

Einführung in die Künstliche Intelligenz

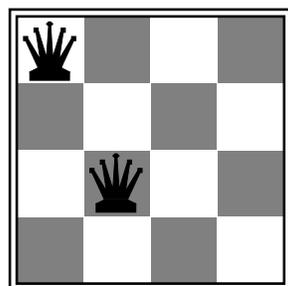
— Übungsblatt 3 —

Übung 3-1 (Spezifikation von Constraint Satisfaction Problemen)

Allgemein ist ein Constraint Satisfaction Problem (“Bedingungserfüllungsproblem”), kurz CSP, folgendermaßen definiert:

Gegeben ist eine Menge von Variablen Q_i mit zugehörigen Domänen (Wertebereiche) D_i (mit $i = 1 \dots n$) sowie eine Menge von Constraints (Bedingungen beziehungsweise n -äre Relationen), welche von den Variablen erfüllt sein müssen. Gesucht ist (mindestens) eine zulässige Belegung der Variablen, d.h. eine Belegung, die unter Berücksichtigung der Wertebereiche die Constraints erfüllt.

Abbildung 3-1 illustriert diesen Problemtyp anhand des n -Damenproblems für $n = 4$. Gegeben ist für jede Spalte i genau eine Dame Q_i , die auf dieser Spalte platziert werden muss; jede Dame Q_i stellt also eine Variable da, welche die Domäne $D_i = \{1, \dots, 4\}$ besitzt (da jede der Damen prinzipiell in einer der vier Zeilen platziert werden kann). Entsprechend den Schachregeln müssen für je zwei Damen Q_i und Q_j (mit $i \neq j$) folgende zwei Constraints erfüllt sein: (i) $Q_i \neq Q_j$ (d.h. sie dürfen nicht in derselben Zeile platziert sein) und (ii) $|Q_i - Q_j| \neq |i - j|$ (d.h. sie dürfen nicht in derselben Diagonale platziert sein). Damit lassen sich die zulässigen Belegungspaare generieren; diese sind beispielsweise für (Q_1, Q_2) die Paare $(1, 3), (1, 4), (2, 4), (3, 1), (4, 1)$ und $(4, 2)$.



$$Q_1 = 1 \quad Q_2 = 3$$

ABBILDUNG 3-1: CSP, illustriert am 4-Damenproblem (Russell & Norvig, Folien).

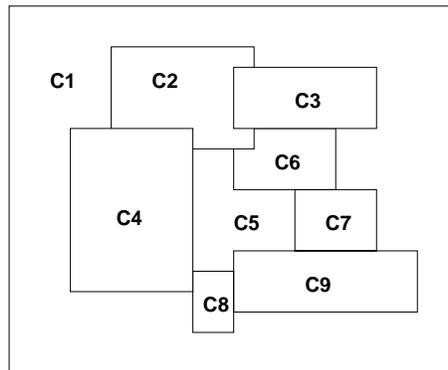


ABBILDUNG 3-2: Instantiierung des Kartenfärbungsproblems.

FORTY	29786
+ TEN	+ 850
+ TEN	+ 850
SIXTY	31486

ABBILDUNG 3-3: Instantiierung des kryptoarithmetischen Problems (Russell & Norvig).

OPTIONAL – 3-1-1. Abbildung 3-2 zeigt eine Instantiierung des sogenannten Kartenfärbungsproblems: gegeben sind hier 9 Bereiche (C1, C2, usw.), wobei jedem Bereich eine von vier möglichen Farben (rot, gelb, grün, blau) derart zugeordnet werden soll, dass keine zwei benachbarten Bereiche dieselbe Farbe besitzen. (Beachten Sie, dass C1 alle anderen Bereiche umfasst.) Formulieren Sie diese Probleminstantiierung als Constraint Satisfaction Problem.

OPTIONAL – 3-1-2. Abbildung 3-3 zeigt eine Instantiierung des sogenannten kryptoarithmetischen Problems samt seiner Lösung. Formulieren Sie die Probleminstantiierung als Constraint Satisfaction Problem.

Übung 3-2 (Uninformiertes Backtracking)

Ein elementarer Lösungsansatz für Constraint Satisfaction Probleme ist das uninformierte Backtracking. Hierunter versteht man eine "CSP-angepasste Tiefensuche", d.h. es wird eine Tiefensuche durchgeführt mit folgenden zwei Anpassungen:

- [1] es wird eine feste und eindeutige Zuordnung zwischen Variablen und Suchtiefe vorgenommen, so dass mit der Tiefe eines Knotes automatisch festgelegt ist, welche Variable an diesem Knoten zu belegen ist und
- [2] ein Knoten im Suchbaum wird nur expandiert, falls durch die bisherigen Variablenbelegungen kein Constraint verletzt ist.

Hierzu einige Erläuterungen:

- Die Anpassung [1] könnte z.B. realisiert werden durch folgende Zuordnung: “der Ursprungsknoten wird expandiert durch die Belegung von Q_1 , ein Knoten auf der Ebene 2 wird expandiert durch die Belegung von Q_2 , usw.” Die Anpassung [1] nutzt den Umstand aus, dass die Reihenfolge der Variablenbelegung unerheblich ist; dies bedeutet insbesondere, dass viele äquivalente Pfade generiert würden, falls jeder Knoten (unabhängig von seiner Tiefe) durch eine Belegung jeder beliebigen noch freien Variable expandiert werden kann.
- Die Anpassung [2] nutzt den Umstand aus, dass die Verletzung eines Constraints unabhängig von allen nachfolgenden Variablenbelegungen erhalten bleibt – die Fortsetzung der Tiefensuche wäre also in jedem Fall “reine Zeitverschwendung”.
- Falls die Variablenbelegung, die zu dem aktuell betrachteten Knoten geführt hat, nicht zulässig ist (und dieser Knoten deshalb wegen [2] nicht weiter expandiert wird), dann erfolgt der eigentliche Backtracking-Schritt: es wird zum unmittelbaren Vorgängerknoten zurückgegangen und geprüft, ob eine andere Belegung der diesem Vorgängerknoten entsprechend seiner Tiefe zugeordneten Variable möglich ist; falls ja, dann wird expandiert, und falls nein, dann wird sofort zu dessen Vorgänger zurückgegangen. Zu beachten ist, dass die maximale Suchtiefe bei Backtracking gleich der Anzahl der Variablen ist (also z.B. n beim n -Damenproblem).

3-2-1. Formulieren Sie das Backtracking-Verfahren in kommentierter Pseudonotation.

3-2-2. Implementieren Sie das Backtracking-Verfahren entsprechend Ihres Pseudocodes und wenden Sie Ihre Implementierung auf mindestens ein k -Damenproblem mit $k \geq 20$ an. Protokollieren Sie dabei (für jedes verwendete k) die Anzahl der erzeugten Knoten, die Anzahl der generierten ungültigen (weil Constraint-verletzenden) Knoten und die gefundene Lösung.

Bearbeitungen in PDF- und Source-Format (World, LaTeX, etc.)
bis zur nächsten Vorlesung an: gerhard.weiss@maastrichtuniversity.nl
