Praktikum 5 GKI

CSP.01: Logikrätsel (Zebrarätsel)

CSP (V, D, C)

V:

$i \in \{1,2,3,4,5\}$

Farbe $= F_i$

Nationalität $= N_i$

Haustier $= HT_i$

Getränk $= G_i$

Zigaretten $= Z_i$

D:

D_F = { blau, gelb, rot, grün, weiß }

D_N = { Ukrainer, Engländer, Japaner, Spanier }

D_{HT} = { Pferd, Fuchs, Schnecken, Zebra, Hund }

D_G = { Tee, Wasser, O-Saft, Kaffee }

D_Z = { Kools, Old Gold, Chesterfield, Parliament, Lucky Strike }

D_{N_1} = { Norweger }

D_{G_3} = { Milch }

(Hinweise im Text => Regeln)

C: $\forall$ i und j | i $\neq$ j und $\in \{1,2,3,4,5\}$

$c_1 = F_i \neq F_j$

$c_2 = N_i \neq N_j$

$c_3 = HT_i \neq HT_j$

$c_4 = G_i \neq G_j$

$c_5 = Z_i \neq Z_j$

$c_6 = N_i =$ Engländer $\Rightarrow F_i =$ rot

$c_7 = N_i =$ Spanier $\Rightarrow HT_i =$ Hund

$c_8 = G_i =$ Kaffee $\Rightarrow F_i =$ grün

$c_9 = N_i =$ Ukrainer $\Rightarrow G_i =$ Tee

$c_{10} = F_i =$ grün $\Rightarrow F_{i-1} =$ weiß

$c_{11} = Z_i =$ Old Gold $\Rightarrow HT_i =$ Schnecken

$c_{12} = Z_i =$ Kools $\Rightarrow F_i =$ gelb

$c_{13} = G_3 =$ Milch

$c_{14} = N_1 =$ Norweger

$c_{15} = Z_i =$ Chesterfield $\Rightarrow HT_{i-1} =$ Fuchs $i \neq 1$

$c_{16} = Z_i =$ Kools $\Rightarrow HT_{i+1} =$ Pferd oder $HT_{i-1} =$ Pferd für $i \neq 1$

$c_{17} = Z_i =$ Lucky Strike $\Rightarrow G_i =$ O-Saft

$c_{18} = N_i =$ Japaner $\Rightarrow Z_i =$ Parliament

$c_{19} = N_i =$ Norweger $\Rightarrow F_{i+1} =$ blau oder $F_{i-1} =$ blau für $i \neq 1$

CSP. 02

Keine mehrfache Belegung durch globale Constraints erlaubt!

{blau, gelb, rot, grün, weiß}

Fehler, da mit einem Assignment gearbeitet wurde.

1.

F_1 = blau Konflikt! [~~blau~~, gelb, rot, grün, weiß]

F_1 = gelb [blau, ~~gelb~~, rot, grün, weiß]

F_2 = blau [~~blau~~, rot, grün, weiß]

F_3 = rot [~~rot~~, grün, weiß]

F_4 = grün [~~grün~~, weiß]

F_5 = weiß Konflikt! [~~weiß~~]

2 Schritte zurück

F_4 = weiß [grün, ~~weiß~~]

F_5 = grün [~~grün~~]

N_2 = Ukrainer [~~Ukrainer~~, Spanier, Engländer, Japaner]

N_3 = Spanier [~~Spanier~~, Engländer, Japaner]

N_4 = Engländer Konflikt! [~~Engländer~~, Japaner]

N_4 = Japaner [Engländer, ~~Japaner~~]

N_5 = Engländer Konflikt! [~~Engländer~~]

[Engländer]

[Engländer, Japaner]

[Spanier, Engländer, Japaner]

N_3 = Engländer [~~Engländer~~, Spanier, Japaner]

N_4 = Spanier [~~Spanier~~, Japaner]

N_5 = Japaner [~~Japaner~~]

HT_1 = Pferd Konflikt! [~~Pferd~~, Fuchs, Schnecken, Zebra, Hund]

HT_1 = Fuchs [Pferd, ~~Fuchs~~, Schnecken, Zebra, Hund]

HT_2 = Pferd [~~Pferd~~, Schnecken, Zebra, Hund]

HT_3 = Schnecken [~~Schnecken~~, Zebra, Hund]

HT_4 = Zebra Konflikt! [~~Zebra~~, Hund]

HT_4 = Hund [Zebra, ~~Hund~~]

HT_5 = Zebra [Zebra]

G_1 = Tee [~~Tee~~, Wasser, O-Saft, Kaffee]

G_2 = Wasser Konflikt! [~~Tee~~, Wasser, O-Saft, Kaffee]

Wasser und Kaffee auch Konflikt!

G_1 = Wasser

G_2 = Tee

G_3 = Milch Vorgabe!

G_4 = O-Saft

G_5 = Kaffee

Z_1 = Kools [~~Kools~~, Old Gold, Chesterfield, Parliament, Lucky Strike]

Z_2 = Old Gold Konflikt! [~~Kools~~, ~~Old Gold~~, Chesterfield, Parliament, Lucky Strike]

Z_2 = Chesterfield [~~Kools~~, Old Gold, ~~Chesterfield~~, Parliament, Lucky Strike]

Z_3 = Old Gold [~~Kools~~, ~~Old Gold~~, ~~Chesterfield~~, Parliament, Lucky Strike]

Z_4 = Parliament [~~Kools~~, ~~Old Gold~~, ~~Chesterfield~~, ~~Parliament~~, Lucky Strike]

Z_5 = Lucky-Strike Konflikt! [~~Kools~~, ~~Old Gold~~, ~~Chesterfield~~, ~~Parliament~~, ~~Lucky Strike~~]

Z_4 = Luck-Strike

Z_5 = Parliament

2.2

MRV (Variable mit den wenigsten freien Werten zuerst betrachten)
(Betrachten immer Nachbarn nach Belegung aus Sicht der Selekten Variablen)

Gradheuristik (Erweiterung von MRV bei Gleichstand)
Szenario MRV: Variablen mit gleiche Anzahl an freien Werten
→ Wähle Variable mit meisten Constraints auf offene Variablen (Einfach die Anzahl der Kanten)

N_1 = Norwegen {Norwegen}
G_3 = Milch {Milch}
G_1 = Tee {~~Tee~~, Wasser, O-Saft, Kaffee} (Keine Belegungen vorhanden, deshalb ohne Wahl)
G_2 = Wasser {~~Wasser~~, O-Saft, Kaffee}
G_4 = O-Saft {~~O-Saft~~, Kaffee}
G_5 = Kaffee {~~Kaffee~~}

N_2 = Ukrainer Konflikt! {Ukrainer, Engländer, Japaner, Spanier}
N_2 = Engländer
N_3 = Ukrainer Konflikt!
N_3 = Spanier
N_4 = Ukrainer Konflikt!

Backtracking mit Tiefensuche arbeitet sehr ineffizient und MRV sowie Gradheuristik,
tragen kaum einen Unterschied in meiner Modellierung von der Laufzeit und den Ergebnissen.
In meiner Modellierung sorgen sie für eine noch schlechtere Laufzeit.

2.3

(AC-3 kürzt Domänen)

$C_{14} = N_1 =$ Norwegen

$C_{13} = G_3 =$ Milch

$C_{19} = N_i =$ Norwegen $\Rightarrow F_{i+1} =$ blau oder $F_{i-1} =$ blau für $i \neq 1$

$(G_3, N_3), (G_3, F_3)$ (G_3, Z_3) (N_1, N_2)

$(N_3, G_3), (F_3, G_3), (Z_3, G_3)$

$(N_1, G_1), (N_1, F_1), (N_1, Z_1), (N_1, HT_1)$

$(G_1, N_1), (F_1, N_1), (Z_1, N_1), (HT_1, N_1)$

Modellierung

Modellierungstechnisch falsch in meiner Herangehensweise, bietet jedoch Übersicht, wenn man die nötige Iteration kennt.

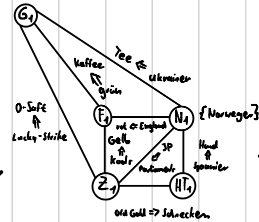
Norwegen neben blauen Haus (+1/-1)

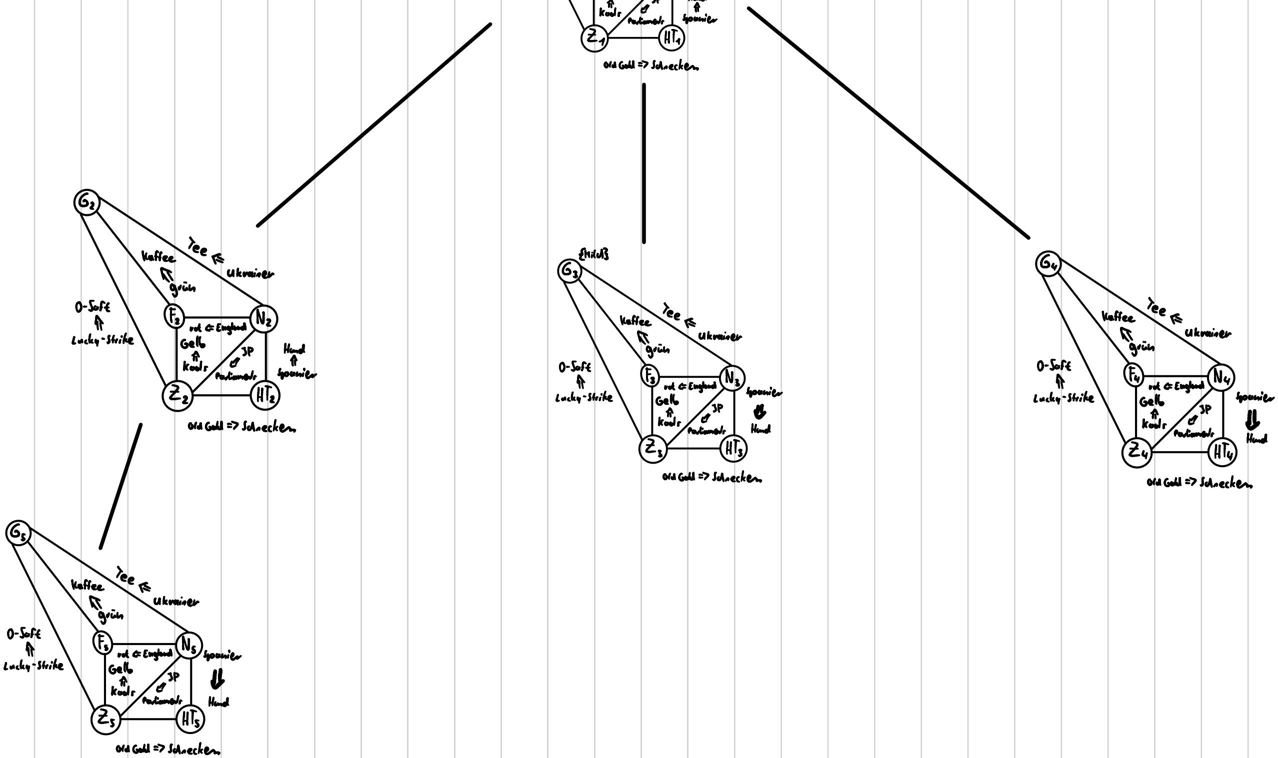
(Haus 2 oder 5)

Grüne Haus ist direkt rechts vom weißen Haus
(Also +1)

Chesterfield Raucher neben Mann mit Fuchs (+1/-1)

Kools Raucher neben Haus mit Pferd (+1/-1)

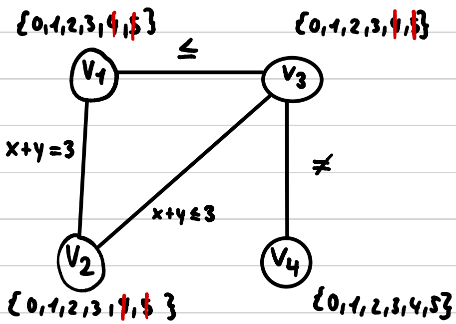




Erklärung Algo:

- Tiefensuche mit Backtracking
- Start mit unbelegtem CSP-Assignement
- Auswahl einer noch nicht belegten Variable
- Reihe nach alle Werte aus ihrer Domäne ausprobieren
- consistent(...)
↳ Prüfen vor dem Eintragen, ob var = value mit dem bisherigen assignement verträglich ist (belegte Variablen nicht verletzt)
↳ Wenn consistent true wird var = value als Assignment übernommen
- Inferenz
↳ Nachbarn anschauen
↳ Werte rausschneißen, die mit var = value nicht kompatibel sind
↳ Nachbar Domäne leer? Failure!
↳ Wenn Inference(...) != failure
  ↳ Heißt keine Domäne leer
  ↳ + die aktuelle Belegung sowie aller bisher abgeleiteten Einschränkungen sind konsistent
  ↳ Wir gehen tiefer in die Suche mit dieser Belegung und den ggf. beschnittenen Domänen
  ↳ Rekursion führt zur Sackgasse und wir kommen zurück → assignement -= {var = value}
  (Domänen rückgängig machen)
- return result ← Lösung

CSP. 03



$D \times D$ = (0,0),(0,1),(0,2),(0,3),(0,4),(0,5)
(1,0),(1,1),(1,2),(1,3),(1,4),(1,5)
(2,0),(2,1),(2,2),(2,3),(2,4),(2,5)
(3,0),(3,1),(3,2),(3,3),(3,4),(3,5)
(4,0),(4,1),(4,2),(4,3),(4,4),(4,5)
(5,0),(5,1),(5,2),(5,3),(5,4),(5,5)

C_1 = (0,3), (1,2), (2,1), (3,0)
$D_{v_1} = \{0,1,2,3\}$
$D_{v_2} = \{0,1,2,3\}$
$D_{v_3} = \{0,1,2,3,4,5\}$
$D_{v_4} = \{0,1,2,3,4,5\}$

Wir brauchen mindestens ein Wert in der Domäne, welche die Bedingung erfüllt.

Queue = $[(v_1,v_2), (v_2,v_1), (v_1,v_3), (v_3,v_1), (v_2,v_3), (v_3,v_2), (v_3,v_4), (v_4,v_3)]$

$[(v_1,v_2), (v_2,v_1), (v_1,v_3), (v_3,v_1), (v_2,v_3), (v_3,v_2), (v_3,v_4), (v_4,v_3)]$
$[\cancel{(v_1,v_2)}, (v_2,v_1), (v_1,v_3), (v_3,v_1), (v_2,v_3), (v_3,v_2), (v_3,v_4), (v_4,v_3)]$

für 4,5 = change = true $\quad D_{v_1} = \{0,1,2,3\}$

Nachbarn:

(v_2,v_1)

(v_3,v_1)

Wird hinten jetzt angehängt und doppelte Elemente kommen in einer Menge nicht vor

$[\cancel{(v_1,v_3)}, (v_2,v_3), (v_3,v_2), (v_3,v_4), (v_4,v_3), (v_2,v_1), (v_3,v_1)]$

Keine Werte gestrichen change = false $D_{v_1} = \{0,1,2,3\}$ Kein ARC_Reduce und keine Nachbarn dranhängen

$[\cancel{(v_2,v_3)}, (v_3,v_2), (v_3,v_4), (v_4,v_3), (v_2,v_1), (v_3,v_1)]$

für 4,5 = change = true $D_{v_2} = \{0,1,2,3\}$

Nachbarn:

(v_3,v_2)

(v_3,v_1)

$[\cancel{(v_3,v_4)}, (v_4,v_3), (v_2,v_1), (v_3,v_2), (v_3,v_1)]$

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

$[\cancel{(v_4,v_3)}, (v_2,v_1), (v_3,v_2), (v_3,v_1)]$

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

$[\cancel{(v_2,v_1)}, (v_3,v_2), (v_3,v_1)]$

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

$[\cancel{(v_3,v_2)}, (v_3,v_1)]$

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

Nachbarn:

(v_1,v_3)

(v_2,v_3)

(v_4,v_3)

$[\cancel{(v_3, v_1)}, (v_1, v_3), (v_2, v_3), (v_4, v_3)]$

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

$[\cancel{(v_1, v_3)}, (v_2, v_3), (v_4, v_3)]$

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

$[(v_2, v_3), (v_4, v_3)]$

change = false

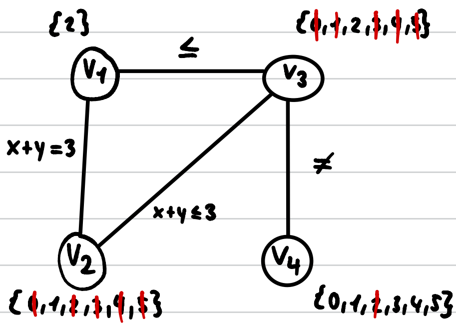
$[(v_4, v_3)]$

change = false

$[\,]$

CSP.04

1.



$$D \times D = (0,0),(0,1),(0,2),(0,3),(0,4),(0,5)$$
$$(1,0),(1,1),(1,2),(1,3),(1,4),(1,5)$$
$$(2,0),(2,1),(2,2),(2,3),(2,4),(2,5)$$
$$(3,0),(3,1),(3,2),(3,3),(3,4),(3,5)$$
$$(4,0),(4,1),(4,2),(4,3),(4,4),(4,5)$$
$$(5,0),(5,1),(5,2),(5,3),(5,4),(5,5)$$

$C_1 = (0,3),(1,2),(2,1),(3,0)$

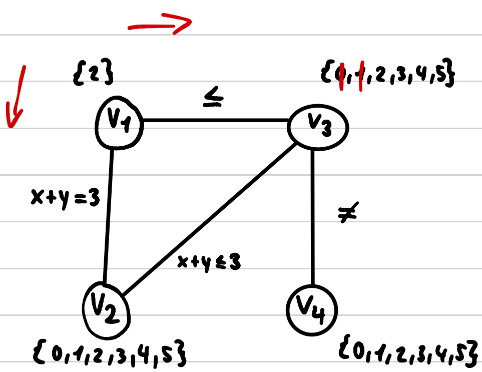
$D_{v_1} = \{2\}$ Änderung!

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

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

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

2.



Im Vergleich zu 1. prüfen wir nur aus einer Perspektive und vernachlässigen bestehende Inkonsistenzen. (Konsistenz nur für alle gerade betrachteten (belegten) Variablen) (einseitig)

$\alpha = \{v_1 \to 2\}$

CSP.05

Hüpfburgen = HB
Go-Kart-Bahn = GKB
Kletterberg = KB
Bar = BAR

CSP(V,D,C)

V:

GKB
KB
HB
BAR

Sie sind für die Planung von Indoor-Spielplätzen zuständig.

Ein Spielplatz hat eine rechteckige Form, etwa 40x100 m. Zur Vereinfachung wird diese Fläche in ein gleichmäßiges Raster unterteilt, beispielsweise 10x10 cm. Es gibt am Rand mehrere Türen (normaler Eingang, Notausgänge).

Auf dieser Grundfläche sollen verschiedene Spielgeräte angeordnet werden, beispielsweise Go-Kart-Bahnen, Hüpfburgen und Kletterberge. Diese Spielgeräte haben selbst eine rechteckige Grundfläche, wobei die Abmessungen rastergenau sein sollen (also Vielfache der Rastergröße). Weiterhin gibt es eine Bar, die als größerer rechteckiger Bereich modelliert werden kann (keine Unterscheidung in Tresen plus Tische plus Stühle o.ä. notwendig).

Die Aufgabe besteht darin, die Spielgeräte und den Bar-Bereich nach bestimmten Randbedingungen anzuordnen. Zu diesen Vorgaben gehört, dass sich die Spielgeräte nicht überlappen oder berühren, dass bestimmte Sicherheitsabstände eingehalten werden (1m zw. den Spielgeräten), dass Entspannungszonen eingerichtet werden und dass Sichtlinien gewährleistet sind (z.B. sollten die Aktivitäten am Kletterberg vom Barbereich aus beobachtet werden können). Die Bar ist bevorzugt direkt am Eingang zu finden. Notausgänge dürfen nicht verstellt werden.

Abstrahieren Sie das gegebene Problem angemessen und geben Sie eine geeignete Modellierung als CSP an. Definieren Sie sich ein paar Spielgeräte und lösen Sie das Problem mit Hilfe von *MAC* und *Min-Conflicts*.

Thema: Modellierung eines Real-World-Problems

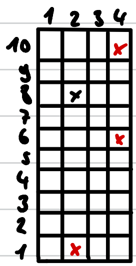
Spalte

Domänen

$D = \{1,2,3,4\} \times \{1,2,3,4,5,6,7,8,9,10\}$

$D_{GKB} = D_{KB} = D_{HB} = D_{BAR} = D$

Zeile



40 Positionen durch 10x10 Raster

40 x 100 10x10

Constraints:

$c_1 =$ GKB ∉ KB ∉ HB ∉ BAR (kein Überlappen)

$c_2 = (2,1) \notin$ GKB, KB, BAR

$c_3 = (4,10) \notin$ GKB, KB, BAR

$c_4 = (4,6) \notin$ GKB, KB, BAR

$c_5 = (x,y) \in BAR \mid y < 2$

$c_6 = (Bar, KB) \mid (x,y) \in D \mid x = x$

$c_7 = (GKB, KB) \mid (x,y) \in D \mid |x_i - x_j| + |y_i - y_j| \geq 2$

$c_8 = (KB, GKB) \mid (x,y) \in D \mid |x_i - x_j| + |y_i - y_j| \geq 2$

$c_9 = (HB, GKB) \mid (x,y) \in D \mid |x_i - x_j| + |y_i - y_j| \geq 2$

$c_{10} = (GKB, HB) \mid (x,y) \in D \mid |x_i - x_j| + |y_i - y_j| \geq 2$

$c_{11} = (HB, KB) \mid (x,y) \in D \mid |x_i - x_j| + |y_i - y_j| \geq 2$

$c_{12} = (KB, HB) \mid (x,y) \in D \mid |x_i - x_j| + |y_i - y_j| \geq 2$

$c_{13} = (x,y) \in BAR, GKB, KB \mid y > 2$

$c_{14} = |BAR| = 1$

$c_{15} = |GKB| = 1$

$c_{16} = |KB| = 1$

$c_{17} = |HB| = 1$

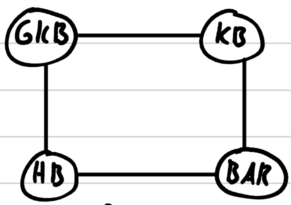
Höhe+1 Breite+1

Position (2,1) = Tür

Position $(\underset{x}{4}, \underset{y}{6})$ = Notausgang 1

Position (4,10) = Notausgang 2

(2,9)



{(2,8)}

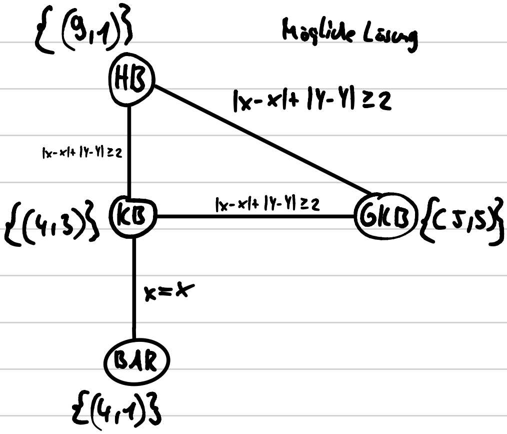
(2,1) (2,2)
(1,1) (1,2)

Kein Berühren ✓

Sicherheitsabstände zwischen Spielgeräten

$|X - X'| + |Y - Y'| \geq 2$

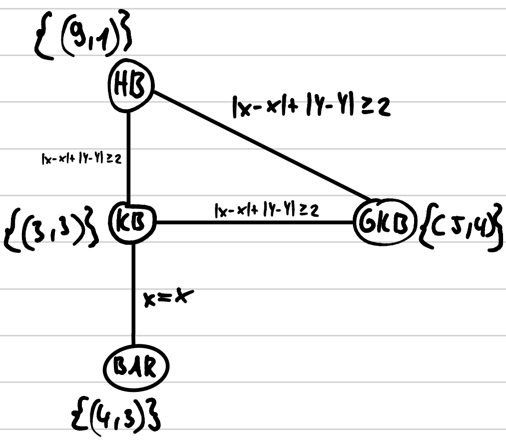
0 + 1



Mögliche Lösung

{(1,1),(1,2),(1,3),(1,4)
(2,1),(2,2),(2,3),(2,4)
(3,1),(3,2),(3,3),(3,4)}

Mac und Min-Conflicts CSP.



Generiere zufälliges CSP:

Suche Knoten mit max. Konflikte und wähle Konfliktvariable:

BAR = {4, 3} 2 Konflikte

KB = {3,2} 1 Konflikt

Variable BAR wird neu belegt

BAR = {4,1} = 1 Konflikt

KB = {3,2} = 1 Konflikt

Suche Knoten mit max. Konflikte und wähle Konfliktvariable:

BAR = {4,1} = 1 Konflikt

KB = {3,2} = 1 Konflikt

Variable BAR wird neu belegt

BAR = {3,1} = 0 Konflikte

KB = {3,2} = 0 Konflikte