

# Künstliche Intelligenz I

Klausur 19.03.2010 (10:30 - 12:00)

## Aufgabe 1 (20P $\Rightarrow$ $\leq$ 20 Min)

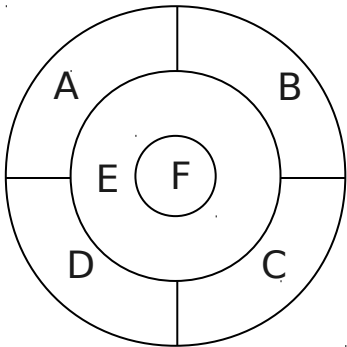


Bild 1

- (3P) Definieren Sie die „Minimum Remaining Values Heuristics“, die für die Backtracking Suchmethode benutzt werden kann.
  - (3P) Definieren Sie die „Degree Heuristics“, die für die Backtracking Suchmethode benutzt werden kann.
  - (3P) Definieren Sie die „Least Constraining Values Heuristics“, die für die Backtracking Suchmethode benutzt werden kann.
  - (11P) Betrachten Sie die Karte auf Bild 1. Färben Sie die Regionen mit den Farben Blau (b), Grün (g) und Rot (r), so dass direkt angrenzende Nachbar-Regionen unterschiedliche Farben haben. Benutzen Sie dafür die Backtracking Suchmethode mit den folgenden Heuristiken:
    - Für die Auswahl von Variablen (Regionen):
      - Minimum Remaining Values (**MRV**) Heuristics
      - Degree (**Deg**) Heuristics, wenn die MRV Heuristics es nicht erlaubt, zwischen Variablen zu entscheiden
      - Wenn keine von den oben genannten Heuristiken erlauben, eine Variable auszuwählen, wählen Sie die Variable (Spalte), die als erste im Alphabet (**Alpha**) vorkommt.
    - Für die Auswahl von Werten (Farben):
      - Least Constraining Values (**LCV**) Heuristics
      - Wenn die LCV Heuristik es nicht erlaubt, zwischen Werten zu entscheiden, wählen Sie die Farbe (**Farbe**) in der folgende Reihenfolge: Blau (b), Grün (g), Rot (r)
- WICHTIG:** Zusätzlich zu der Lösung geben Sie auch:
- Für jede Wahl die ausgewählte:
    - Region
    - Farbe
    - Benutzte Heuristiken (MRV, Deg oder Alpha, und LCV oder Farbe)
  - Sowie für jede übrige Region (noch nicht gefärbt/durchsucht):
    - Die mögliche Wahl von Farben (vor der Wahl)
    - Die Anzahl übriger Nachbar-Regionen (vor der Wahl)

**BITTE UMDREHEN!!**

## Aufgabe 2 (20P ⇒ ≤ 20 Min)

Das Spiel „Drei gewinnt“ ist von dem bekannten Spiel „Vier gewinnt“ abgeleitet. Das senkrecht stehende hohle Spielbrett besteht aus 3 Spalten (senkrecht, a bis c) und 3 Reihen (waagrecht, 1 bis 3). Wenn ein Spieler einen Spielstein (Kreis oder Kreuz) in eine Spalte fallen lässt, besetzt dieser den untersten freien Platz der Spalte. Im ersten Brettzustand auf Bild 2 (ganz links) kann der Spieler, der mit Kreisen spielt, also nur seinen Stein in die Positionen a2, b3 oder c3 setzen. Gewinner ist der Spieler, der es als Erster schafft, drei seiner Spielsteine waagrecht, senkrecht oder diagonal in eine Linie zu bringen. Das Spiel endet unentschieden, wenn das Spielbrett komplett gefüllt ist, ohne dass ein Spieler eine Dreierlinie gebildet hat.

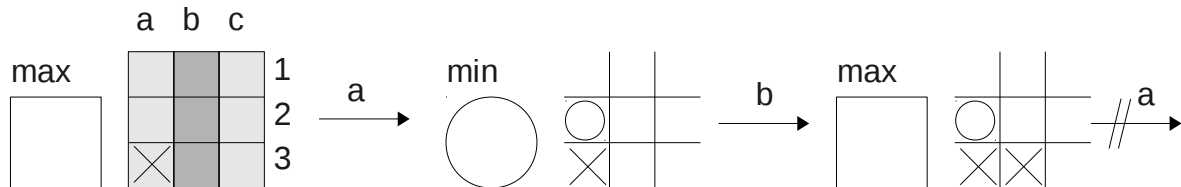


Bild 2

Beantworten Sie die folgenden Fragen:

- A. (4P) Was für eine Art Spiel ist das? Begründen Sie.
  - I. Perfect ODER imperfect information?
  - II. Deterministisch ODER Chancemäßig?
- B. (3P) Der  $\alpha$ - $\beta$  Algorithmus (minimax mit  $\alpha$ - $\beta$  Verfahren) kann verwendet werden, um dieses Spiel zu spielen. Unter welcher(n) Bedingung(en) gegenüber den Spielern berechnet der Algorithmus den optimalen Zug? Begründen Sie.
- C. (3P) Der  $\alpha$ - $\beta$  Algorithmus braucht eine Nutzfunktion.
  - I. Welche Bedingung(en) muss/müssen eine solche Nutzfunktion erfüllen, damit diese mit dem  $\alpha$ - $\beta$  Algorithmus verwendet werden kann? Begründen Sie.
  - II. Geben Sie ein Beispiel einer Nutzfunktion für das Spiel „Drei gewinnt“ an, die mit dem  $\alpha$ - $\beta$  Algorithmus verwendet werden kann.
- D. (10P) Zeichnen Sie den Entscheidungsbaum für den Zug des Spielers, der mit Kreisen spielt, für den Brettzustand, der auf Bild 3 dargestellt ist:

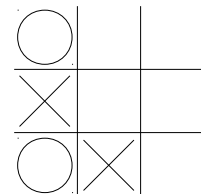


Bild 3

und geben Sie die Spalte an, die der Algorithmus ergibt: *b* oder *c*.

Wenden Sie dafür den  $\alpha$ - $\beta$  Algorithmus (minimax mit  $\alpha$ - $\beta$  Verfahren) an, und gehen Sie davon aus, dass der Spieler mit den Kreisen am Zug ist.

**WICHTIG:** Beachten Sie dafür die folgende Regeln:

1. Durchsuchen Sie immer die Spalten von links nach rechts. Das heißt, dass z.B. die Wahl von Spalte *b* nur möglich ist, wenn Spalte *a* voll ist, oder wenn der Teil des Baumes mit der Wahl von Spalte *a* schon besucht wurde.
2. An jedem Knoten des Baumes zeichnen Sie, so wie auf Bild 2 dargestellt ist:
  1. Ein Quadrat für das Ergebnis von der Maximum Funktion oder ein Kreis für das Ergebnis von der Minimum Funktion.
  2. Den Zustand des Brettspieles.
3. An jede Kante des Baumes schreiben Sie die Spalte, die ausgewählt wurde: *a*, *b* oder *c*.  
Wenn ein Teil des Baumes aufgrund  $\alpha$ - $\beta$  Pruning nicht durchsucht wird, streichen Sie den Pfeil mit 2 Strichen durch  $\not\!\!\! \rightarrow^a$  (s. dritten Pfeil auf dem Bild).

**BITTE UMDREHEN!!**

### Aufgabe 3 (5P ⇒ ≤ 5 Min)

Übersetzen Sie die folgenden Sätze in Prädikatenlogik erster Stufe, annehmend, dass das Diskursuniversum aus allen Menschen auf der Welt besteht.

- (a) Johan ist Hendriks Bruder.
- (b) Ein Hannoveraner singt.
- (c) Alle Hannoveraner jubeln.
- (d) Jeder hat eine Mutter. (Benutzen Sie die Variable  $x$  für den Sohn und  $y$  für die Mutter).
- (e) Entweder Christian oder Paul ermordet Lisa.

### Aufgabe 4 (10P ⇒ ≤ 10 Min)

Welches Ergebnis liefert Prolog auf die folgenden Anfragen?

- a.  $[a,b,c,d] = [a,[b,c,d]]$ . (1p)
- b.  $[a,b,c,d] = [a,b,[c,d]]$ . (1p)
- c.  $[a,b,c,d] = [a,b|[c,d]]$ . (1p)
- d.  $[a,b,c,d] = [a,b,c,[d]]$ . (1p)
- e.  $[a,b,c,d] = [a,b,c|[d]]$ . (1p)
- f.  $[a,b,c,d] = [a,b,c,d,[]]$ . (1p)
- g.  $[a,b,c,d] = [a,b,c,d|[[]]]$ . (1p)
- h.  $[] = \_$ . (1p)
- i.  $[] = [\_]$ . (1p)
- j.  $[] = [\_|[[]]]$ . (1p)

### Aufgabe 5 (10P ⇒ ≤ 10 Min)

Schreiben Sie ein Prolog - Prädikat

`merge(List1, List2, List3),`

wobei List1 und List2 zwei sortierte Listen sind, die in List3 (sortiert) zusammengemischt werden.

*z.B. merge([2,4,7], [1,3,4,8], [1,2,3,4,4,7,8]).*

### Aufgabe 6 (10P ⇒ ≤ 10 Min)

Schreiben Sie ein Prolog Prädikat, `rev/2`, welches eine Liste umdreht.

*z.B. rev([1,2,3,4], [4,3,2,1]).*

### Aufgabe 7 (15P ⇒ ≤ 15 Min)

Die Fibonacci Zahlen sind definiert wie folgt:

$\text{fib}(0) = 0$   
 $\text{fib}(1) = 1$   
 $\text{fib}(n) = \text{fib}(n-1) + \text{fib}(n-2)$  for  $n \geq 2$ .

Schreiben Sie in Prolog ein Prädikat welches die Fibonacci Zahlen berechnet.

?- `fib(3,N)`.  
 $N=2$

?- `fib(4,N)`.  
 $N=3$