

8. Übung zur Vorlesung Künstliche Intelligenz

Institut für Informatik, FU Berlin, SoSe 2006
Prof. Dr. Raúl Rojas, Marco Block, Ernesto Tapia

Alle Übungsaufgaben sollen mit SWI-Prolog gelöst werden. Neben der schriftlichen Abgabe, sind die Programmieraufgaben **zusätzlich** per e-mail an den Tutor zu schicken. **Eine e-mail ersetzt nicht die schriftliche Abgabe!** Zur Erinnerung: Testläufe gehören zur Abgabe und werden ebenfalls bewertet.

1. Aufgabe (9 Punkte) *AlphaBeta-Algorithmus*

Pseudocode von $\alpha\beta\text{MaxKnoten}$:

```
 $\alpha\beta\text{MaxKnoten}(\text{Knoten } x, \text{Integer } \alpha, \beta) \rightarrow \text{Integer } \{$   
  if ( $x$  ist Blatt)  
    return evaluation( $x$ )  
  else {  
    //  $x_1, \dots, x_k$  seien die Kinder von  $x$   
     $w := \alpha$   
    for  $i := 1$  to  $k$  do {  
       $v := \alpha\beta\text{MinKnoten}(x_i, w, \beta)$   
      if ( $v > w$ )  
         $w := v$   
      // Test auf  $\beta$ -Cutoff  
      if ( $w \geq \beta$ )  
        return  $w$   
    }  
    return  $w$   
  }  
}
```

a) (1 Punkt) Geben Sie eine Methode $\alpha\beta\text{Search}$ in Pseudocode an, die für eine gegebene Stellung s den AlphaBeta-Algorithmus startet. Zur Verfügung steht Ihnen ein Flag $s.\text{whitetomove}$, dass nur dann *true* ist, wenn Weiss am Zug ist.

b) Ergänzen Sie $\alpha\beta\text{MaxKnoten}$, $\alpha\beta\text{MinKnoten}$ und $\alpha\beta\text{Search}$ um folgende Funktionalitäten und geben Sie den zugehörigen Pseudocode an:

(i) (2 Punkt) Tiefenbeschränkung bis Tiefe t .

(ii) (6 Punkte) Die Hauptvariante HV (*Move-Liste*) steht nach Ausführung des AlphaBeta-Algorithmus zur Verfügung. Überlegen Sie sich eine geeignete Datenstruktur und beschreiben Sie Ihr Verfahren genau. Ergänzen Sie Ihren Code gegebenenfalls mit Funktionen wie $x.\text{getMove}(x_i) \rightarrow \text{Move}$, die z.B. den gültigen Namen des Zuges zurückgibt, wenn man von Knoten x zu Knoten x_i gelangen möchte. Sie können diese Funktion verwenden, ohne den Code angeben zu müssen.

2. Aufgabe (10 Punkte) *TicTacToe-Spiel mit MinMax-Algorithmus*

Schreiben Sie ein TicTacToe-Programm in Prolog, C++ oder Java. Ihr Programm sollte folgende Eigenschaften besitzen:

- (i) (2 Punkte) Eine anwenderfreundliche Eingabemöglichkeit.
- (ii) (5 Punkte) Nach Eingabe eines Zuges immer den besten Antwortzug finden (MinMax-Algorithmus ohne beschränkte Suchtiefe ist dabei ausreichend).
- (iii) (3 Punkte) Verwenden Sie folgende Codierung der Felder und geben Sie nach jedem Computerzug die von ihm prognostizierte Hauptvariante an:

	<i>A</i>	<i>B</i>	<i>C</i>
3	–	–	<i>x</i>
2	<i>o</i>	<i>x</i>	–
1	<i>x</i>	<i>o</i>	–

Ein Zug wird dabei zuerst mit der Spalte und dann mit der Zeile angegeben, z.B. *A3*, *C1*.

Abgabe: (Montag) 26.06.2006, 12:00 Uhr (s.t.)
(verspätete Abgaben werden nicht mehr entgegen genommen!)