Einstein-Rätsel: CSP-Formulierung

1 CSP.01: Logikrätsel

1.1 Problemstellung

Das berühmte Einstein-Rätsel besteht aus:

- Es gibt 5 Häuser mit verschiedenen Farben
- In jedem Haus wohnt eine Person mit unterschiedlicher Nationalität
- Jede Person trinkt ein anderes Getränk, raucht eine andere Zigarettenmarke und hat ein anderes Haustier

1.2

sectionFormulierung als CSP

1.3 Variablen

Für jedes Haus i (wobei $i \in \{1, 2, 3, 4, 5\}$) haben wir 5 Variablen:

- Farbe_i: die Farbe des Hauses i
- Nationalität_i: die Nationalität der Person in Haus i
- Getränk_i: das Getränk, das in Haus i getrunken wird
- Zigarette_i: die Zigarettenmarke, die in Haus i geraucht wird
- Tier_i: das Haustier in Haus i

Insgesamt: 25 Variablen (5 Häuser $\times$ 5 Attribute)

1.4 Wertebereiche (Domänen)

$$
\begin{aligned}
D(\text{Farbe}_i) &= \{\text{Rot, Grün, Weiß, Gelb, Blau}\} \\
D(\text{Nationalität}_i) &= \{\text{Engländer, Spanier, Ukrainer, Norweger, Japaner}\} \\
D(\text{Getränk}_i) &= \{\text{Kaffee, Tee, Milch, Orangensaft, Wasser}\} \\
D(\text{Zigarette}_i) &= \{\text{Old-Gold, Kools, Chesterfield, Lucky-Strike, Parliaments}\} \\
D(\text{Tier}_i) &= \{\text{Hund, Schnecken, Fuchs, Pferd, Zebra}\}
\end{aligned}
$$

1.5 Constraints

1.5.1 Globale Constraints (All-Different)

Für jedes Attribut müssen alle Werte unterschiedlich sein:

- **C1**: $\text{Farbe}_1, \text{Farbe}_2, \text{Farbe}_3, \text{Farbe}_4, \text{Farbe}_5$ sind alle verschieden
- **C2**: $\text{Nationalität}_1, \ldots, \text{Nationalität}_5$ sind alle verschieden
- **C3**: $\text{Getränk}_1, \ldots, \text{Getränk}_5$ sind alle verschieden
- **C4**: $\text{Zigarette}_1, \ldots, \text{Zigarette}_5$ sind alle verschieden
- **C5**: $\text{Tier}_1, \ldots, \text{Tier}_5$ sind alle verschieden

1.5.2 Unäre Constraints

- **C6**: $\text{Nationalität}_1 = \text{Norweger}$
  (Der Norweger wohnt im ersten Haus)
- **C7**: $\text{Getränk}_3 = \text{Milch}$
  (Im mittleren Haus wird Milch getrunken)

1.5.3 Binäre Constraints (innerhalb eines Hauses)

- **C8**: $\forall i : \text{Nationalität}_i = \text{Engländer} \Rightarrow \text{Farbe}_i = \text{Rot}$
  (Der Engländer wohnt im roten Haus)
- **C9**: $\forall i : \text{Nationalität}_i = \text{Spanier} \Rightarrow \text{Tier}_i = \text{Hund}$
  (Der Spanier hat einen Hund)
- **C10**: $\forall i : \text{Nationalität}_i = \text{Ukrainer} \Rightarrow \text{Getränk}_i = \text{Tee}$
  (Der Ukrainer trinkt Tee)
- **C11**: $\forall i : \text{Farbe}_i = \text{Grün} \Rightarrow \text{Getränk}_i = \text{Kaffee}$
  (Im grünen Haus wird Kaffee getrunken)
- **C12**: $\forall i : \text{Zigarette}_i = \text{Old-Gold} \Rightarrow \text{Tier}_i = \text{Schnecken}$
  (Derjenige, der Old-Gold raucht, hat Schnecken)
- **C13**: $\forall i : \text{Farbe}_i = \text{Gelb} \Rightarrow \text{Zigarette}_i = \text{Kools}$
  (Im gelben Haus werden Kools geraucht)
- **C14**: $\forall i : \text{Zigarette}_i = \text{Lucky-Strike} \Rightarrow \text{Getränk}_i = \text{Orangensaft}$
  (Derjenige, der Lucky-Strike raucht, trinkt Orangensaft)
- **C15**: $\forall i : \text{Nationalität}_i = \text{Japaner} \Rightarrow \text{Zigarette}_i = \text{Parliaments}$
  (Der Japaner raucht Parliaments)

1.5.4 Binäre Constraints (zwischen benachbarten Häusern)

- **C16**: $\exists i \in \{1,2,3,4\}$: $\text{Farbe}_i = \text{Weiß} \wedge \text{Farbe}_{i+1} = \text{Grün}$
  (Das grüne Haus steht direkt rechts vom weißen Haus)
- **C17**: $\forall i$: $\text{Zigarette}_i = \text{Chesterfield} \Rightarrow (\text{Tier}_{i-1} = \text{Fuchs} \vee \text{Tier}_{i+1} = \text{Fuchs})$
  (Der Chesterfield-Raucher wohnt neben dem Mann mit Fuchs)
- **C18**: $\forall i$: $\text{Zigarette}_i = \text{Kools} \Rightarrow (\text{Tier}_{i-1} = \text{Pferd} \vee \text{Tier}_{i+1} = \text{Pferd})$
  (Kools wird im Haus neben dem Haus mit Pferd geraucht)
- **C19**: $\text{Nationalität}_1 = \text{Norweger} \Rightarrow \text{Farbe}_2 = \text{Blau}$
  (Der Norweger wohnt neben dem blauen Haus)

CSP.03: Kantenkonsistenz mit AC-3

Domäne $D = \{0, \dots, 5\}$, Variablen v_1, v_2, v_3, v_4 mit $D_{v_i} = D$. Constraints:

$$\begin{aligned}
c_1 &= ((v_1, v_2), \{(x,y) \in D^2 \mid x + y = 3\}),\\
c_2 &= ((v_2, v_3), \{(x,y) \in D^2 \mid x + y \leq 3\}),\\
c_3 &= ((v_1, v_3), \{(x,y) \in D^2 \mid x \leq y\}),\\
c_4 &= ((v_3, v_4), \{(x,y) \in D^2 \mid x \neq y\}).
\end{aligned}$$

1) Constraint-Graph

Knoten: v_1, v_2, v_3, v_4. Kanten: v_1–v_2 durch c_1, v_2–v_3 durch c_2, v_1–v_3 durch c_3, v_3–v_4 durch c_4.

2) AC-3 Handsimulation

Startdomänen:

$$D_1 = D_2 = D_3 = D_4 = \{0, 1, 2, 3, 4, 5\}.$$

Initiale Queue gerichteter Bögen:

$$(v_1 \to v_2), (v_2 \to v_1), (v_2 \to v_3), (v_3 \to v_2), (v_1 \to v_3), (v_3 \to v_1), (v_3 \to v_4), (v_4 \to v_3).$$

Im Folgenden steht D_i für D_{v_i}. Bei Reduktion werden die geänderten Domänen gezeigt.

1. **Bogen** $\mathrm{v}_1 \to \mathrm{v}_2$ mit $x + y = 3$. **Reduktion** entferne $\{4, 5\}$ aus D_1.
   Domänen $D_1 = \{0, 1, 2, 3\}$, $D_2 = D_3 = D_4 = \{0, \dots, 5\}$.
   Queue füge $(v_3 \to v_1)$ hinzu.
2. **Bogen** $\mathrm{v}_2 \to \mathrm{v}_1$ mit $x + y = 3$. **Reduktion** entferne $\{4, 5\}$ aus D_2.
   Domänen $D_1 = \{0, 1, 2, 3\}$, $D_2 = \{0, 1, 2, 3\}$, $D_3 = D_4 = \{0, \dots, 5\}$.
   Queue füge $(v_3 \to v_2)$ hinzu.
3. **Bogen** $\mathrm{v}_2 \to \mathrm{v}_3$ mit $x + y \leq 3$. **Reduktion** keine.
   Domänen unverändert. **Queue** kein Update.

4. **Bogen** $\mathrm{v}_3 \to \mathrm{v}_2$ mit $x + y \leq 3$. **Reduktion** entferne $\{4, 5\}$ aus D_3.
   Domänen $D_1 = \{0, 1, 2, 3\}$, $D_2 = \{0, 1, 2, 3\}$, $D_3 = \{0, 1, 2, 3\}$, $D_4 = \{0, \ldots, 5\}$.
   Queue füge $(v_1 \to v_3)$ und $(v_4 \to v_3)$ hinzu.

5. **Bogen** $\mathrm{v}_1 \to \mathrm{v}_3$ mit $x \leq y$. **Reduktion** keine.
   Domänen unverändert. **Queue** kein Update.

6. **Bogen** $\mathrm{v}_3 \to \mathrm{v}_1$ mit $x \leq y$. **Reduktion** keine.
   Domänen unverändert. **Queue** kein Update.

7. **Bogen** $\mathrm{v}_3 \to \mathrm{v}_4$ mit $x \neq y$. **Reduktion** keine.
   Domänen unverändert. **Queue** kein Update.

8. **Bogen** $\mathrm{v}_4 \to \mathrm{v}_3$ mit $x \neq y$. **Reduktion** keine.
   Domänen unverändert. **Queue** kein Update.

Endzustand bogenkonsistent:

$$D(v_1) = \{0, 1, 2, 3\}, \quad D(v_2) = \{0, 1, 2, 3\},$$
$$D(v_3) = \{0, 1, 2, 3\}, \quad D(v_4) = \{0, 1, 2, 3, 4, 5\}.$$

CSP.04: Kantenkonsistenz nach Zuweisung $\alpha = \{v_1 \to 2\}$

Gegeben sei das CSP aus der vorigen Aufgabe mit

$$D = \{0, 1, 2, 3, 4, 5\}, \quad \alpha = \{v_1 \mapsto 2\}.$$

Ausgangszustand der Wertebereiche

Gemäß Hinweis gilt:

$$\begin{aligned} D(v_1) &= \{2\}, \\ D(v_2) &= \{0, 1, 2, 3, 4, 5\}, \\ D(v_3) &= \{0, 1, 2, 3, 4, 5\}, \\ D(v_4) &= \{0, 1, 2, 3, 4, 5\}. \end{aligned}$$

Herstellen der Kantenkonsistenz

1) Constraint c_1: (v_1, v_2) mit $x + y = 3$. Da $v_1 = 2$, muss $v_2 = 1$ gelten.

$$D(v_2) = \{1\}.$$

2) Constraint c_3: (v_1, v_3) mit $x \leq y$. Mit $v_1 = 2$ folgt $v_3 \in \{2, 3, 4, 5\}$.

$$D(v_3) = \{2, 3, 4, 5\}.$$

3) Constraint c_2: (v_2, v_3) mit $x + y \leq 3$. Da $v_2 = 1$, gilt $1 + y \leq 3 \Rightarrow y \leq 2$. Schnitt mit $\{2, 3, 4, 5\}$ liefert $D(v_3) = \{2\}$.

4) Constraint c_4: (v_3, v_4) mit $x \neq y$. Mit $v_3 = 2$ folgt $v_4 \neq 2$.

$$D(v_4) = \{0, 1, 3, 4, 5\}.$$

Endzustand nach Kantenkonsistenz in α

$$\begin{aligned} D(v_1) &= \{2\}, \\ D(v_2) &= \{1\}, \\ D(v_3) &= \{2\}, \\ D(v_4) &= \{0,1,3,4,5\}. \end{aligned}$$

Forward-Checking in α und Vergleich mit Kantenkonsistenz

Ausgangslage

Gegeben ist die partielle Belegung

$$\alpha = \{v_1 \to 2\}, \quad D = \{0,1,2,3,4,5\}.$$

Zu Beginn gelten also:

$$D(v_1) = \{2\}, \quad D(v_2) = D(v_3) = D(v_4) = \{0,1,2,3,4,5\}.$$

Forward-Checking ausgehend von $v_1 = 2$

Beim Forward-Checking werden nur die Domänen jener noch unbelegten Variablen reduziert, die direkt mit der gerade belegten Variablen durch ein Constraint verbunden sind.

1. **Constraint** $c_1 : (v_1, v_2)$ **mit** $x + y = 3$
   Für $v_1 = 2$ gilt $2 + y = 3 \Rightarrow y = 1$.
   Somit $D(v_2) = \{1\}$.

2. **Constraint** $c_3 : (v_1, v_3)$ **mit** $x \leq y$
   Für $v_1 = 2$ ergibt sich $y \geq 2 \Rightarrow D(v_3) = \{2,3,4,5\}$.

3. **Constraint** $c_4 : (v_3, v_4)$
   v_1 ist hier nicht beteiligt, daher keine Veränderung.
   $D(v_4) = \{0,1,2,3,4,5\}$.

4. **Constraint** $c_2 : (v_2, v_3)$
   Da v_2 noch nicht endgültig belegt ist (Forward-Checking prüft nur Nachbarn der zuletzt belegten Variablen), erfolgt hier keine zusätzliche Einschränkung.

Ergebnis des Forward-Checkings

$$\begin{aligned} D(v_1) &= \{2\}, \\ D(v_2) &= \{1\}, \\ D(v_3) &= \{2,3,4,5\}, \\ D(v_4) &= \{0,1,2,3,4,5\}. \end{aligned}$$

Vergleich mit der Kantenkonsistenz in α

Beim vorherigen Schritt (Kantenkonsistenz in α) ergab sich:

$$D(v_1) = \{2\}, \quad D(v_2) = \{1\}, \quad D(v_3) = \{2\}, \quad D(v_4) = \{0,1,3,4,5\}.$$

Unterschiede:

- Forward-Checking reduziert die Domänen nur auf Basis der direkt betroffenen Constraints von v_1.
- Die Kantenkonsistenz propagiert weiter: aus $v_2 = 1$ folgt über c_2, dass $v_3 = \{2\}$; daraus wiederum über c_4, dass $v_4 \neq 2$.
- Somit ist das Ergebnis der Kantenkonsistenz *stärker eingeschränkt* als das des Forward-Checkings.

CSP.05: Problemabstraktion

Das Indoor-Spielplatz-Planungsproblem lässt sich als **Constraint Satisfaction Problem (CSP)** mit räumlichen Platzierungsconstraints modellieren.

CSP-Formalisierung

Variablen

Für jedes Objekt $i \in \{1, \ldots, n\}$ (Spielgeräte + Bar) definieren wir:

- $x_i, y_i \in \mathbb{N}_0$: Koordinaten der linken unteren Ecke in Rastereinheiten
- $o_i \in \{0, 90\}$: Orientierung (optional, falls Rotation erlaubt)

Für die Basisformulierung: $V = \{x_1, y_1, \ldots, x_n, y_n\}$

Domänen

Gegeben:

- Spielplatzgröße: $L \times B$ (z. B. 400×1000 Rastereinheiten bei 10 cm-Raster)
- Objektabmessungen: $w_i \times h_i$ für jedes Objekt i

Domänen:

$$D_{x_i} = \{0, 1, \ldots, L - w_i\} \quad (1)$$
$$D_{y_i} = \{0, 1, \ldots, B - h_i\} \quad (2)$$

Constraints

Keine Überlappung (Hard Constraint)

Für alle Objektpaare $i \neq j$:

$$\text{noOverlap}(i, j): \quad x_i + w_i \leq x_j \lor x_j + w_j \leq x_i \lor y_i + h_i \leq y_j \lor y_j + h_j \leq y_i \quad (3)$$

Sicherheitsabstand (Hard Constraint)

Mindestabstand $d_{\text{safety}} = 10$ Rastereinheiten (1 m) zwischen Spielgeräten:
Für alle Spielgeräte-Paare i, j (Bar ausgenommen):

$$\begin{aligned} \text{safetyDistance}(i,j): \quad & x_i + w_i + d_{\text{safety}} \leq x_j \vee x_j + w_j + d_{\text{safety}} \leq x_i \vee \\ & y_i + h_i + d_{\text{safety}} \leq y_j \vee y_j + h_j + d_{\text{safety}} \leq y_i \end{aligned} \tag{4}$$

Notausgang-Freiheit (Hard Constraint)

Für jeden Notausgang an Position (e_x, e_y) mit Breite e_w und Zone e_d (Freihaltetiefe):

$$\text{emergencyExit}(i,e): \quad \neg\text{overlaps}(\text{rect}_i, \text{exitZone}_e) \tag{5}$$

Die Exitzone ist definiert als Rechteck vor dem Ausgang.

Bar am Eingang (Soft/Hard Constraint)

Sei b der Index der Bar und (m_x, m_y) die Position des Haupteingangs:

$$\text{barNearEntrance}(b): \quad \text{distance}((x_b, y_b), (m_x, m_y)) \leq d_{\max} \tag{6}$$

Oder als Soft Constraint mit Kostenfunktion:

$$\text{minimize distance}((x_b, y_b), (m_x, m_y)) \tag{7}$$

Sichtlinien (Soft Constraint)

Für spezifische Objekt-Paare (z. B. Bar b und Kletterberg k):

$$\text{lineOfSight}(b,k): \quad \neg\exists j \neq b, k : \text{blocksView}(j, b, k) \tag{8}$$

Vereinfachte Variante: Maximiere Anzahl der Objekte, die von der Bar aus „sichtbar“ sind (z. B. keine anderen Objekte zwischen den Mittelpunkten).

Entspannungszonen (Soft Constraint)

Definiere Zonen $Z = \{z_1, \ldots, z_m\}$ als rechteckige Bereiche ohne Spielgeräte:

$$\text{relaxZone}(z): \quad \forall i : \neg\text{overlaps}(\text{rect}_i, z) \vee i = \text{bar} \tag{9}$$

Oder als Optimierungskriterium: Maximiere freie Fläche in bestimmten Bereichen.