielbaum, sehen wie folgt aus:



Knoten	Wert
A	3
B	3
C	2
D	1
E	9
F	2
G	6

Table 1: Minimax Werte

Pruning



Hier werden die Kinder von links nach rechts durchlaufen.

In MAX F wird 2 zurückgegeben, weshalb $\alpha = 3 \geq \beta = 2$ gilt und Pruning der restlichen Kinder von MIN C auslöst.

In MIN D wird als erstes 2 zurückgegeben, weshalb wieder $\alpha = 3 \geq \beta = 2$ gilt und Pruning ausgelöst wird.

Im folgendem werden die Fortschreitenden alpha und beta werte im Durchlaufen dieses Prunings dargestellt.

Max A ($\alpha = -\infty, \beta = +\infty$)

→ Min B ($\alpha = -\infty, \beta = +\infty$)

→ 8; $\beta = 8$

→ 7; $\beta = 7$

→ 3; $\beta = 3$

← Ergebnis 3 → $\alpha = 3$

→ Min C ($\alpha = 3, \beta = +\infty$)

→ Max E ($\alpha = 3, \beta = +\infty$)

→ 9; $\alpha = 9$

→ 1; $\alpha = 9$

→ 6; $\alpha = 9$

← Ergebnis 9 → $\beta = 9$

→ Max F ($\alpha = 3, \beta = 9$)

→ 2; $\alpha = 3$

→ 1; $\alpha = 3$

→ 1; $\alpha = 3$

← Ergebnis 2 → $\beta = 2$

→ $\alpha = 3, \beta = 2 \rightarrow \alpha \geq \beta \Rightarrow$ Pruning übriger

Min (Teilbäume)

← Ergebnis 2

→ Min D ($\alpha = 3, \beta = -\infty$)

→ 2; $\beta = 2$

→ $\alpha \geq \beta \Rightarrow$ Pruning übriger

Min D Teilbäume

← Ergebnis 2

Max A gibt 3 zurück

Erweitertes Pruning

Wenn die Teilbäume MAX E und MAX F getauscht werden, wird das Pruning der Restteilbäume von MIN C früher ausgeführt, weshalb dann zusätzlich, wie zuvor MAX G, auch MAX E durch das Pruning entfernt.

Games.04: Suchtiefe begrenzen

Die $Eval(s)$ Methode, besagt für das Bewerten eines Tic Tac Toe Spielzustandes, dass ein Terminaler Zustand für X_n , bei $n = \text{"Anzahl an X in einer Reihe/Spalten/Diagonalen ohne O"}$, wenn $n = 3$ gilt, ist $Eval(s) = 1$. Falls dies für 0 anstatt X gilt, ist $Eval(s) = -1$.

Falls ein Zustand terminiert, aber keiner dieser Bedingungen wahr ist, gilt $Eval(s) = 0$.

Um nicht-Terminierende Zustände zu bewerten, gilt jedoch folgende Funktion:

$$Eval(s) = 3X_2 + X_1 - (3O_2 + O_1)$$

Kurz gefasst wird hier die Wertung daran bestimmt, wie viele Zeichen jeder Spieler alleine in einer Reihe/Spalte/Diagonalen hat. Hierbei werden Reihen/..., in denen ein Spieler zwei Zeichen hat ($n=2$), dreifach so hoch gewertet.

Im folgendem sind drei Beispiele für die Bewertung eines Terminalen Zustandes und drei Beispiele für einen Zwischenzustand, mit ihren jeweiligen Bewertungen durch $Eval(s)$.

Terminierende

$$X_3 = 1$$

X	0	X
	X	
X	0	0

$$n \neq 3$$

$$u + l = 0$$

X	0	X
X	0	0
0	X	X

$$O_3 = -1$$

X		0
	X	0
X		0

Zwischenzustände

Q		X
X	X	O
O		

$$\begin{aligned}
 Eval &= 0 + 1 - (0 + 1) \\
 &= 1 - 1 \\
 &= 0
 \end{aligned}$$

Q	X	X
	Q	O
		X

$$\begin{aligned}
 Eval &= 0 + 1 - (3 \cdot 1 + 1) \\
 &= 1 - 4 \\
 &= -3
 \end{aligned}$$

X		O
	O	
		X

$$\begin{aligned}
 Eval &= 0 + 2 - (3 \cdot 1 + 0) \\
 &= 2 - 3 \\
 &= -1
 \end{aligned}$$

Games.05: Minimax generalisiert

Wenn Minimax für das Lösen eines nicht-Nullsummenspiels, mit drei Spielern verwendet wird, unter der Annahme, dass keine Allianzen gebildet werden können, agiert jeder Spieler egoistisch und versucht in seinem Zug, seinen eigenen Ergebniswert so sehr zu vergrößern, wie möglich.

Im folgenden sind die jeweils gewählten Tripel, welche für das Evaluieren des gegebenen Suchbaums gewählt wurden.

