Prof. Dr. Gerd Stumme, Robert Jäschke
Fachgebiet Wissensverarbeitung

21.11.2007

3. Übung „Künstliche Intelligenz“

Wintersemester 2007/2008

1 Suche

Berechnen Sie mit dem „Greedy“-Suchalgorithmus den Weg von *Arad* nach *Bucharest*. Verwenden Sie dazu die Heuristik und die Landkarte aus der letzten Übungsstunde und protokollieren Sie die einzelnen Schritte tabellarisch. Inwiefern unterscheidet sich die Lösung, welche Sie mit dem A^*-Algorithmus gefunden haben? Beschreiben Sie dazu inhaltlich den Unterschied zu A^* und klären Sie den Unterschied zur Gleichen-Kosten-Suche.

Schritt	OPEN	CLOSED
1	Arad	
2	Sibiu, Timisoara, Zerind	Arad
3	Fagaras, Rimnicu Vilcea, Timisoara, Zerind, Oradea	Arad, Sibiu
4	Bucharest, Rimnicu Vilcea, Timisoara, Zerind, Oradea	Arad, Sibiu, Fagaras

Der Greedy-Algorithmus findet schnell eine Lösung, jedoch nicht unbedingt die optimale – im Gegensatz zu A^*. Der gefundene Weg unter Verwendung des Greedy-Algorithmus ist 28 Kilometer länger als der von A^* gefundene Weg. Die Gleiche-Kosten-Suche verwendet die Kostenfunktion $g(n)$, der Greedy-Algorithmus die heuristische Funktion $h(n)$. Der A^*- Algorithmus verwendet zur Berechnung einer Lösung die Bewertungsfunktion $f(n) = g(n) + h(n)$

2 Constraintprobleme

1. Das 4-Damen-Problem: Das Ziel ist es 4 Damen so auf einem 4x4 Schachbrett zu plazieren, dass keine der Damen eine andere angreift. Folgende Bedingungen müssen erfüllt sein: keine zwei Damen dürfen in derselben Reihe, Spalte oder Hauptdiagonale stehen. Stellen Sie dieses Problem als Constraint-Problem dar. Sie dürfen annehmen, dass in jeder Spalte maximal eine Dame stehen kann.

Eine Zuordnung der Zahlen 1,2,3,4 zu den Spalten x_i mit $i \in \{1, \dots, 4\}$ soll gefunden werden (z. B. $x_1 \leftarrow 1$), so dass keine Dame auf dem Schachbrett bedroht wird. Die Formulierung des Suchraumes mit Constraints schränkt den Suchraum ein, wobei die erlaubten Zustände durch Werte einer Menge von Variablen definiert sind. Für die Variablen und Domäne ergeben sich: $V = \{x_1, x_2, x_3, x_4\}$ und $D = \{1, 2, 3, 4\}$. Folgende Constraints für C_{x_i,x_j} sind möglich:

a) $C_{12} = C_{23} = C_{34} = \{(1, 3), (1, 4), (2, 4), (3, 1), (4, 1), (4, 2)\}$

b) $C_{13} = C_{24} = \{(1, 2), (1, 4), (2, 1), (2, 3), (3, 2), (3, 4), (4, 1), (4, 3)\}$

c) $C_{14} = \{(1, 2), (1, 3), (2, 1), (2, 3), (2, 4), (3.1), (3, 2), (3, 4), (4, 2), (4, 3)\}$

2. In der Rechnung in Abbildung 1 sind die Buchstaben so durch die Ziffern 0 bis 9 zu ersetzen, dass die Rechnung aufgeht. Formulieren Sie das Problem als Constraintproblem. Geben Sie Variablen $v \in V$ und die dazugehörige Domäne $D(v)$ sowie die Constraints an. Beachten Sie die Berechnung von Überträgen und deren Umsetzung.

```
 SEND
+MORE
-----
MONEY
```

Abbildung 1: Ein Additionsproblem.

Für die Domänen der Variablen ergeben sich: $D, E, N, O, R, Y \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, für die führenden Stellen $M, S \in \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ und $U_1, U_2, U_3, U_4 \in \{0, 1\}$ für die Überträge. Es ergeben sich folgende Constraints:

$$
\begin{aligned}
C_1 &= D + E = Y + 10 * U_1 \\
C_2 &= N + R + U1 = E + 10 * U_2 \\
C_3 &= E + O + U2 = N + 10 * U_3 \\
C_4 &= S + M + U3 = O + 10 * U_4 \\
C_5 &= M = U_4
\end{aligned}
$$

3. Beschreiben Sie den Unterschied zwischen einem Suchbaum und einem Und-Oder-Baum. Zeichnen Sie sich hierfür die Struktur auf und beschreiben Sie die Komponenten und Funktionsweisen beider Fälle (Suchproblem, Anfangszustand, Operatoren, Zustände, Zielzustand).

Der Und-Oder-Baum ist ein Spezialfall eines Suchbaumes. Die Lösung kann nach dem Top-Down (zielgerichtet) bzw. Bottom-Up (datengetrieben) Verfahren geschehen.
Ein Und-Oder Baum besitzt zwei Knotentypen: *Oder-Knoten* (alternativen Methoden) und *Und-Knoten* (simultan zu erfüllende Ziele).

Suchbaum (im Falle einer Zustandsraumbeschreibung)

- **Suchproblem:** charakterisiert durch Startzustand und Zielzustandsbeschreibung
- **Anfangszustand:** Wurzelknoten ist Ausgangszustand
- **Operatoren:** Operatoren können einen Zustand in einen anderen transformieren
- **Zustände:** Knoten in einem Suchbaum, auf denen Operatoren angewendet werden
- **Zielzustand:** ein oder mehrere Knoten im Suchbaum

Und-Oder Baum (im Falle einer Problemzerlegung)

- **Suchproblem:** beschreibt eine Situation, in der ein Ziel durch das Lösen von Teilzielen erreicht werden kann
- **Anfangszustand:** Menge mit Ausgangsproblemen
- **Operatoren:** Ersetzung eines Problems in der Menge durch seine Teilprobleme (Problemzerlegung); Alternative Zerlegungen als Verzweigung; Primitive Probleme werden aus der Menge gestrichen
- **Zustände:** Menge offener Teilprobleme

3 Nochmal: Prädikatenlogik

Stellen Sie fest, ob die folgenden beiden Formeln erfüllbar sind. Geben Sie für jede Formel eine Struktur mit Individuen und Interpretation an, so dass die Struktur ein Modell der Formel ist, oder begründen Sie, warum die Formel nicht erfüllbar ist.

$$
\begin{aligned}
F := \text{istGluecklich}(y) \;\wedge\; \text{istKindVon}(x,y) \;\wedge\; \text{istDoktor}(x) \;\wedge \\
\Big(\forall u \forall v : (\text{istKindVon}(u,v) \;\wedge\; \text{istDoktor}(u)) \\
\Rightarrow (\text{istVater}(v) \;\wedge\; \text{istGluecklich}(v))\Big)
\end{aligned}
$$

Die Struktur $\mathcal{A} := (U_{\mathcal{A}}, I_{\mathcal{A}})$ sei durch die Menge der Individuen $U_{\mathcal{A}} = \{Peter, Paul\}$ und die Interpretationen

- $I_{\mathcal{A}}(x) = Paul$
- $I_{\mathcal{A}}(y) = Peter$
- $I_{\mathcal{A}}(\text{istDoktor}) = \{Paul\}$
- $I_{\mathcal{A}}(\text{istGluecklich}) = \{Peter\}$
- $I_{\mathcal{A}}(\text{istKindVon}) = \{(Paul, Peter)\}$
- $I_{\mathcal{A}}(\text{istVater}) = \{Peter\}$

definiert. $\mathcal{A}$ ist ein Modell für F, denn

1. $\text{istGluecklich}(y) \wedge \text{istKindVon}(x, y) \wedge \text{istDoktor}(x)$ ist wahr nach Interpretation von x und y,
2. $(\forall u \forall v : (\text{istKindVon}(u, v) \wedge \text{istDoktor}(u) \wedge$
$\Rightarrow (\text{istVater}(v) \wedge \text{istGluecklich}(v)))$ ist für alle Belegungen von u und v wahr (nur mit der Belegung $u = Paul$ und $v = Peter$ ist die Prämisse wahr – dann jedoch auch die Konklusion).

$$\begin{aligned} G := \forall x \forall y : (\neg\, \text{hatBelag}(x, y) \vee \text{Pizza}(x)) \wedge \forall x \forall y : (\neg\, \text{hatBelag}(x, y) \vee \text{PizzaBelag}(y)) \wedge \\ \forall x : (\neg\, \text{Pizza}(x) \vee \neg\, \text{PizzaBelag}(x)) \wedge \text{PizzaBelag}(aubergine) \wedge \\ \text{PizzaBelag}(cheddar) \wedge \text{hatBelag}(aubergine, cheddar) \end{aligned}$$

Diese Formel ist nicht erfüllbar. Denn angenommen, sie wäre erfüllbar. Dann existiert eine Struktur $\mathcal{A} = (U_\mathcal{A}, I_\mathcal{A})$, die ein Modell von G ist. Dann gilt:

$$I_\mathcal{A}(\mathrm{PizzaBelag}) \supseteq \{I_\mathcal{A}(cheddar), I_\mathcal{A}(aubergine)\} \tag{1}$$
$$I_\mathcal{A}(\mathrm{hatBelag}) \supseteq \{(I_\mathcal{A}(cheddar), I_\mathcal{A}(aubergine))\}. \tag{2}$$

Wegen (2) und „$\forall x \forall y : (\neg\,\mathrm{hatBelag}(x, y) \vee \mathrm{Pizza}(x))$“ würde dann (mit $I_\mathcal{A}(x) = I_\mathcal{A}(aubergine)$ und $I_\mathcal{A}(y) = I_\mathcal{A}(cheddar)$) auch gelten

$$I_\mathcal{A}(\mathrm{Pizza}) \supseteq \{I_\mathcal{A}(aubergine)\}. \tag{3}$$

Mit (1), (3) und „$\forall x : (\neg\,\mathrm{Pizza}(x) \vee \neg\,\mathrm{PizzaBelag}(x))$“ gilt dann für $I_\mathcal{A}(x) = I_\mathcal{A}(aubergine)$:

$$\begin{aligned}
I_\mathcal{A}(aubergine) &\in \overline{I_\mathcal{A}(\mathrm{Pizza})} \cup \overline{I_\mathcal{A}(\mathrm{PizzaBelag})} \\
&= \overline{I_\mathcal{A}(\mathrm{Pizza}) \cap I_\mathcal{A}(\mathrm{PizzaBelag})} \\
&\subseteq \overline{\{I_\mathcal{A}(aubergine)\} \cap \{I_\mathcal{A}(cheddar), I_\mathcal{A}(aubergine)\}} \\
&= \overline{\{I_\mathcal{A}(aubergine)\}}
\end{aligned}$$

Dies ist ein Widerspruch! Damit ist $\mathcal{A}$ kein Modell von G im Widerspruch zur Annahme und daher G nicht erfüllbar.