



Grundlagen der Künstlichen Intelligenz WS 05/06

Albayrak, Fricke, Jain (AOT) - Obermayer, Martin (NI)

3. Übungsblatt
Abgabe: 30.11.2005

Aufgabe 1

(6 Punkte)

Wir betrachten das n-Damen Problem von Übungsblatt 2, Aufgabe 2. Für die Bearbeitung dieser Aufgabe formulieren wir das n-Damen Problem als CSP Problem wie folgt:

Die **Variablen** Q_1, \dots, Q_n des CSP repräsentieren die n Spalten des Bretts. Der **Wertebereich** (Domäne) jeder Variable Q_i ist $D_i = \{1, 2, \dots, n\}$. Der Wert einer Variablen Q_i repräsentiert die Zeile der i -ten Spalte in der die i -te Dame steht. Die **Constraints** sind durch logische Ausdrücke der Form

$$i \neq j \Rightarrow Q_i \neq Q_j$$

(d.h. keine zwei Damen befinden sich in derselben Zeile) und

$$i \neq j \Rightarrow |Q_i - Q_j| \neq |i - j|$$

(d.h. keine zwei Damen befinden sich in derselben Spalte) beschrieben. Der Constraint „keine zwei Damen befinden sich in derselben Spalte“ ist implizit durch die gewählte Repräsentation gegeben.

Lösen Sie das n-Damen Problem für $n = 5, 6, \dots, 30$ mit den im Anhang beschriebenen Algorithmen $A = BT, BT+MRV+FC, BT+MRV+AC$.

- Stellen Sie in einer Grafik für jeden oben genannten Algorithmus A die Gesamtzahl der erzeugten Knoten als Funktion der oben genannten Problemgrößen n an.
- Stellen Sie in einer Grafik für jeden oben genannten Algorithmus A die „CPU-Zeit“ als Funktion der oben genannten Problemgrößen n an.
- Analysieren und diskutieren Sie die Resultate. Lohnen sich aufwändigere Heuristiken wie Arc Consistency?

Abzugeben sind

- Der Quellcode und eine README Datei an bjj@dai-labor.de. Die README Datei beschreibt, wie das Programm kompiliert und gestartet wird.
- Ein Ausdruck des Quellcodes.
- Problemformulierung
- Graphische Darstellung der Ergebnisse von b), d.h. zwei Plots mit je 3 Kurven.
- Diskussion (max. 5-7 Sätze).

Aufgabe 2

(4 Punkte)

Modellierung eines Reinigungs-Roboters im STRIPS Formalismus.

- a) Entwerfen Sie für diese Domäne sinnvolle Operatoren. Beachten Sie bitte dabei die folgenden Regeln:
 - i) Die Reinigung des Herds oder des Kühlschranks verschmutzt den Boden.
 - ii) Der Herd muss gereinigt sein bevor die Fettpfanne mit Aluminiumfolie abgedeckt wird.
 - iii) Das Reinigen des Kühlschranks produziert kleine Abfälle auf der Ablage und verschmutzt sie dadurch.
 - iv) Wenn die Ablage oder der Boden gewischt werden verschmutzt dies den Abfluss.
- b) Definieren Sie einen Anfangszustand einer Küche, die einen schmutzigen Herd, Kühlschrank, Ablage und Boden hat. Der Abfluss ist sauber und der Abfall ist weggebracht. Spezifizieren Sie auch einen Zielzustand in dem alles sauber ist, es keinen Abfall gibt und die Fettpfanne im Herd mit Aluminium abgedeckt ist.

Hinweise:

- **Abgabe der Lösungen:** Bitte Seite 4 ausgefüllt abgeben.
- Beschreibungen der Algorithmen befinden sich auf den Seiten 2-3.

Anhang: Algorithmen

- **Algorithmus BT**

1. Form a one-element stack consisting of an empty assignment of the variables.
2. Until the first assignment in the stack is complete and consistent or the stack is empty
 - a) Remove the first assignment from the stack.
 - b) Select an unassigned variable of the current assignment.
 - c) Create new assignments by assigning all consistent values to the selected unassigned variable.

- d) Push the new assignments, if any, to the top of the stack.
3. If a complete and consistent assignment is found, announce success; otherwise announce failure.

- **Algorithmus BT+MRV**

BT+MRV unterscheidet sich von BT in der Wahl einer unbelegten Variablen in Schritt 2b).

2b')

- i. Select an unassigned variable of the current assignment with minimum number of consistent values (MRV; minimum remaining value).
- ii. If there are two or more MRV-variables then **randomly** select one of them.
- iii. If the selected MRV-variable has zero consistent values, then omit steps 2c) and 2d) and continue with the next iteration in step 2.

- **Algorithmus BT+MRV+FC**

BT+MRV+FC ist eine Variante von BT+MRV, die **Forward Checking (FC)** verwendet, um in jeder Iteration die Anzahl der konsistenten Werte für die freien Variablen zu berechnen.

- **Algorithmus BT+MRV+AC**

BT+MRV+AC ist eine Variante von BT+MRV, die **Arc Consistency (AC)** verwendet, um in jeder Iteration die Anzahl der konsistenten Werte für die freien Variablen zu berechnen.

Der Algorithmus für AC ist von der folgenden Form:

Let $D(X_i)$ denote the set all consistent values from the domain of X_i .

1. Form a queue consisting of all arcs between all pairs of variables related by a constraint.
2. Until the queue is empty
 - a) Remove the first arc (X_i, X_j) from the queue.
 - b) Delete each value x from the domain $D(X_i)$ for which there is no consistent value y from the domain $D(X_j)$ such that the assignment $\{ X_i \leftarrow x, X_j \leftarrow y \}$ is consistent.
 - c) If an inconsistent value has been removed from the domain $D(X_i)$, then add all arcs of the form (X_k, X_i) to the queue, where $X_k \neq X_j$ is a variable related to X_i by a constraint.

Übungsblatt 3

Name & Matrikelnummer 1:

Name & Matrikelnummer 2:

Bitte Aufgabenteile ankreuzen, die vollständig bearbeitet wurden. Kommentare zur Bearbeitung können in der entsprechenden Spalte oder unterhalb der Tabelle eingetragen werden

Aufgabe	Bearbeitet	Kommentar
1		
2a		
2b		