Kapitel MK:IV

IV. Modellieren mit Constraints

- ❑ Einführung und frühe Systeme
- ❑ Konsistenz I
- ❑ Binarization
- ❑ Generate-and-Test
- ❑ Backtracking-basierte Verfahren
- ❑ Konsistenz II
- ❑ Konsistenzanalyse
- ❑ Weitere Analyseverfahren
- ❑ FD-CSP-Anwendungen
- ❑ Algebraische Constraints
- ❑ Intervall Constraints
- ❑ Optimierung und Überbestimmtheit

Einführung und frühe Systeme

Constraints sind eine Softwaretechnik zur deklarativen Beschreibung und zum effizienten Lösen großer Probleme.

Ein Constraint definiert eine Relation zwischen Variablen.

Beispiel:

$F = P * A$, $F \sim$ Kraft, $P \sim$ Druck, $A \sim$ Fläche

Ausgedrückt mit Fakten und Regeln:

```
Druck = 10 Bar
Fläche = 20 cm ˆ 2
IF (Druck = ?X) AND (Fläche = ?Y)
THEN (Kraft = ?X * ?Y)
```

Fragen:

- Welchen Zusammenhang modelliert die Regel?
- Was modelliert die Regel nicht?
- Wie kann man das reparieren?

Einführung und frühe Systeme

Beispiel (Fortsetzung):

$$F = P * A$$

Modellierung:

```
IF (Druck = ?X) AND (Fläche = ?Y)
THEN (Kraft = ?X ⋆ ?Y)
IF (Druck = ?X) AND (Kraft = ?Y)
THEN (Fläche = ?Y / ?X)
IF (Fläche = ?X) AND (Kraft = ?Y)
THEN (Druck = ?Y / ?X)
```

Es fehlt noch eine Regel. Welche?

Bemerkungen:

- Regeln stellen *gerichtete* Zusammenhänge auf.
  Prinzip der Kausalität bzw. Ursache-Wirkung
- Aus Modellierungssicht sind oft deklarative Zusammenhänge gewünscht.
- Deklarative Zusammenhänge sind typischerweise ungerichtet.

Einführung und frühe Systeme

Algebraische Constraints in EL [Sussmann/Stallman 1977]

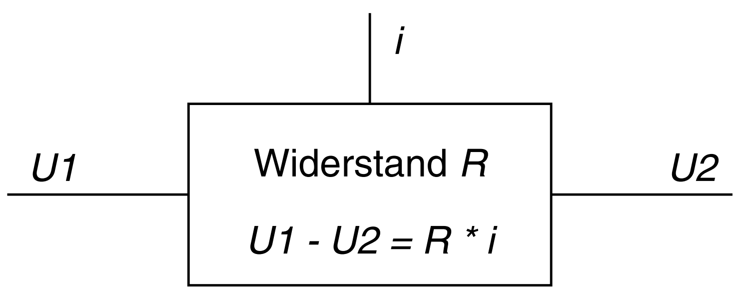
Anwendung: Zustandsgrößenberechnung in Schaltkreisen.

Constraint für einen Widerstand:

- Variablen: $U \sim$ Spannung, $R \sim$ Widerstand, $i \sim$ Stromstärke
- Relation: $U = R \cdot i$, Ohm'sches Gesetz

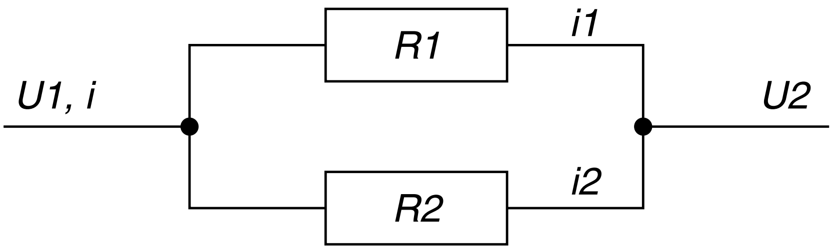
Modellierung des Constraints mittels Regeln:

```
R, i          ->  U1 - U2 = R * i
U1, U2, i     ->  R = (U1 - U2)/i
U1, U2, R     ->  i = (U1 - U2)/R
```



Einführung und frühe Systeme

Algebraische Constraints in EL (Fortsetzung)



- Potential-Constraints:

  $U_1 - U_2 = R_1 \cdot i_1$

  $U_1 - U_2 = R_2 \cdot i_2$
- Bilanz-Constraint: $i_1 + i_2 = i$

lokale Sicht: Beschränkung des Stromflusses

- `U1, U2, R1, R2 -> i1, i2, i`
- lokale(s) Analyse (Berechnungsverfahren) ist ausreichend, selbst bei nicht-linearen Zusammenhängen.

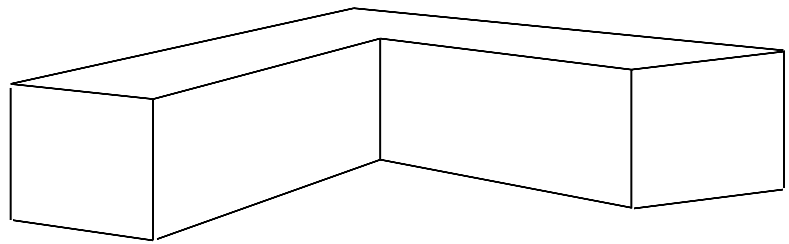
globale Sicht: Widerstände bestimmen Spannungsteilung

- `U1, R1, R2, i -> ?`
- globale Analyse notwendig: Berechnung des Spannungsabfalls mittels der Berechnung des Ersatzwiderstands
- in EL: Propagierung von Variablen und Constraints.

Einführung und frühe Systeme

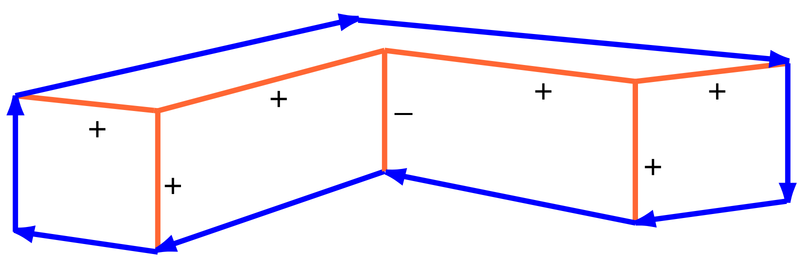
Nicht-algebraische Constraints in Waltz [Waltz 1972]

Anwendung: Objekterkennung auf Basis von Umrissen.



Konzept: Zuordnung von Linientypen mittels Markierungregeln:

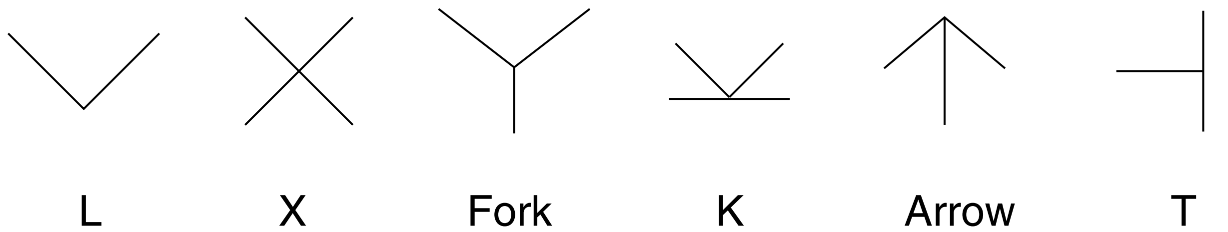
1. Randlinien. Grenzen Objekt von Umgebung ab. Markierung: „>“ oder „<“
2. Zyklen. Randlinien die einen gerichteten Kreis bilden, dienen zur Unterscheidung bzw. Trennung sich überlappender Objekte.
3. Innere Linien. konvexe Form (nach außen) oder konkave Form (nach innen). Markierung: „+“ und „–“



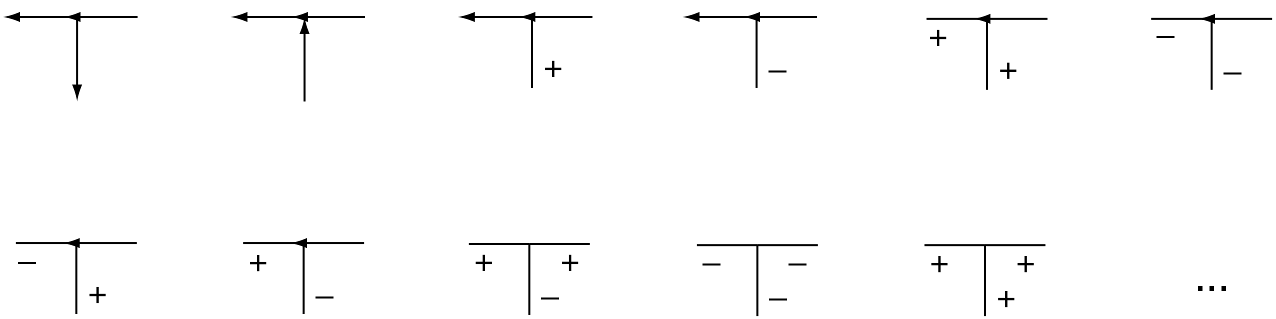
Einführung und frühe Systeme

Nicht-algebraische Constraints in Waltz (Fortsetzung)

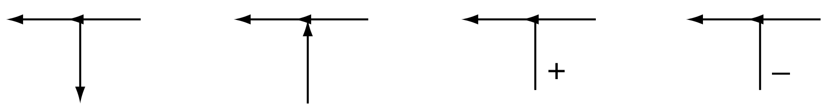
Eckentypen:



Mögliche Markierungen der T-Ecke:



Sinnvolle Markierungen der T-Ecke:

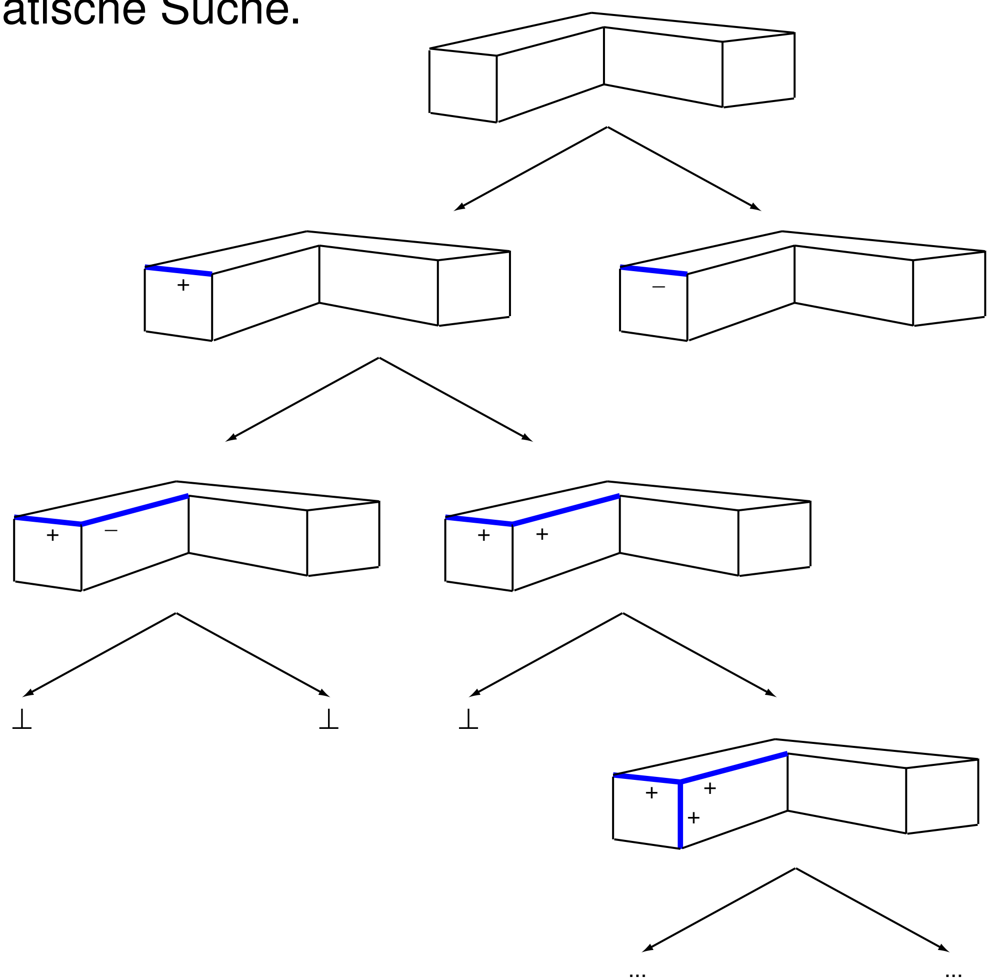


Einführung und frühe Systeme

Nicht-algebraische Constraints in Waltz (Fortsetzung)

Ziel: Erzeugung einer Markierung, die konsistent mit den Markierungsregeln ist.

Methode: systematische Suche.



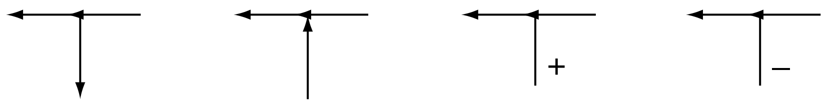
Bemerkungen:

- Die Entdeckung einer geeigneten Markierung durch Backtracking ist chancenlos.

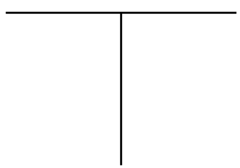
Einführung und frühe Systeme

Nicht-algebraische Constraints in Waltz (Fortsetzung)

Sinnvolle Markierungen der T-Ecke:



Betrachtung der sinnvollen Markierungen als Constraints:



$$
\begin{aligned}
\{\ &(v_1 =<,\ v_2 =<,\ v_3 = +)\ ,\\
&(v_1 =<,\ v_2 =<,\ v_3 = -)\ ,\\
&(v_1 =<,\ v_2 =<,\ v_3 =<)\ ,\\
&(v_1 =<,\ v_2 =<,\ v_3 =>)\ \}
\end{aligned}
$$

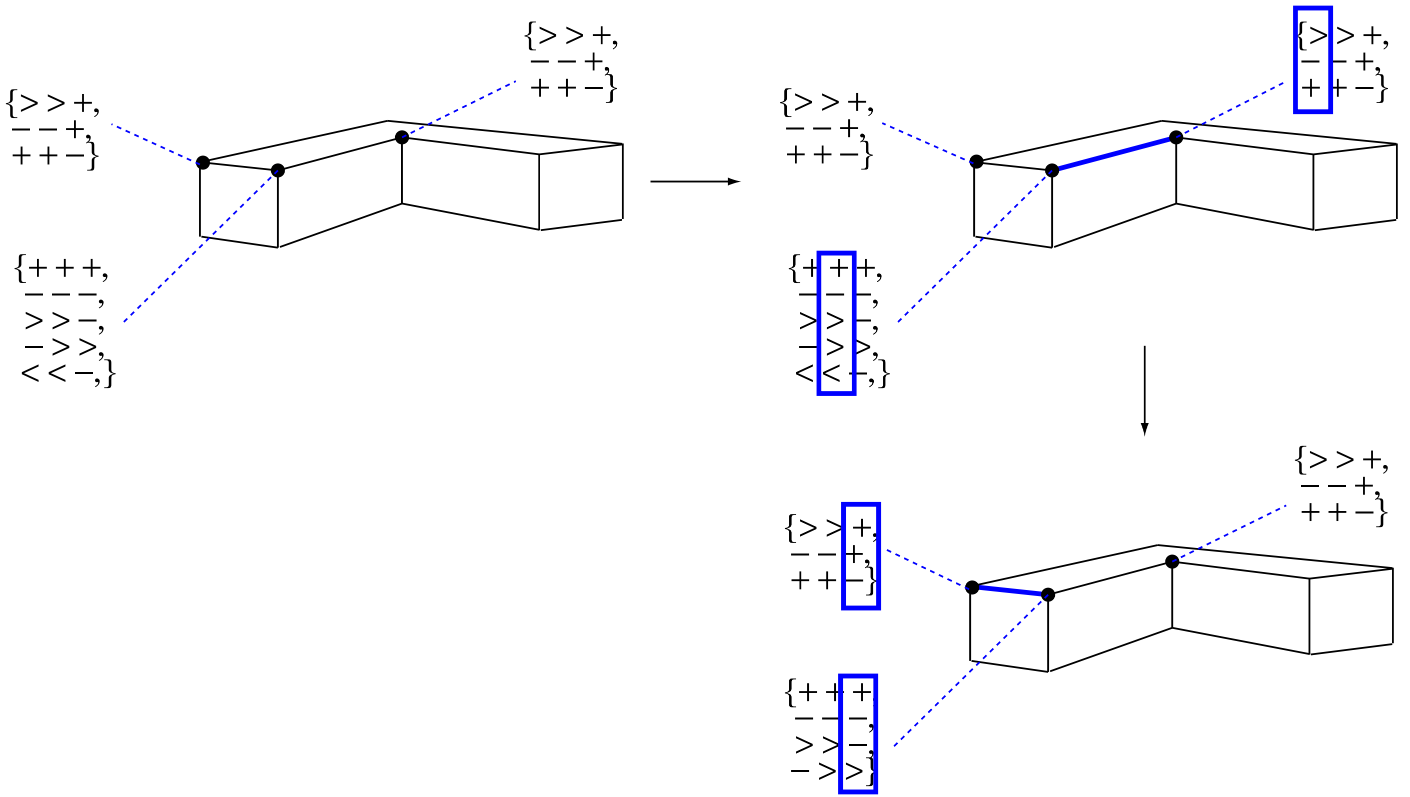
Die Kanten zwischen den Ecken definieren, welche Variablen von zwei verschiedenen Constraints zu unifizieren sind.

➜ Constraint-Netz

Einführung und frühe Systeme

Nicht-algebraische Constraints in Waltz (Fortsetzung)

Waltz-Constraint-Filtern anstatt Backtracking:



Bemerkungen:

- Eventuell führt Waltz-Filtern nicht zu einer eindeutigen Lösung. Warum nicht?
- Was kann man dann tun?

Einführung und frühe Systeme

Gegenüberstellung von EL und Waltz

EL:

- Variablen besitzen Grundbereichen mit unendlich vielen Elementen.
- Gleichungen zur Einschränkung der Constraints notwendig.
- Algebra notwendig, um Gleichungen auszuwerten.

Waltz:

- Symbolische Werte $+, -, <, >$ für die Variablen.
- Grundbereiche aller Constraint-Variablen sind endlich.

➜ Elemente aller Constraint-Relationen sind aufzählbar.

Bemerkungen:

- Die Algebra setzt der Leistungsfähigkeit des Ansatzes enge Grenzen.
- Was bedeutet die Verwendung von iterativen statt geschlossenen Verfahren bei der Lösung von Gleichungssystemen?

Einführung und frühe Systeme

Constraint-Repräsentationsformen

Tabellen.

Funktionen.

- `Normalgewicht = Körpergrösse - 100`, explizit in `Normalgewicht`
- $0 = 2 + 2 \cdot x - 7x^2$, implizit in x

Prädikate.

- Untergewicht: Gewicht / Normalgewicht < 0.8
- OCL: „`context Rechteck inv: a > 0`“

Explizite Darstellung einer Relation.

- $A = \{(1,1,2),(1,2,3),(2,1,3),(4,5,9)\}$
- $B = \{(\texttt{red, blue}),(\texttt{red, green})\}$

Aussagenlogische Formeln.

- $\alpha \wedge \beta \vee \neg\gamma \rightarrow \alpha$

Einführung und frühe Systeme

Constraint-Repräsentationsformen (Fortsetzung)

Extensionale Definition.

Aufzählung aller Elemente einer Menge. [vgl. Lexikon der Linguistik]

Intensionale Definition.

Angabe der charakteristischen Eigenschaft der Elemente einer Menge.

Einführung und frühe Systeme

Einsatz von Constraints

1. Constraints als Repräsentationsformalismus.
   - Repräsentation von Relationen
     (*ungerichtete* Beziehungen zwischen Variablen)
   - insbesondere: Darstellung von *lokalen* Randbedingungen

2. Constraints als Berechnungskonzept.
   Gegeben: Input (Belegung) für eine Teilmenge von Variablen.
   Aufgabe: Berechnung des Outputs (Belegung) der unbekannten Variablen.
   Beispiel: logische Schaltungen

3. Constraints zur effizienten Suchsteuerung.
   Die Komplexität eines Problems wird duch Auswertung *lokaler* Information reduziert.

→ Einschränkung des Lösungsraums für die Variablen.
Stichwort: Constraint-Filtern

Einführung und frühe Systeme

Einsatz von Constraints (Fortsetzung)

Zwei Ziele im Zusammenhang mit Constraint-Problemen:

1. Satisfaction Problem. Gibt es eine Lösung?
2. Solution Problem. Bestimmung einer Lösung, falls eine existiert, unter Beachtung aller Constraints.

Erfolgreiche Einsatzbereiche von Constraints:

- Planen
- Scheduling
- Optimierung
- kombinatorische Fragestellungen
- Probleme mit heterogenen Wissensformen
- Simulation mit qualitativen Modellen
- Puzzles

Bemerkungen:

- Oft impliziert die Lösung von (1) die Lösung von (2).
- Generelle Abkürzung für beide Problemstellungen: CSP
- Ein Algorithmus, der ein Constraint-Problem löst, heißt Constraint-Löser *(Constraint Solver)*.

Konsistenz I

Definition 1 (Constraint, erfüllt, Constraint-Netz)

Sei $X = \{x_1, x_2, \ldots, x_n\}$ eine Menge von n Variablen mit den Grundbereichen *(Domains)* $D_1, D_2, \ldots, D_n$.

1. Ein m-stelliger Constraint C ist definiert durch eine Teilmenge $X_C = \{x_{i_1}, x_{i_2}, \ldots, x_{i_m}\}$ der Variablen X und einer m-stelligen Relation auf den zugehörigen Grundbereichen:

   $$C \subseteq D_C, \quad \text{mit } D_C = D_{i_1} \times D_{i_2}, \ldots, \times D_{i_m}$$

   Dabei sei vereinbart, dass $i_1 < i_2 < \ldots < i_m$ gilt.

Konsistenz I

Definition 1 (Constraint, erfüllt, Constraint-Netz)

Sei $X = \{x_1, x_2, \ldots, x_n\}$ eine Menge von n Variablen mit den Grundbereichen *(Domains)* $D_1, D_2, \ldots, D_n$.

1. Ein m-stelliger Constraint C ist definiert durch eine Teilmenge $X_C = \{x_{i_1}, x_{i_2}, \ldots, x_{i_m}\}$ der Variablen X und einer m-stelligen Relation auf den zugehörigen Grundbereichen:

   $$C \subseteq D_C, \quad \text{mit } D_C = D_{i_1} \times D_{i_2}, \ldots, \times D_{i_m}$$

   Dabei sei vereinbart, dass $i_1 < i_2 < \ldots < i_m$ gilt.

2. Sei C ein Constraint definiert auf X_C. Dann ist C erfüllt mit der Belegung $(d_1, d_2, \ldots, d_n)$, falls gilt:

   $$d_i \in D_i, \; i = 1, 2, \ldots, n \quad \text{und} \quad (d_1, d_2, \ldots, d_n)\mid_{D_C} \in C,$$

   $(d_1, d_2, \ldots, d_n)\mid_{D_C}$ bezeichnet die Projektion von $(d_1, d_2, \ldots, d_n)$ auf D_C.

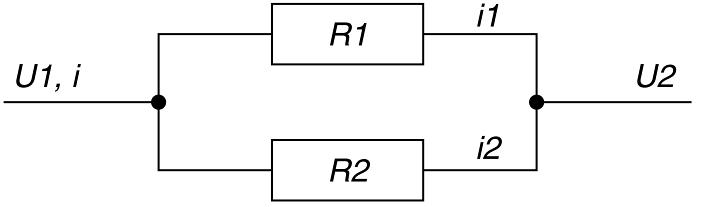
3. Ein Constraint-Netz $\mathcal{C}$ besteht aus einer Menge von Constraints mit einer gemeinsamen Variablenmenge X.

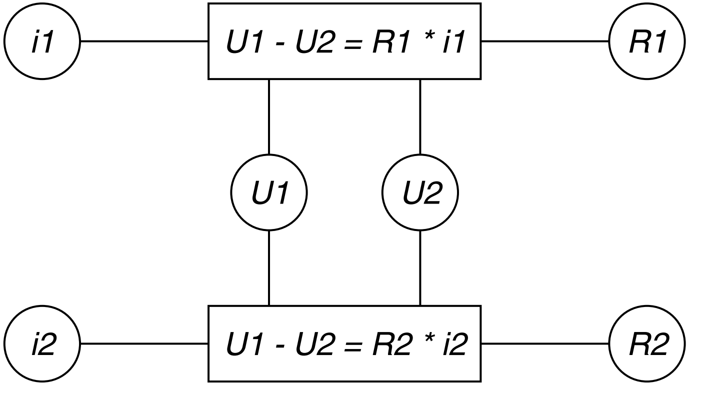
Bemerkungen:

- Einstellige Constraints werden auch als unäre Constraints bezeichnet.
- Zweistellige Constraints werden auch als binäre Constraints bezeichnet.

Konsistenz I

Beispiel für ein algebraisches Constraint-Netz:





- $X = \{i_1, i_2, U_1, U_2, R_1, R_2\}$
- $D_{i_1} = D_{i_2} = \mathbf{R}$
- $D_{U_1} = D_{U_2} = D_{R_1} = D_{R_2} = \mathbf{R}^+$

Konsistenz I

Definition 2 (lokale Konsistenz)

Sei $\mathcal{C} = \{C_1, C_2 \ldots, C_k\}$ eine Menge von Constraints über den Variablen $x_1, x_2, \ldots, x_n$ mit den zugehörigen Grundbereichen $D_1, D_2, \ldots, D_n$.
Weiterhin seien $\hat{D}_1, \hat{D}_2, \ldots, \hat{D}_n$ gegeben mit $\hat{D}_i \subseteq D_i, i = 1, 2, \ldots, n$.

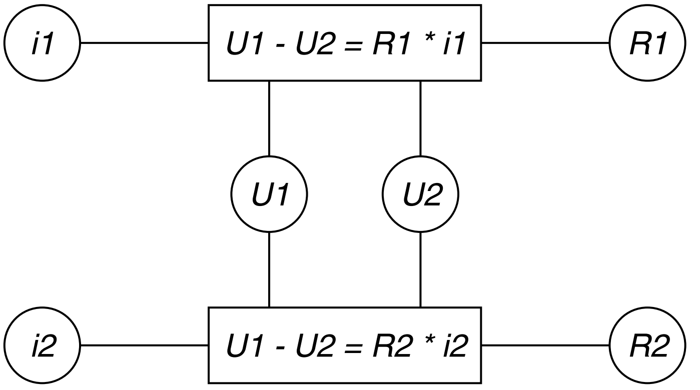
Dann nennt man das Tupel $(\hat{D}_1, \hat{D}_2, \ldots, \hat{D}_n)$ eine lokal konsistente Lösung für $\mathcal{C}$, genau dann wenn gilt:

$$
\begin{gathered}
\forall i \in \{1, 2, \ldots, n\} \\
\forall d \in \hat{D}_i \\
\forall C \in \mathcal{C} \\
\exists d_1 \in \hat{D}_1,\ \exists d_2 \in \hat{D}_2,\ \ldots,\ \exists d_n \in \hat{D}_n :
\end{gathered}
$$

$$
(d_1, d_2, \ldots, d_n)\ |_{D_C} \in C \quad \text{mit } d_i = d
$$

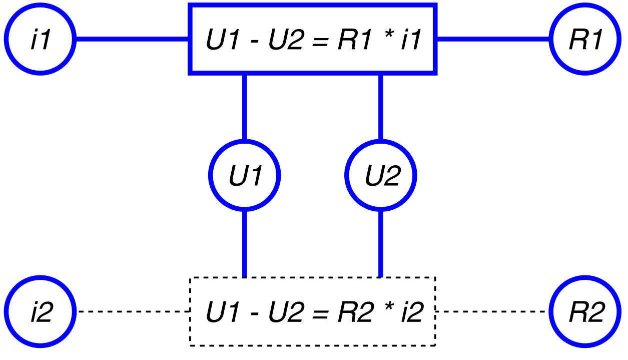
Konsistenz I

Beispiel:

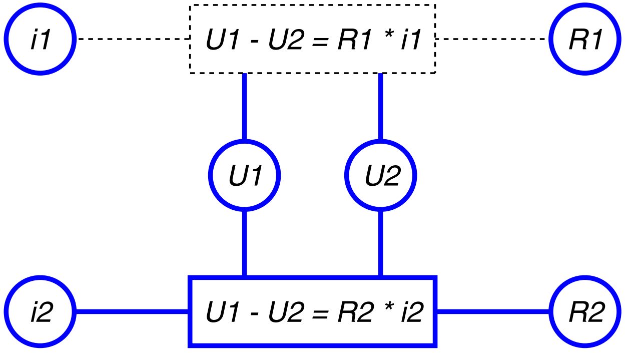


- Zuordnung von Grundbereichen zu Variablen: $(i_1, i_2, U_1, U_2, R_1, R_2)$
- lokal konsistente Lösung: $(\{0.2, 0.6\}, \{0.6\}, \{10\}, \{0, 4\}, \{10, 50\}, \{10, 50\})$

lokale Situation 1:



lokale Situation 2:



Bemerkungen:

- Diese Lösung ist nicht global konsistent: Setzt man i_1 auf 0.6, so findet man keinen passenden Wert für i_2.

Konsistenz I

Definition 3 (globale Konsistenz)

Sei $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$ eine Menge von Constraints über den Variablen $x_1, x_2, \ldots, x_n$ mit den zugehörigen Grundbereichen $D_1, D_2, \ldots, D_n$.
Weiterhin seien $\hat{D}_1, \hat{D}_2, \ldots, \hat{D}_n$ gegeben mit $\hat{D}_i \subseteq D_i, i = 1, 2, \ldots, n$.

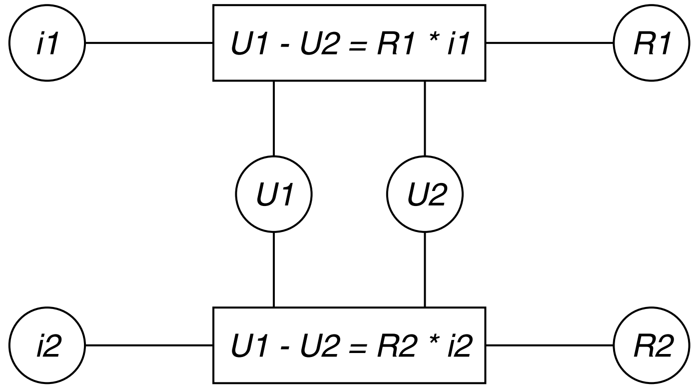
Dann nennt man das Tupel $(\hat{D}_1, \hat{D}_2, \ldots, \hat{D}_n)$ eine global konsistente Lösung für $\mathcal{C}$, genau dann wenn gilt:

$$\forall i \in \{1, 2, \ldots, n\}$$
$$\forall d \in \hat{D}_i$$
$$\exists d_1 \in \hat{D}_1,\ \exists d_2 \in \hat{D}_2,\ \ldots,\ \exists d_n \in \hat{D}_n$$
$$\forall C \in \mathcal{C} :$$

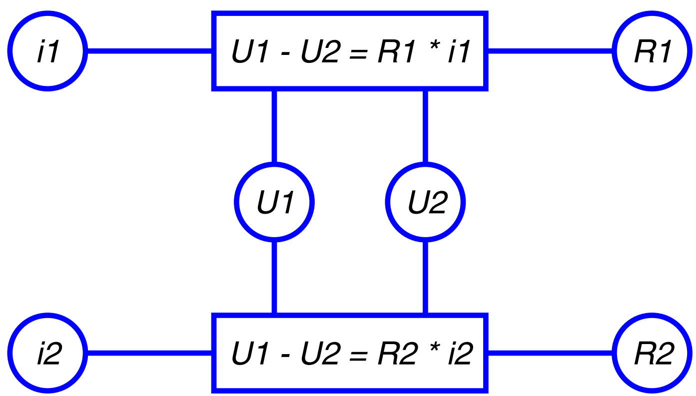
$$(d_1, d_2, \ldots, d_n)\ |_{D_C} \in C \quad \text{mit } d_i = d$$

Konsistenz I

Beispiel:



- Zuordnung von Grundbereichen zu Variablen: $(i_1, i_2, U_1, U_2, R_1, R_2)$
- global konsistente Lösung: $(\{0.1\}, \{0.2\}, \{10\}, \{4\}, \{60\}, \{30\})$



Konsistenz I

Definition 4 (Constraint-Graph)

Sei $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$ eine Menge von Constraints über den Variablen $X = \{x_1, x_2, \ldots, x_n\}$. Ein Constraint-Graph von $\mathcal{C}$ ist ein ungerichteter Graph $G_\mathcal{C} = \langle V_\mathcal{C}, E_\mathcal{C} \rangle$, der wie folgt definiert sein kann.

1. Einfacher Constraint-Graph.
   (a) $V_\mathcal{C} = \{v_1, v_2, \ldots, v_{n+k}\}$ ist eine Menge von $n + k$ Knoten.
   (b) $\varphi : V_\mathcal{C} \to X \cup \mathcal{C}$ ist eine bijektive Abbildung.
   (c) $E_\mathcal{C} = \{\{v, w\} \mid v, w \in V_C,\ \varphi(v) = C, C \in \mathcal{C},\ \varphi(w) \in X_C\}$

2. Kanten-Constraint-Graph.
   (a) $V_\mathcal{C} = \{v_1, v_2, \ldots, v_n\}$ ist eine Menge von n Knoten.
   (b) $\varphi : V_\mathcal{C} \to X$ ist eine bijektive Abbildung.
   (c) $E_\mathcal{C} = \{\{v_{i_1}, v_{i_2}, \ldots, v_{i_m}\} \mid v_{i_j} \in V_C,$
   $\{\varphi(v_{i_1}), \varphi(v_{i_2}), \ldots, \varphi(v_{i_m})\} = X_C,\ C \in \mathcal{C}\}$

Konsistenz I

Definition 4 (Constraint-Graph (Fortsetzung))

Sei $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$ eine Menge von Constraints über den Variablen $X = \{x_1, x_2, \ldots, x_n\}$. Ein Constraint-Graph von $\mathcal{C}$ ist ein ungerichteter Graph $G_{\mathcal{C}} = \langle V_{\mathcal{C}}, E_{\mathcal{C}} \rangle$, der wie folgt definiert sein kann.

3. Knoten-Constraint-Graph.

   (a) $V_{\mathcal{C}} = \{v_1, v_2, \ldots, v_k\}$ ist eine Menge von k Knoten.

   (b) $\varphi : V_{\mathcal{C}} \to \mathcal{C}$ ist eine bijektive Abbildung.

   (c) $E_{\mathcal{C}} = \{\{v, w\} \mid v, w \in V_C,\ v \neq w,$
   $\varphi(v) = C_v,\ \varphi(w) = C_w,\ X_{C_v} \cap X_{C_w} \neq \emptyset,\ C_v, C_w \in \mathcal{C}\}$

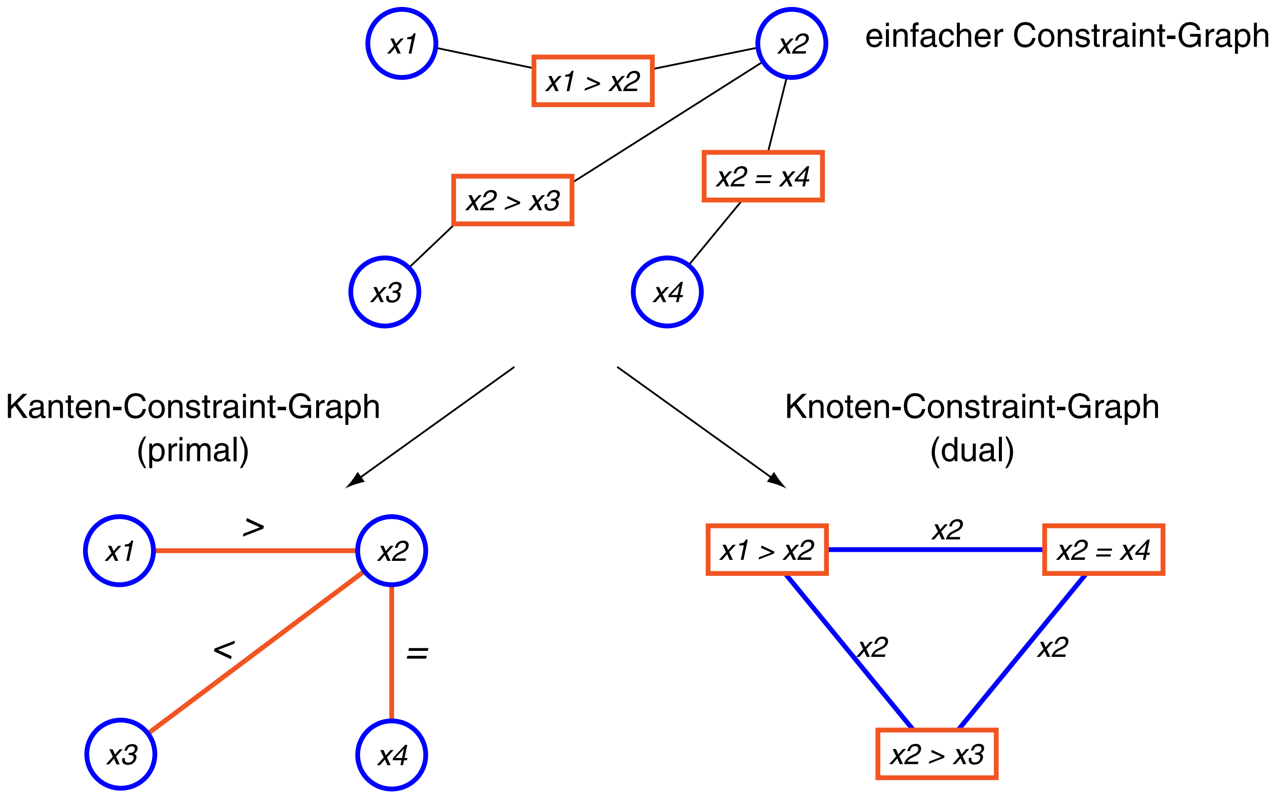
Bemerkungen:

- Der Kanten-Constraint-Graph ist im allgemeinen Fall ein Hypergraph, wobei jeder Knoten einer Variablen entspricht. Eine (Hyper-)Kante repräsentiert die Constraints, die zwischen den Variablen der inzidenten Knoten bestehen.
- Beim Knoten-Constraint-Graph entspricht jeder Knoten einem Constraint. Eine Kante repräsentiert die gemeinsamen Variablen der Constraints der inzidenten Knoten.
- Kanten-Constraint-Graph und Knoten-Constraint-Graph sind dual zueinander. Der Kanten-Constraint-Graph wird auch primaler, der Knoten-Constraint-Graph dualer Constraint-Graph genannt.

Konsistenz I

Beispiel-Constraint-Graph:

- $X = \{x_1, x_2, x_3, x_4\}$
- $D_1 = \{4, 5\},\ D_2 = \{3, 4, 5\},\ D_3 = \{1, 2\},\ D_4 = \{4, 5\}$
- $C_1(x_1, x_2) : x_1 > x_2, \quad C_2(x_2, x_3) : x_2 > x_3, \quad C_3(x_2, x_4) : x_2 = x_4$



Konsistenz I

Satz 5 (hinreichende Konsistenzbedingungen [Freuder 1982]**)**

Sei $\mathcal{C}$ ein Constraint-Netz und sei $(\hat{D}_1, \hat{D}_2, \ldots, \hat{D}_n)$ eine lokal konsistente Lösung von $\mathcal{C}$. Dann gilt:

1. Wenn alle Variablen von $\mathcal{C}$ eindeutig bestimmt sind – d. h., $|\hat{D}_i| = 1,\ i = 1, 2, \ldots, n$,

   oder

2. wenn der einfache Constraint-Graph $G_\mathcal{C}$ von $\mathcal{C}$ keine Zyklen enthält,

dann ist $(\hat{D}_1, \hat{D}_2, \ldots, \hat{D}_n)$ auch eine global konsistente Lösung von $\mathcal{C}$.

