Kapitel 3: Constraints

Teil 1

(Dieser Foliensatz basiert auf Material von Mirjam Minor, Humboldt-Universität Berlin, WS 2000/01)

Constraint Satisfaction Problem (CSP)

Problem bei Suchverfahren: Kombinatorische Explosion

Häufig sind zusätzlich zum eigentlichen Problem noch eine Reihe einschränkender Nebenbedingungen (**Constraints**) gegeben, die ebenfalls zu erfüllen sind.

Wie können Constraints bei der Suche nach einer Lösung helfen ?

- durch Testen von Zwischenergebnissen bzgl. der Constraints lassen sich Irrwege frühzeitig erkennen
- durch Verkettung mehrerer Constraints treten manchmal Lösungen hervor (**Constraint Propagation**)

Anwendungsbeispiele

- Stundenplanerstellung
- Färbeprobleme in Graphen
- Transportprobleme / Logistik / Produktionsplanung
- Interpretation zweidimensionaler Linienzeichnungen / Bilderkennung
- Algebren zum zeitlichen Schließen

Mögliche Fragestellungen

- Ist eine Lösung unter den gegebenen Nebenbedingungen möglich?
- Wie sieht der Lösungsraum aus ?
- Falls keine Lösung existiert, kann man durch minimale Abweichung von den Constraints (abschwächen bzw. weglassen) zu einer Lösung kommen ?

Formale Definitionen

Gegeben: Variablenmenge $V = \{v_1, ..., v_n\}$ mit den Wertebereichen $Dom(v_i)$ für alle v_i $(i = 1, ..., n)$.
Sei $Dom(V) = Dom(v_1) \times ... \times Dom(v_n)$

Ein **k-stelliges Constraint** C über $V' = \{v'_1, ..., v'_k\} \subseteq V$ ist ein Teilmenge $C \subseteq Dom(v'_1) \times ... \times Dom(v'_k)$.

Ein Constraint mit Stelligkeit $k = 1$ heißt **unärer Constraint**
Ein Constraint mit Stelligkeit $k = 2$ heißt **binärer Constraint**

Formale Definitionen (2)

Ein **Constraint-Netz** $\mathcal{C}$ über V ist eine Menge $\mathcal{C} = \{C_1, ..., C_m\}$, wobei jedes C_i ein Constraint über einer Menge $V_i \subseteq V$ ist. Ein **binäres Constraint-Netz** ist ein Constraint-Netz, das nur unäre und binäre Constraints enthält.

Sei $\beta : V \to \bigcup_i Dom(v_i)$ eine Belegung, die jedem $v_i \in V$ ein Element aus $Dom(v_i)$ zuordnet.

Die Belegung β **erfüllt** ein Constraint C über $V' = \{v'_1, ..., v'_k\}$, falls $[\beta(v'_1), ..., \beta(v'_k)] \in C$. $[\beta(v'_1), ..., \beta(v'_k)] \in C$ heißt **lokale Lösung**.

Die Belegung β erfüllt das Constraint-Netz $\mathcal{C}$ und heißt **globale Lösung**, falls β alle $C \in \mathcal{C}$ erfüllt.

Beispiele

1. Das folgende Problem soll mit Constraints formuliert werden: Finde zwei natürliche Zahlen x und y, so daß $x + y = 7$ unter der Voraussetzung $x > y > 2$.

$V = \{x, y\}$, $Dom(x) = Dom(y) = \mathbb{N}$
$C_1 = \{y \in \mathbb{N} : y > 2\} \subseteq Dom(y)$ über $V_1 = \{y\}$
$C_2 = \{[x, y] \in \mathbb{N} \times \mathbb{N} : x > y\} \subseteq Dom(x) \times Dom(y)$ über $V_2 = \{x, y\}$
$C_3 = \{[x, y] \in \mathbb{N} \times \mathbb{N} : x + y = 7\} \subseteq Dom(x) \times Dom(y)$ über $V_3 = \{x, y\}$
$\mathcal{C} = \{C_1, C_2, C_3\}$

$x = 2$ und $y = 1$ ist eine lokale Lösung bzgl. C_2.
$x = 4$ und $y = 3$ ist eine globale Lösung.

Beispiele

2. Das Constraintnetz $\mathcal{C} = \{C_1, C_2, C_3, C_4\}$ sei wie folgt gegeben

$V = \{x, y, z\}$, $Dom(x) = Dom(y) = Dom(z) = [0, 1]$
$C_1 = \{[x, y] : x > y\}$ über $V_1 = \{x, y\}$
$C_2 = \{[y] : y > 0.5\}$ über $V_2 = \{y\}$
$C_3 = \{[x, z] : x + z = 1\}$ über $V_3 = \{x, z\}$
$C_4 = \{[x, z] : x < z\}$ über $V_4 = \{x, z\}$

$[0.5, 0.7, 0.5]$ ist eine lokale Lösung bzgl. C_3.

Eine globale Lösung existiert nicht (Aus C_1, C_2, C_4 folgt $z > x > y > 0.5$; Widerspruch mit C_3 $x + z = 1$). Die gegebenen Voraussetzungen sind inkonsistent. Ohne C_4 wäre $[0.7, 0.6, 0.3]$ eine globale Lösung.

Binäre Constraint-Netze als Graphen

Knoten entsprechen den Variablen, Kanten entsprechen den binären Constraints; Unäre Constraints sind als Einschränkung des Wertebereichs der Variablen enthalten. Bsp.: Vierfarbenproblem

Constraint-Netze höherer Ordung lassen sich in binäre C.-N. überführen.
$\rightarrow$ Vereinfachung / Vereinheitlichung der Probleme
$\rightarrow$ verschiedene Verfahren (speziell für endliche Wertebereiche) zum Finden einer Lösung
Aber: Komplexität der Umwandlung hoch bzw. Umwandlung nicht möglich

CSP als Suchproblem

Ausgangszustand: Die leere Zuweisung, d.h. alle Variablen sind undefiniert

Nachfolgerfunktion: Belegung einer weiteren Variablen, so dass kein Konflikt mit vorherigen Zuweisungen entsteht.

Zieltest: Sind alle Variablen definiert?

Pfadkosten: konstant, z.B. 1 pro Schritt

Lösung von CSP-Suchproblemen

Jede Lösung hat Tiefe $|V|$. Daher ist Tiefensuche geeignet.

Es kommt nicht auf den Suchpfad an. Daher sind lokale Methoden gut geeignet.

Heuristik des minimalen Konflikts: Wert der Zuordnung $x \leftarrow w$ ist Summe über alle noch undefinierten Variablen y der Anzahl der Werte $v \in D(y)$, so dass die Belegung $\{x \leftarrow w, y \leftarrow v\}$ ein Constraint verletzt. Bsp.: 4×4-Dame mit $(a, 1)$ belegt. Ist $(b, 3)$ oder $(b, 4)$ besser?

Komplexität/Unentscheidbarkeit

Constraint-Erfüllungsprobleme sind bei endlichen Wertebereichen im allgemeinen NP-vollständig.
In den meisten praktischen Anwendungen sind sie aber um Größenordnungen besser als allgemeine Suchverfahren.

Gleichungssysteme und speziell Diophantische Gleichungen (ganzahlige Lösungen für $a_1 x_{1,1}^{i_{1,1}} ... x_{n,1}^{i_{n,1}} + ... + a_m x_{1,m}^{i_{1,m}} ... x_{n,m}^{i_{n,m}} = c$) lassen sich als Constraint-Erfüllungsprobleme (Constraint Satisfaction Problem; CSP) beschreiben. Diese sind beweisbar unentscheidbar, d.h. es ist kein universeller Lösungsalgorithmus für CSP möglich.