Übungsblatt: Constraints

CSP.01: Logikrätsel

Variablen

Farben:

- Rot, Grün, Weiß, Gelb, Blau

Nationalitäten:

- Engländer, Spanier, Ukrainer, Norweger, Japaner

Haustiere:

- Hund, Schnecken, Fuchs, Pferd, Zebra

Zigarettenmarken:

- Oldgold, Kools, Chesterfield, LuckyStricke, Parliaments

Getränke:

- Kaffee, Tee, Milch, Orangensaft, Wasser

↳ 25 variablen

Wertebereiche (Domänen)

Für alle Variablen gilt:

- $Dom(V) = \{1, 2, 3, 4, 5\}$

Ausnahmen wegen der Hinweise

- $Dom(Norweger) = \{1\}$
- $Dom(Milch) = \{3\}$
- $Dom(Weiß) = \{1, 2, 3, 4\}$
- $Dom(Grün) = \{2, 3, 4, 5\}$

Constraints

Unäre Constraints:

- Norweger = 1
- Milch = 3

Binäre Gleichheits-Constraints:

- Engländer = Rot
- Spanier = Hund
- Kaffee = Grün
- Ukrainer = Tee
- OldGold = Schrecken
- Kools = Gelb
- LuckyStricke = Orangensaft
- Japaner = Parliaments

Binäre Positions-Constraints:

- Grün = Weiß + 1
- |Norweger − Blau| = 1
- |Chesterfield − Fuchs| = 1
- |Kools − Pferd| = 1

Alle verschiedenen Constraints:

- Für jede Kategorie (Farben, Nationalitäten, Haustiere, Getränke, Zigaretten) gilt: Alle Werte sind paarweise verschieden.

CSP.03 : Kantenkonsitenz mit AC-3

Definition

Variablen:

- v_1, v_2, v_3, v_4

Domänen:

- $D = \{0, 1, 2, 3, 4, 5\}$

Constraints:

1. c_1 zwischen v_1 und v_2

   $c_1 = \{(x, y) \mid x + y = 3\}$

2. c_2 zwischen v_2 und v_3

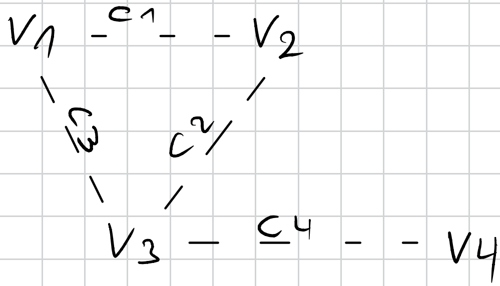
   $c_2 = \{(x, y) \mid x + y \leq 3\}$

3. c_3 zwischen v_1 und v_3

   $c_3 = \{(x, y) \mid x \leq y\}$

4. c_4 zwischen v_3 und v_4

   $c_4 = \{(x, y) \mid x \neq y\}$



Iteration 1 : (v_1, v_2) - Constraints $x+y=3$

Aktuell: $D_{v_1} = D_{v_2} = \{0,1,2,3,4,5\}$

Prüfen:

- $x=0$ braucht $y=3 \in D_{v_2}$ -> ok
- $x=1$ braucht $y=2 \in$ -> ok
- $x=2$ braucht $y=1 \in$ -> ok
- $x=3$ braucht $y=0 \in$ -> ok
- $x=4$ braucht $y=-1$ nicht in D -> 4 raus
- $x=5$ braucht $y=-2$ -> 5 raus

-> neue Domäne $\{0,1,2,3\}$

Queue danach:

$Q = (v_2, v_1), (v_2, v_3), (v_3, v_2), (v_1, v_3), (v_3, v_1), (v_3, v_4), (v_4, v_3)$

Iteration 2: (v_2, v_1) Constraint $c_1: x+y=3$

$D_{v1} = \{0, 1, 2, 3\}$

$D_{v2} = \{0, 1, 2, 3, 4, 5\}$

Prüfen:

- $x = 0$ braucht $y = 3 \in D_{v1} \rightarrow$ ok
- $x = 1$ braucht $y = 2 \in \rightarrow$ ok
- $x = 2$ braucht $y = 1 \in \rightarrow$ ok
- $x = 3$ braucht $y = 0 \in \rightarrow$ ok
- $x = 4$ braucht $y = -1 \rightarrow$ 4 raus
- $x = 5$ braucht $y = -2 \rightarrow$ 5 raus

$\rightarrow$ neue Domäne $D_{v2} = \{0, 1, 2, 3\}$

Queue danach:

$Q = (v_2, v_3), (v_3, v_2), (v_1, v_3), (v_3, v_1), (v_3, v_4), (v_4, v_3)$

Iteration 3 (v_2, v_3) Constraint $c_2: x + y \leq 3$

$D_{v_2} = \{0, 1, 2, 3\}$

$D_{v_3} = \{0, 1, 2, 3, 4, 5\}$

Prüfen:

- $x = 0$ wähle $y = 3 \rightarrow 0 + 3 \leq 3 \rightarrow$ ok
- $x = 1$ wähle $y = 2 \rightarrow 1 + 2 \leq 3 \rightarrow$ ok
- $x = 2$ wähle $y = 1 \rightarrow 2 + 1 \leq 3 \rightarrow$ ok
- $x = 3$ wähle $y = 0 \rightarrow 3 + 0 \leq 3 \rightarrow$ ok

↳ kein Wert wird entfernt

↳ Domänen bleiben gleich

Queue:

$Q = (v_3, v_2), (v_1, v_3), (v_3, v_1), (v_3, v_4), (v_4, v_3)$

Iteration 4 (v_3, v_2) Constraint $c_2: x + y \leq 3$

$D_{v_3} = \{0, 1, 2, 3, 4, 5\}$

$D_{v_2} = \{0, 1, 2, 3\}$

Prüfen:

- $x = 0$ wähle $y = 0 \rightarrow$ ok
- $x = 1$ wähle $y = 1 \rightarrow$ ok
- $x = 2$ wähle $y = 1 \rightarrow$ ok
- $x = 3$ wähle $y = 0 \rightarrow$ ok
- $x = 4$ braucht $y = -1 \rightarrow$ 4 raus
- $x = 5$ braucht $y = -2 \rightarrow$ 5 raus

$\rightarrow$ neue Domäne $D_{v_3} = \{0, 1, 2, 3\}$

Queue danach:

$Q = (v_1, v_3), (v_3, v_1), (v_3, v_4), (v_4, v_3)$

Iteration 5 (v_1, v_3) Constraint c_3 $x \leq y$

$D_{v_1} = \{0, 1, 2, 3\}$

$D_{v_3} = \{0, 1, 2, 3\}$

Prüfen:

- $x = 0$ wähle $y = 0$ → ok
- $x = 1$ wähle $y = 3$ → ok
- $x = 2$ wähle $y = 2$ → ok
- $x = 3$ wähle $y = 3$ → ok

↳ Domänen bleiben gleich

Queue danach:

$Q = (v_3, v_1), (v_3, v_4), (v_4, v_3)$

Iteration 6 (v_3, v_1) Constraint c_3 $x \leq y$

$D_{v_3} = \{0, 1, 2, 3\}$

$D_{v_1} = \{0, 1, 2, 3\}$

Prüfen:

- $x = 0$ wähle $y = 0$ → ok
- $x = 1$ wähle $y = 3$ → ok
- $x = 2$ wähle $y = 2$ → ok
- $x = 3$ wähle $y = 3$ → ok

↳ Domänen bleiben gleich

Queue danach:

$Q = (v_3, v_4), (v_4, v_3)$

Iteration 7 (v_3, v_4) Constraint $c_4 : x \neq y$

$D_{v_3} = \{0, 1, 2, 3\}$

$D_{v_4} = \{0, 1, 2, 3, 4, 5\}$

Da D_{v_4} immer mehrere Werte enthält, gibt es immer mindestens einen ungleichen Wert.

Queue danach:

$Q = (v_4, v_3)$

Iteration 8 (v_4, v_3) Constraint $c_4 : x \neq y$

$D_{v_4} = \{0, 1, 2, 3, 4, 5\}$

$D_{v_3} = \{0, 1, 2, 3\}$

Da D_{v_3} vier Werte hat, gibt es für jeden Wert von v_4 einen ungleichen.

z.B $x = 0$ $y = 1 \rightarrow$ ok

Queue danach:

$Q = \emptyset$

AC-3 stoppt.

Endergebnis der Domänen nach AC-3

$D_{v_1} = \{0, 1, 2, 3\}$

$D_{v_2} = \{0, 1, 2, 3\}$

$D_{v_3} = \{0, 1, 2, 3\}$

$D_{v_4} = \{0, 1, 2, 3, 4, 5\}$

CSP.04 Forward Checking und Kantenkonsistenz

Gegeben $\alpha = \{ v_1 \rightarrow 2 \}$

- $D_{v1} = \{2\}$
- $D_{v2} = D_{v3} = D_{v4} = \{0, 1, 2, 3, 4, 5\}$

1) Kantenkonsistenz in α

1. $c1 = (v_1, v_2)$, $x + y = 3$

   $v_1 = 2 \rightarrow v_2 = 1 \rightarrow D_{v2} = \{1\}$

2. $c3 = (v_1, v_3)$, $x \leq y$

   $2 \leq v3 \rightarrow D_{v3} = \{2, 3, 4, 5\}$

3. v4 ist nicht direkt mit v_1 verbunden, deshalb keine Änderung.

Domains nach Kantenkonsistenz:

- $D_{v1} = \{2\}$
- $D_{v2} = \{1\}$
- $D_{v3} = \{2, 3, 4, 5\}$
- $D_{v4} = \{0, 1, 2, 3, 4, 5\}$

Forward-Checking in α

Forward Checking betrachtet nur die direkten Constraints von $v1$.

1. $c1: v1 + v2 = 3$
   ↳ $v_2 = 1 \rightarrow D_{v_2} = \{1\}$
2. $c3: v1 \leq v3$
   ↳ $D_{v3} = \{2,3,4,5\}$
3. $v4$ ist kein direkter Nachbar
   ↳ D_{v4} bleibt unverändert

Domains nach Forward Checking:

$D_{v1} = \{2\}$

$D_{v2} = \{1\}$

$D_{v3} = \{2,3,4,5\}$

$D_{v4} = \{0,1,2,3,4,5\}$

Vergleich:

Beide Verfahren ergeben den selben Zustand.

- Kantenkonsistenz propagiert theoretisch weiter
- Forward-Checking schränkt nur Nachbarn einer belegten variable ein.

CSP.05: Planung von Indoor-Spielplätzen

Vereinfachung:

- kleines Raster 10 x 6 Felder
- Bar (B), Größe 3x2
- Hüpfburg (H), Größe 3x2
- Kletterberg (K), Größe 2x2
- Eingang unten links, Zelle (0,0)
- Notausgang unten rechts, Zelle (9,0)

Variablen:

Für jedes Objekt nur die linke untere Ecke (x,y)

- Bx, By - Bar
- Hx, Hy - Hüpfburg
- Kx, Ky - Kletterberg

Domänen:

Raster ist $x \in \{0, \dots, \dots, 9\}$ $y \in \{0, \dots, \dots, 5\}$

- Bar (3x2)

  $B_x \in \{0, \dots, \dots, 7\}$, $B_y \in \{0, \dots, \dots, 4\}$

- Hüpfburg (3x2)

  $H_x \in \{0, \dots, \dots, 7\}$, $H_y \in \{0, \dots, \dots, 4\}$

- Kletterberg (2x2)

  $K_x \in \{0, \dots, \dots 8\}$, $K_y \in \{0, \dots, \dots, 4\}$

Constraints

Für jedes paar (B, H), (B, K), (H, K)

- die Rechtecke müssen mindestens ein Feld auseinander liegen

z.B. für B und H

- B ist links von H inkl. Abstand

  $\hookrightarrow B_x + 3 + 1 \leq H_x$

- oder B ist unter H

  $\hookrightarrow B_y + 2 + 1 \leq H_y$

Selbes Schema für (B, K) und (H, K)

Bar nahe am Eingang

- Eingang ist bei $(0, 0)$
- Bar steht in einem der linken Felder unten:

  $B_y = 0$ und $B_x \in \{0, 1, 2, 3\}$

Notausgang bei $(9, 0)$

- kein Gerät darf hier liegen, wird durch die Domänen sichergestellt

Sichtlinie Bar → Kletterberg

- Bar und Kletterberg sollen in derselben Zeile stehen und die Bar links vom Kletterberg:

  ↳ $B_y = K_y$ und $B_x + 3 < K_x$