

3. Übungsblatt zur Künstlichen Intelligenz

Wintersemester 2004/2005

Die Übungen am 28. Oktober und am 2. November fallen aus! Die Abgabe der Übungsblätter verschiebt sich entsprechend.

Aufgabe 3.1* (linear ordering problem) Mögliche Kaufentscheidungen für Produkte seien gegeben durch eine Menge von Alternativen. In einem Test werden Konsumenten jeweils genau zwei dieser Produkte vorgestellt. Man fragt sie, welches der beiden Produkte sie vorziehen würden. Die dabei getroffenen Entscheidungen werden gezählt.

Produkt X	Produkt Y	Produkt X besser ($>$)	Produkt Y besser ($<$)
A	B	16	14
A	C	15	15
A	D	12	18
A	E	5	25
B	C	4	26
B	D	16	14
B	E	1	29
C	D	19	11
C	E	2	28
D	E	12	18

Leider sind Konsumenten, wenn man Ihnen immer nur zwei Alternativen anbietet, keineswegs konsistent in ihren Präferenzen. Ziel ist es daher, eine Rangfolge der Produkte festzulegen, die möglichst gut die Präferenzen der Konsumenten abbildet. Hierzu wird eine Bewertung für Rangfolgen eingeführt, die wie folgt definiert wird:

Für eine Rangfolge werden sämtliche Paare von Produkten betrachtet. Dabei wird die Summe aller Konsumententscheidungen gebildet, die inkonsistent mit der gegebenen Rangfolge sind. Eine Rangfolge ist dann am besten, wenn sie die kleinste Summe aufweist. Dies bedeutet, dass die Gesamtzahl der Konsumentenentscheidungen, die dieser Reihenfolge widersprechen, am geringsten ist.

Beispiele: Die Rangfolge $A > B > C > D > E$ erhält die Bewertung $14 + 15 + 18 + 25 + 26 + 14 + 29 + 11 + 28 + 18 = 198$. Die 14 ist hierbei die Anzahl der Entscheidungen mit $A < B$, 15 mit $A < C$ und 11 mit $C < D$. Die Rangfolge $C > E > B > D > A$ erhält die Bewertung $28 + 4 + 11 + 15 + 1 + 12 + 5 + 14 + 16 = 106$ und ist somit besser als die erste Rangfolge. Da es bei n Produkten $n!$ verschiedene Rangfolgen gibt, ist ein naiver Lösungsansatz (Generierung aller Rangfolgen mit Bewertung) für größere n nicht möglich. Daher bietet sich ein intelligentes Suchverfahren an.

Hinweise zum Zustandsraum:

- Für die Konstruktion einer Lösung können Sie zunächst die Entscheidung treffen, welches Produkt Platz 1 einnehmen soll, dann legen Sie Platz 2 fest, usw.
- Wenn Sie Platz 1 für Produkt X vergeben haben, gehen dadurch in Ihre Bewertung sämtliche Fälle für Produkte Y mit $X < Y$ ein. Beispiel: Mit der Wahl von A auf Platz 1 ergeben sich in der Bewertung die Anzahlen für $A < B$ (13), $A < C$ (25) und $A < D$ (18). Für die weiteren Entscheidungen setzt sich dies fort.

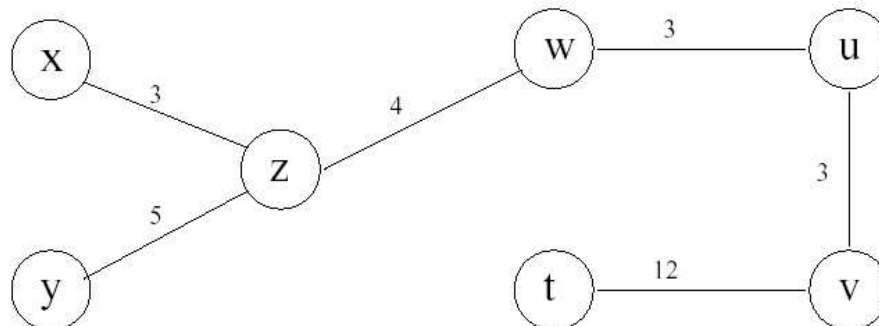
Geben Sie den Zustandsraum inklusive der Funktion g und einer zulässigen Funktion h für dieses Problem an!

Schreiben Sie ein Java-Programm, das den A*-Algorithmus verwendet, zur Lösung von Linear Ordering Problemen! Orientieren Sie sich dabei an den Beispielimplementationen der Suchprobleme aus dem AIMA Buch!

Wie lautet die optimale Lösung für die Daten der obigen Tabelle? Wie lange rechnet Ihr Programm für Probleme der Größe $n = 5$, $n = 20$ und $n = 100$?

Aufgabe 3.2 (constraint satisfaction)

Constraint Satisfaction erlaubt das Folgern und Ableiten von Zuweisungen an Variablen, die konsistent bzw. inkonsistent mit den Bedingungen (constraints) der Zielkonfiguration sind. Die Bedingungen werden durch Gleichungen und Ungleichungen repräsentiert. Zwei Constraint Propagation Methoden sollen im Folgenden untersucht werden: *Backtracking* und *arc consistency*. Die Verfahren unterscheiden sich bezüglich ihrer Vollständigkeit der Inferenz und ihres Rechenaufwandes.



Betrachten Sie nun obiges Constraintnetz. Jeder Knoten stellt eine Variable dar und jede Kante (arc) ist mit einer Zahl beschriftet. Die Variablen können nur Zahlen zwischen 0 und 9 annehmen (inklusive 0 und 9). Jede Kante repräsentiert eine Bedingung, die die beiden durch die Kante verbundenen Variablen erfüllen müssen. Die Bedingung besteht darin, dass jede Variable den gleichen Wert modulo der Kantenzahl haben muss. Zum Beispiel repräsentiert die Kante (x, z) mit dem Wert 3 die Bedingung: $x \bmod 3 = z \bmod 3$. Sie kann erfüllt werden durch die Zuweisungen $x = 2, z = 2$, oder $x = 5, z = 2$ oder $x = 7, z = 1$.

Für obigen Graphen seien die Belegungen $x = 2, y = 0$ und $t = 0$ bereits vorhanden. Geben Sie die Variablenbelegungen an, die durch backtracking und arc consistency.