

Aufgabe 02

Der Graph:

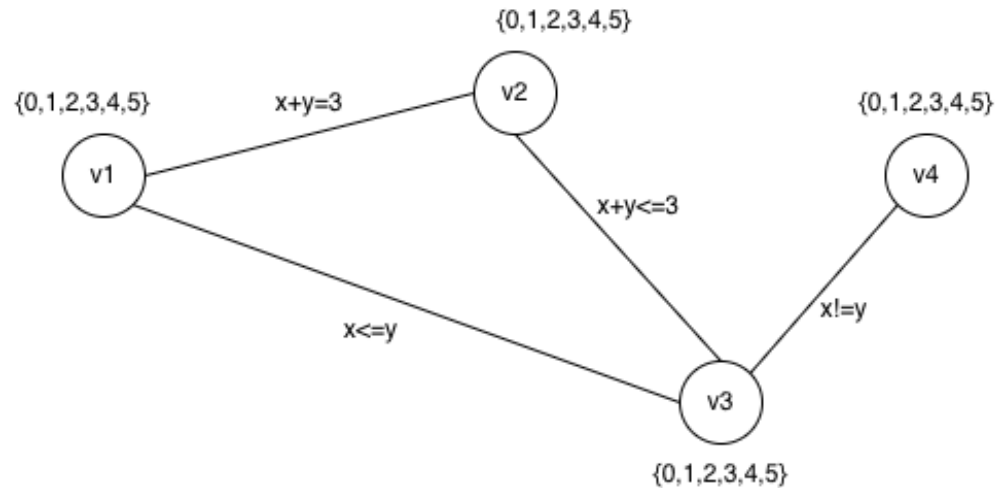
$\{v1, v2, v3, v4\}, \{Dv1 = Dv2 = Dv3 = Dv4\}, \{C1, C2, C3, C4\}$

$C1 = ((v1, v2), \{(x, y) \in D^2 \mid x+y=3\})$

$C2 = ((v2, v3), \{(x, y) \in D^2 \mid x+y \leq 3\})$

$C3 = ((v1, v3), \{(x, y) \in D^2 \mid x \leq y\})$

$C4 = ((v3, v4), \{(x, y) \in D^2 \mid x \neq y\})$



Handsimulation:

Methode AC3(Parameter: Constraints)

Schritt 1: Queue aufbauen, alle Kanten Richtung beachten! (keine Dopplungen!)

Schritt 2: Alle Elemente der Queue durchlaufen, bis sie leer ist

in Schleife:

eine Kante x, y aus Queue entfernen

rufe für diese Kante reduce-Methode auf:

wenn reduce-Methode true:

wenn die Domäne leer ist: return false

wenn in Domäne von x Änderungen stattfanden: alle Rückrichtungen betrachten

-> alle Nachbarn z in Queue packen (mit Schleife)

[//die werden in dieselbe Queue gepackt, keine neue Queue! Queue: hinten anhängen]

Return true

Methode ARC_Reduce(Constraints, Kante x, y)

Schritt 1: per default ist Veränderung false (change)

Schritt 2: mit einer Schleife gehen wir alle Werte der Domäne von x durch:

wenn aktueller Wert aus Domäne x keinen Partner in Domäne y hat, mit dem der Constraint erfüllt wird:

Entferne aktuellen Wert aus Domäne x

setze Veränderung auf true

Return Veränderung

Also: es gab eine Veränderung (oder mehrere) in x. Jetzt werden alle Rückrichtungen geprüft. Dies geschieht so lange, bis die Queue leer ist d.h. bis alle Veränderungen abgearbeitet sind.

Durchlauf 1:

Queue: $\{(v1, v2), (v2, v1), (v2, v3), (v3, v2), (v1, v3), (v3, v1), (v3, v4), (v4, v3)\}$

aktuelle Kante: $(v1, v2)$ beide Domänen $\{0, 1, 2, 3, 4, 5\}$, Constr. : $x+y=3$

reduce:

X 0 Parter in y: 3

X 1 Parter in y: 2

X 2 Parter in y: 1

X 3 Parter in y: 0

X 4 Parter in y: -

X 5 Parter in y: -

Entfernt werden 4 und 5. Es gab Änderungen, also wird die Queue aktualisiert mit $(v2, v1), (v3, v1)$:

Queue: $\{(v2, v1), (v2, v3), (v3, v2), (v1, v3), (v3, v1), (v3, v4), (v4, v3), (v3, v1)\}$

$(v2, v1)$ ist doppelt, wird nicht aufgenommen.

Domäne Dv1: $\{0,1,2,3\}$

Durchlauf 2:

aktuelle Kante: $(v2, v1)$, Dv2 $\{0,1,2,3,4,5\}$ Dv1: $\{0,1,2,3\}$ Constraint: $x+y=3$

reduce:

X 0 Parter in y: 3

X 1 Parter in y: 2

X 2 Parter in y: 1

X 3 Parter in y: 0

X 4 Parter in y: -

X 5 Parter in y: -

Domäne Dv2: $\{0,1,2,3\}$

Entfernt werden 4 und 5. Es gab Änderungen, also wird die Queue aktualisiert mit $(v1, v2), (v3, v2)$

$\{(v2, v3), (v3, v2), (v1, v3), (v3, v1), (v3, v4), (v4, v3), (v3, v1), (v1, v2)\}$ [(v3, v2) ist schon in Queue]

Durchlauf 3:

Aktuelle Kante: $(v2, v3)$, Dv2 $\{0,1,2,3\}$ Dv3 $\{0,1,2,3,4,5\}$ Constraint: $x+y \leq 3$

reduce:

X 0 Parter in y: 0,1,2,3

X 1 Parter in y: 0,1,2

X 2 Parter in y: 0,1

X 3 Parter in y: 0

X 4 Parter in y: -

X 5 Parter in y: -

Domäne Dv3: $\{0,1,2,3\}$

Entfernt werden 4 und 5. Es gab Änderungen, also wird die Queue aktualisiert mit (v1, v2) (v3, v2)
Queue: {(v3, v2), (v1, v3), (v3, v1), (v3, v4), (v4, v3), (v3, v1), (v1, v2)} beide existieren schon in Queue

Durchlauf 4:

Aktuelle Kante: (v3, v2) Dv3 {0,1,2,3} Dv2 {0,1,2,3} Constraint: $x+y \leq 3$

X 0 Parter in y: 0,1,2,3

X 1 Parter in y: 0,1,2

X 2 Parter in y: 0,1,

X 3 Parter in y: 0

Keine Änderungen in Domains.

Queue: {(v1, v3), (v3, v1), (v3, v4), (v4, v3), (v3, v1), (v1, v2)}

Durchlauf 5:

Aktuelle Kante: (v1, v3) Dv1 {0,1,2,3}, Dv3 {0,1,2,3} Constraint: $x \leq y$

X 0 Parter in y: 0,1,2,3

X 1 Parter in y: 1,2,3

X 2 Parter in y: 2, 3

X 3 Parter in y: 3

Keine Änderung in Domains.

Queue: : {(v3, v1), (v3, v4), (v4, v3), (v3, v1), (v1, v2)}

Durchlauf 6 ist die Gegenrichtung (v3, v1). Keine Änderungen, da Constraint $x \leq y$. Die Werte werden so aussehen wie in Durchlauf 5.

Queue: {(v3, v4), (v4, v3), (v3, v1), (v1, v2)}

Durchlauf 7:

Aktuelle Kante: (v3, v4), Dv3 {0,1,2,3} Dv4 {0,1,2,3,4,5}, Constraint: $x \neq y$

X 0 Parter in y: 1,2,3,4,5

X 1 Parter in y: 0,2,3,4,5

X 2 Parter in y: 0,1,3,4,5

X 3 Parter in y: 0,1,2,4,5

Keine Änderung in Domains.

Queue: : {(v4, v3), (v3, v1), (v1, v2)}

Durchlauf 8:

Aktuelle Kante: (v4, v3), Dv4 {0,1,2,3,4,5}, Dv3 {0,1,2,3}, Constraint: $x \neq y$

X 0 Parter in y: 1,2,3

X 1 Parter in y: 0,2,3

X 2 Parter in y: 0,1,3

X 3 Parter in y: 0,1,2

X 4 Parter in y: 0,1,2,3

X 5 Parter in y: 0,1,2,3

Keine Änderung in Domains.

Queue: : {(v3, v1), (v1, v2)}

Durchlauf 9:

Aktuelle Kante: (v3, v1), Dv3 {0,1,2,3}, Dv1 {0,1,2,3}, Constraint: $x \leq y$

Die beiden Mengen sind gleich. Es gibt für jeden mindestens einen passenden y-Wert.

Keine Änderungen.

Queue: : {(v1, v2)}

Durchlauf 10:

Aktuelle Kante: (v1, v2), Dv1 {0,1,2,3}, Dv2{0,1,2,3}, Constraint: $x+y=3$

X 0 Parter in y: 3

X 1 Parter in y: 2

X 2 Parter in y: 1

X 3 Parter in y: 0

Keine Änderung in Domains.

Queue: {}

Abbruch von AC-3.

