

## Aufgabe 05

Da wir das Problem abstrahieren sollen und „Raster“ sich sehr nach Koordinatenkreuz anhört, könnte man das Problem mit Koordinaten modellieren. Dabei haben die Türen 2 (x,y) Koordinaten, die Spielgeräte und die Bar jeweils 4 Koordinaten. Die Ruhezonen definiere ich auch als Rechteck. Das habe ich versucht und dabei versucht, die Constraints mathematisch aufzubauen. Das war überraschend schwer und ich werde sie jetzt quasi mit Pseudo-Mathe beschreiben, da wir das Problem ja nur durchspielen und nicht programmieren müssen.

### Constraints Tür:

Die Tür muss in einer Wand sein:

$C1 = \{\text{beide } x \text{ | oder beide } y \text{ müssen ein Min oder Max sein}\}$

Die Tür darf nicht direkt an einem Rechteck sein:

$C2 = \{\text{die Kante der Tür darf nicht mit Kante eines Rechtecks überschneiden}\}$

### Constraints Rechteck:

Die Rechtecke dürfen sich nicht überschneiden:

$C3 = \{\text{keine Schnittpunkte}\}$

Die Rechtecke müssen einen Sicherheitsabstand von min 1m haben:

$C4 = \{(\text{Kante von äußerer Ecke Rechteck A bis äußere Ecke Rechteck B}) - (\text{Kantenlängen von Rechteck A + B}) \geq 1\}$

$(C4 = \{\text{alle Werte der Koordinaten } +1, \text{ außer, es ist ein min oder max}\})$

Es muss eine Sichlinie herrschen:

$C5 = \{\text{Die Gerade zwischen dem Mittelpunkt Bar und dem Mittelpunkt eines Rechtecks darf nicht unterbrochen sein}\}$

### Es sollen Ruhezonen vorhanden sein

$C6 = \{\text{pro 3 Spielgeräte eine Ruhezone}\}$

Soft- Constraint: Bar in der Nähe der Tür

### MAC: Der Inferenz-Schritt ist der AC-3 Algorithmus

Vorgehen: suche eine Variable aus und setze sie. Verkleinere die Domänen der anderen Variablen.

Für die Handsimulation habe ich 2 Türen(3 lang), 1 Bar(4x4), 3 Spielgeräte (jw 3x4) und 1 Ruhezone(2x2).

Meine Grundfläche ist ein Raster von 10x15 Kästchen (auf einem karierten Papier ☺)

Die Domäne für alle Variablen sind gleich: Alle möglichen Tupel in meinem Koordinatensystem.

Die Domäne ist also:  $\{(x,y) | 0, \dots, 10\} \times \{0, \dots, 15\}$ .

Ich werde eine Liste mit gestrichenen Koordinaten führen, das ist einfacher als die Menge des kartesischen Produktes aufzuschreiben.

Durchlauf 1:

Ich setze als erste Variable die (Eingangstür) Tür:

Constraint: x oder y muss min oder max sein:

Koordinaten: (0,6), (0,9)

Liste gestrichener Koordinaten: { (0,6), (0,9) }

## Durchlauf 2:

platziere die Bar

-> in Nähe von Eingang also Richtung 0

Koordinaten: (0,0), (4,0), (0,4), (4, 4)

Koordinaten Sicherheitsabstand: (5,0), (0,5), (5, 5)

Liste gestrichener Koordinaten:{ (0,6), (0,9), (0,0), (4,0), (0,4), (4, 4), 5,0), (0,5), (5, 5)}

## Durchlauf 3:

platziere Spielgerät 1: (0,6) (0,9) (4,6) (4,9)

-> Sicherheitsabstand eingehalten

-> Sichtlinie ununterbrochen

-> keine Schnittpunkte mit anderen Rechtecken

Koordinaten Sicherheitsabstand: (0,7) (0,10) (5,7) (5,10)

Liste gestrichener Koordinaten:{ (0,6), (0,9), (0,0), (4,0), (0,4), (4, 4), 5,0), (0,5), (5, 5), (0,6) (0,9) (4,6) (4,9), (0,7), (0,10), (5,7), (5,10)}

## Durchlauf 4

platziere Spielgerät 2: (6,6)(6,9)(10,6)(10,9)

-> Sicherheitsabstand eingehalten

-> Sichtlinie ununterbrochen

-> keine Schnittpunkte mit anderen Rechtecken

Koordinaten Sicherheitsabstand (7,7)(7,10)(10,7)(10,10)

Liste gestrichener Koordinaten:{ (0,6), (0,9), (0,0), (4,0), (0,4), (4, 4), 5,0), (0,5), (5, 5), (0,6) (0,9) (4,6) (4,9), (0,7), (0,10), (5,7), (5,10), (6,6)(6,9)(10,6)(10,9), 7,7)(7,10)(10,7)(10,10)}

## Durchlauf 5

platziere Spielgerät 3: (12,5)(12,9)(15,5)(15,9)

-> Sicherheitsabstand eingehalten

-> Sichtlinie ununterbrochen

-> keine Schnittpunkte mit anderen Rechtecken

Koordinaten Sicherheitsabstand: (13, 6)(13,10)(15,6)(15,10)

Liste gestrichener Koordinaten:{ (0,6), (0,9), (0,0), (4,0), (0,4), (4, 4), 5,0), (0,5), (5, 5), (0,6) (0,9) (4,6) (4,9), (0,7), (0,10), (5,7), (5,10), (6,6)(6,9)(10,6)(10,9), (7,7)(7,10)(10,7)(10,10), (12, 5)(12,9)(15,5)(15,9), (13, 6)(13,10)(15,6)(15,10)}

## Durchlauf 6

platziere zweite Tür: (15,0)(15,3)

Liste gestrichener Koordinaten:{ (0,6), (0,9), (0,0), (4,0), (0,4), (4, 4), 5,0), (0,5), (5, 5), (0,6) (0,9) (4,6) (4,9), (0,7), (0,10), (5,7), (5,10), (6,6)(6,9)(10,6)(10,9), 7,7)(7,10)(10,7)(10,10), (12, 5)(12,9)(15,5)(15,9), (13, 6)(13,10)(15,6)(15,10), (15,0)(15,3)}

## Durchlauf 7

platziere Ruhezone: (8,2)(8,4)(10,2)(10,4)

-> Sicherheitsabstand eingehalten

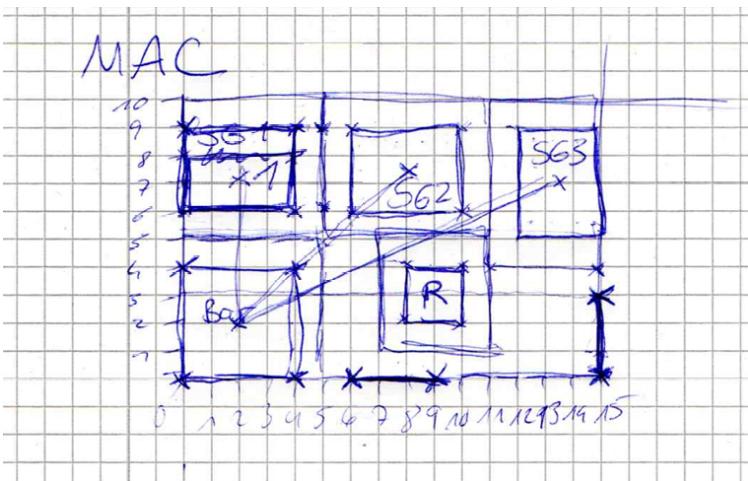
-> Sichtlinie ununterbrochen

-> keine Schnittpunkte mit anderen Rechtecken

Koordinaten Sicherheitsabstand: (9,3)(8,5)(11,3)(11,5)

Damit ist alles platziert.

Lösung:



Min-Conflicts

Vorgehen: zufällige Startposition aller Variablen festlegen

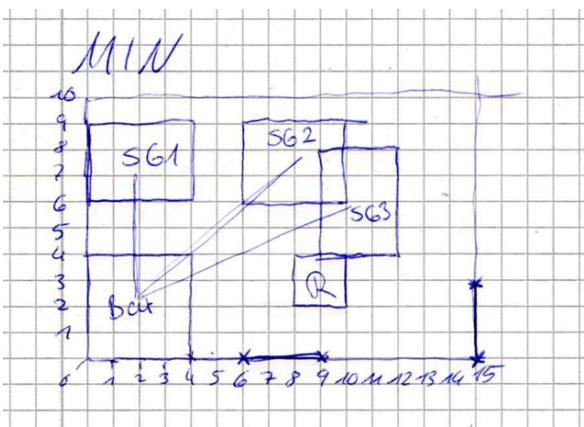
Konflikte zählen

eine Variable mit Konflikten wählen und auf eine Position mit weniger/ ohne Konflikte setzen

Wiederholen bis alles konfliktfrei

( 2 Türen(3 lang), 1 Bar(4x4), 3 Spielgeräte (jw 3x4) und 1 Ruhezone(2x2))

Erste Setzung der Variablen:



Bar: ohne Konflikte

Eingangstür/Notausgang: ohne Konflikte

SG1: ohne Konflikte

SG2: 2 Konflikte, 2 Schnittpunkte mit SG3

SG3: 3 Konflikte, 2 Schnittpunkte mit SG2 und 1 Schnittpunkt mit Ruhezone

Ruhezone: 1 Konflikt, Schnittpunkt mit SG3

SG3 wird neu platziert, da hier die meisten Konflikte auftreten.

Bar: ohne Konflikte

Eingangstür/Notausgang: ohne Konflikte

SG1: ohne Konflikte

SG2: ohne Konflikte

SG3: ohne Konflikte

Ruhezone: ohne Konflikte

Die neu platzierung von SG3 mit den meisten Konflikten hat dazu geführt, dass eine Lösung ohne Konflikte entstanden ist.

