

# Einführung in die Künstliche Intelligenz

SS 18

Prof. Dr. J. Fürnkranz, Prof. Dr. K. Kersting



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

---

## 3. Übungsblatt

---

### Aufgabe 1 Constraint Satisfaction Problem

---

Gegeben sei das folgende Cryptarithmic-Problem:

$$\begin{array}{r} A \\ + B \\ \hline B C \end{array}$$

- Geben Sie eine Formalisierung des Problems als Constraint Satisfaction Problem an, so dass die Addition für dezimal Zahlen korrekt ist. Das heißt, geben Sie noch die Wertebereiche für die vorkommenden Variablen  $A, B, C, U$  an und spezifizieren sie die Constraints.
- Wenden Sie den *Backtracking-Search* Algorithmus auf das Problem an. Nehmen Sie an, dass die Methode *Select-Unassigned-Variable* die Variablen in folgender Reihenfolge zurückgibt:  $C, B, A, U$ . Darüber hinaus liefert die Methode *Order-Domain-Values* die möglichen Werte in aufsteigender Reihenfolge.
- Wenden Sie nun zusätzlich *Forward-Checking* an und machen Sie sich klar, an welchen Stellen allgemeines Constraint-Propagation den Suchraum weiter reduziert hätte.

---

### Aufgabe 2 Minimax, Alpha-Beta Suche

---

Spielen Sie *3-Gewinnt*. Die Regeln sind wie folgt: Abwechselnd wird ein Stein in eine Spalte geworfen, er fällt dabei bis zum untersten freien Feld in dieser Spalte. In volle Spalten kann kein Stein mehr geworfen werden. Der Spieler, der die erste horizontale, vertikale oder diagonale Dreierreihe seiner Steine erhält, hat gewonnen. Bei Ihrem aktuellen Spiel sind sie MAX-Spieler mit weißen Steinen und sind gerade am Zug. Der Spielzustand ist der folgende:

	1	2	3	4
3		●	●	
2		○	○	
1		●	○	

- Zeichnen Sie zunächst den kompletten weiteren Spielbaum. Benutzen Sie für die Expansionen die Reihenfolge von links nach rechts, also das Ablegen des nächsten Steines in aufsteigender Spaltennummer.  
*Hinweis: Der Spielbaum ist groß. Eventuell sollte der Baum auf zwei Blätter verteilt werden.*

---

b) Wenden Sie den Minimax Algorithmus auf den aktuellen Zustand an. Endzustände in denen Weiß gewinnt bzw. verliert seien mit  $+1$  bzw.  $-1$  bewertet.

c) Wenden Sie den Alpha-Beta Algorithmus mit den initialen Schranken  $\alpha = -1, \beta = +1$  an. Die Abarbeitungsreihenfolge der Expansionen sei wie bei der Spielbaumgenerierung von links nach rechts. Geben Sie zusätzlich an, mit welchen Schranken die Knoten besucht werden.

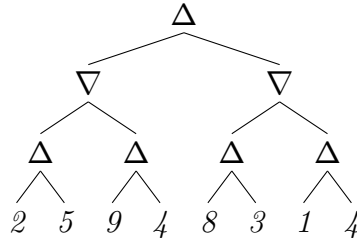
d) Wenden Sie nun erneut den Alpha-Beta Algorithmus (mit derselben Initialisierung) an. Jedoch seien diesmal die Alpha-Beta-Schranken mit  $-\infty, +\infty$  bewertet. Wie erklären Sie sich das unterschiedliche Pruning-Ergebnis im Vergleich zu Teilaufgabe c)?

---

### Aufgabe 3 NegaMax Formulierung, Minimal Window

---

Im folgenden Spielbaum geben die  $\Delta$ - und  $\nabla$ -Symbole an, dass der MAX-Spieler bzw. der MIN-Spieler am Zug ist.



Geben Sie in den folgenden Teilaufgaben jeweils an, mit welchen Schranken die Knoten besucht werden und wie sie sich im Laufe des Algorithmus verändern. Darüber hinaus sind alle Schranken mit  $(-\infty, +\infty)$  initialisiert.

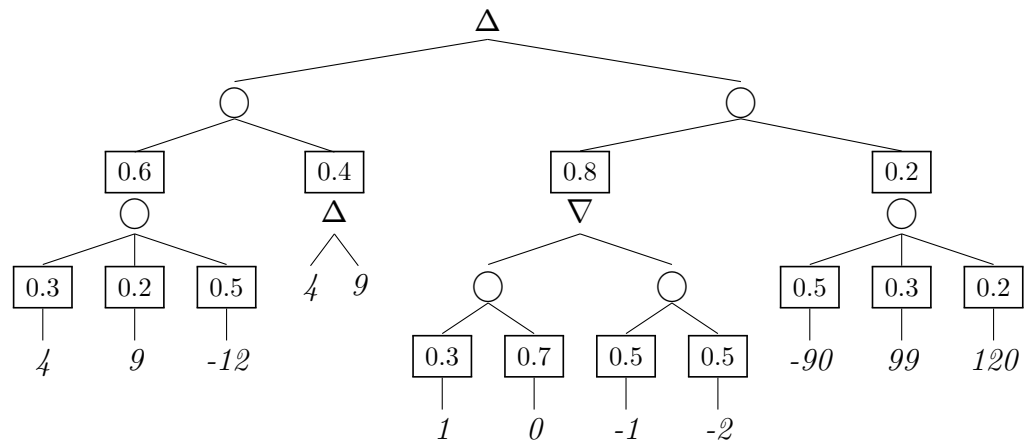
- Wenden Sie den Alpha-Beta Algorithmus an.
- Wenden Sie den Alpha-Beta Algorithmus in der NegaMax-Formulierung an. Negieren Sie dafür zunächst die Bewertungen der Endzustände oder ignorieren Sie die letzte Negierung des Rückgabewertes in der Rekursion, also wenn *AlphaBeta* mit einem Blattknoten aufgerufen wird (siehe Folien S. 33).
- Wiederholen Sie a) und b) mit der Minimal Window Erweiterung.

---

### Aufgabe 4 Expectimax

---

Im folgenden ist ein Zwei-Spieler Spielbaum dargestellt, der Zufalls-Knoten enthält. Bei Zufalls-knoten, die durch  $\bigcirc$  dargestellt sind, geben die Werte in den Boxen die Wahrscheinlichkeiten für die entsprechende Pfade/Ereignisse an.



- Versuchen Sie zu überlegen, welcher Zug gespielt werden wird.
- Wenden Sie den Expectimax-Algorithmus auf den obigen Spielbaum an.