

7. Übung zur Vorlesung Künstliche Intelligenz

Institut für Informatik, FU Berlin, SoSe 2005
Prof. Dr. Raúl Rojas, Marco Block, Ernesto Tapia

Neben der schriftlichen Abgabe, sind die Programmieraufgaben **zusätzlich** per e-mail an den Tutor zu schicken. Eine e-mail ersetzt nicht die schriftliche Abgabe!

Aus gegebenem Anlass ist es notwendig, auf diesem Übungszettel folgende Daten aufzuführen:

- Nachname, Vorname
- Matrikelnummer
- Studiengang (Bachelor, Master, Diplom) (⇐ sehr wichtig!)
- Fachsemester

1. Aufgabe (12 Punkte)

In der Vorlesung wurde der MinMax-Algorithmus im PseudoCode vorgestellt, dabei wurden zwei Knotentypen verwendet `maxKnoten` und `minKnoten`.

- (4 Punkte) Geben Sie eine Variante dieses Algorithmus an, die mit nur einer Funktion auskommt. Sie dürfen dabei folgende Methode als gegeben ansehen: `whiteToMove` → liefert `true`, wenn Weiss am Zug ist, ansonsten `false`.
- (8 Punkte) Erweitern Sie die Variante aus a), indem zusätzlich die HauptVariante mitgespeichert wird. Argumentieren Sie genau, warum Ihre Lösung am Ende den "besten Weg" liefert.

2. Aufgabe (20 Punkte + 10 Zusatzpunkte)

Die folgende Aufgabe beschäftigt sich mit dem Klassikerspiel Tic-Tac-Toe. Die Teilaufgaben können mit folgenden Programmiersprachen entwickelt werden:

Prolog, Java, C, C++, C#.

- (5 Punkte) Wählen Sie eine geeignete Brett-Darstellung und schreiben Sie einen Zuggenerator für Tic-Tac-Toe.
- (5 Punkte) Schreiben Sie eine Bewertungsfunktion für Tic-Tac-Toe. Diese soll sich lediglich auf *terminale* Stellungen beziehen. Wenn **X** das Spiel gewinnt, dann liefert die Funktion eine 1, bei Remis eine 0 und sollte **O** gewinnen, so liefert die Funktion eine -1.

Tipp: Illegale Stellungen, in denen z.B. beide gewinnen brauchen sie nicht berücksichtigen, da Ihr Zuggenerator aus a) vermutlich korrekt arbeitet ;).

- (5 Punkte) Ziel dieser Aufgabe ist es nun, einen spielbaren Agenten zu erhalten. Implementieren Sie das MinMax-Verfahren und verwenden Sie den Zuggenerator aus a) und die Bewertungsfunktion aus b).

- d) (5 Punkte) Erweitern Sie die Suche auf die Speicherung der Hauptvariante. Geben Sie diese beim Spiel nach jedem Computerzug (inklusive aktuellen Zug) aus. Beispiel:

Startstellung:

```
  a b c
3 X 0 X
2 - - 0
1 - - -
```

ComputerZug:

```
  a b c
3 X 0 X
2 - - 0
1 X - -
```

HauptVariante: a1=X, a2=0, b2=X.

- e) (Zusatzaufgabe 10 Punkte) Implementieren Sie das Alpha-Beta-Verfahren und verwenden Sie eine zusätzliche Heuristik, die eine Sortierung der Zuglisten vornimmt, nachdem Sie generiert wurden. Finden Sie eine gute Heuristik und vergleichen Sie die Anzahl der verwendeten Knoten paarweise für das MinMax-Verfahren, Alpha-Beta-Verfahren (ohne Heuristik) und für das Alpha-Beta-Verfahren (mit Heuristik).

Abgabe: (Montag)06.06.2005, 12:00 Uhr (s.t.)
(verspätete Abgaben werden nicht mehr entgegen genommen!)