

---

Aufgaben zur Klausur **Expertensysteme** im Fach **Software** im WS 96/97 (II 29)

Zeit: 60 Minuten

erlaubte Hilfsmittel: keine

Bitte tragen Sie Ihre Antworten und fertigen Lösungen ausschließlich an den gekennzeichneten Stellen in das Aufgabenblatt ein. Ist ihre Lösung wesentlich umfangreicher, so überprüfen Sie bitte nochmals Ihren Lösungsweg.

Viel Erfolg !

Diese Klausur besteht einschließlich dieses Deckblattes aus 7 Seiten

---

**Aufgabe 1:**

Transformieren Sie die folgende Formel des Prädikatenkalküls in Klauselform:

$$\neg(\exists x \in M \bullet a(x) \Rightarrow (\exists y \in M \bullet \forall x \in M \bullet b(y) \Rightarrow c(y, x)))$$

Resultat:

.....

.....

.....

---

**Aufgabe 2:**

Welche der folgenden Problemkennwerte gelten für das *SEND+MORE=MONEY* Zahlenrätsel?

1. Ist das Problem zerlegbar? ja  nein
  2. Können Lösungsschritte ignoriert werden? ja  nein
  3. Ist die Lösung voraussagbar? ja  nein
  4. Ist eine gute Lösung gefordert? ja  nein
  5. Spielt das Wissen eine große Rolle? ja  nein
  6. Ist eine Erklärungskomponente nötig? ja  nein
-

**Aufgabe 3:**

Der A\*-Algorithmus kann durch Wahl der Kostenfunktion  $g$  für den Weg vom Start bis zum Knoten und durch eine heuristische Funktion  $h$  für die Güte eines Knotens (Nähe zum Ziel) gesteuert werden.

Wie verhält sich der Algorithmus, wenn  $g$  und  $h$  wie folgt gewählt werden?

- 1.  $g(n)$  = Länge des Pfades vom Start zum Knoten  $n$   
 $h(n) = 1$  für alle Knoten  $n$

.....

- 2.  $g(n) = 0$  für alle Knoten  $n$   
 $h(n) \geq 0$  Nähe zum Ziel

.....

- 3.  $g(n) = 1$  für alle Knoten  $n$   
 $h(n) = 1$  für alle Knoten  $n$

.....

- 4.  $g(n) = -$  Länge des Pfades vom Start zum Knoten  $n$   
 $h(n) = 0$  für alle Knoten  $n$

.....

- 5. Welche Auswirkungen ergeben sich für den Algorithmus, wenn die Wegekosten gegenüber der heuristischen Funktion ein geringeres Gewicht erhalten ?

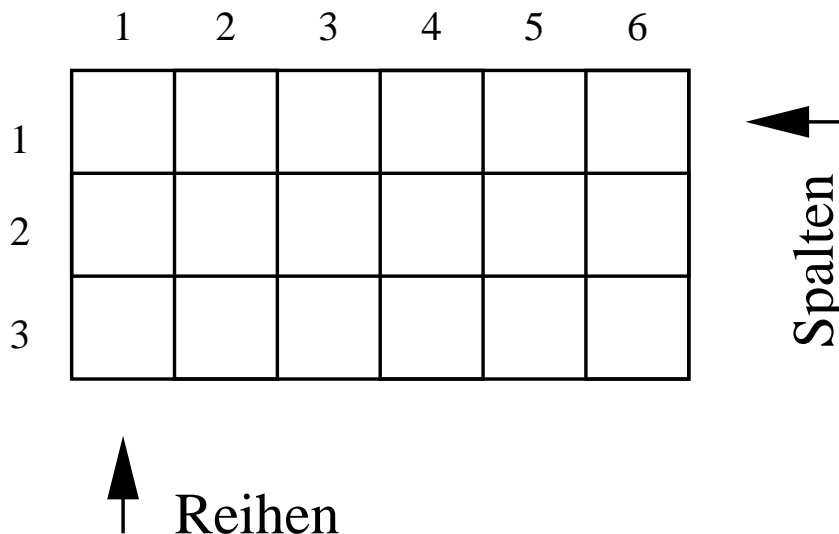
Die Anzahl der besuchten Knoten verringert sich

Die Anzahl der besuchten Knoten erhöht sich

Die Änderung hat keinen Einfluß auf die Anzahl der besuchten Knoten

#### Aufgabe 4:

Es soll ein Lastwagen mit Paletten beladen werden. Auf den Wagen passen genau 3 Paletten nebeneinander, 6 Paletten hintereinander. Die zur Verfügung stehenden Plätze sind in der folgenden Skizze veranschaulicht:



Auf den Wagen sollen die folgenden 8 Paletten geladen werden:

- p1** soll als erste beladen werden, da sie als letzte ausgeliefert werden soll. Sie ist so schwer, daß keine andere Palette daneben paßt.
- p2** soll als letzte beladen werden, da sie als erste ausgeliefert werden soll, sie ist ebenfalls so schwer, daß keine andere daneben paßt.
- p3,p4,p5** von diesen Paletten paßt jeweils nur eine in eine Reihe, da zwei von diesen wieder zu schwer für eine Reihe sind.
- p6** soll weiter hinten im Wagen plaziert sein als p3.
- p7** soll weiter hinten im Wagen plaziert sein als p4.
- p8** soll genau hinter p5 im Wagen plaziert sein.

Es soll die gesamte Länge der Ladefläche ausgenutzt werden.

Die Zuordnung der Paletten zu Stellplätzen soll mit Hilfe eines CLP Programms berechnet werden. In einem CLP Programm sind neben den üblichen Prolog-Prädikaten noch zusätzliche Prädikate für Einschränkungen eingebaut:

- die Operatoren  $\# =$ ,  $\# \setminus =$ ,  $\# >$ ,  $\# >=$ ,  $\# <$ ,  $\# = <$  für lineare Einschränkungen
- $L :: \text{min}..\text{max}$  zur Einschränkung der Wertemenge einer Liste von Variable auf ein Intervall  $\text{min}..\text{max}$  (Beispiel:  $[S, E, N, D] :: 0..9$ )
- $\text{alldistinct}(L)$  zur Einschränkung, daß alle Variablen der Liste  $L$  mit unterschiedlichen Werten belegt sein müssen (Beispiel:  $\text{alldistinct}([S, E, N, D])$ )

Die Einschränkungen werden mit folgendem hier nicht vollständig angegebenen Prolog-Prädikat beschrieben,  $S_i$  sind die Spalten,  $R_j$  die Reihen, dieses sind die Constraint-Variablen:

*mkvariable(PL) : -*

```
PL = [  
    p(p1, S1, R1),  
    p(p2, S2, R2),  
    p(p3, S3, R3),  
    p(p4, S4, R4),  
    p(p5, S5, R5),  
    p(p6, S6, R6),  
    p(p7, S7, R7),  
    p(p8, S8, R8)  
],  
nicht2PalettenAmGleichenPlatz(PL),  
... / * zuentwickelndeEinschränkungen * /  
.
```

Entwickeln Sie die zusätzlich notwendigen Einschränkungen, oben mit ... gekennzeichnet. Das Prädikat dafür, daß 2 Paletten nicht am selben Platz stehen können, soll hier nicht entwickelt werden. Die Einschränkungen werden gruppiert:

Die Einschränkungen an die Wertebereiche der Constraint-Variablen:

```
.....  
.....  
.....
```

Die Einschränkungen an die Paletten p1 und p2:

```
.....  
.....  
.....  
.....
```

Die Einschränkungen, daß Paletten p3, p4 und p5 nicht in eine Reihe passen:

.....  
.....  
.....  
.....

Die Einschränkungen an die Paletten p6, p7 und p8.

.....  
.....  
.....  
.....

Weitere notwendige Einschränkungen:

.....  
.....  
.....  
.....

