Klausur Planen und Entscheiden SS 2009 Lösung

Jan Strohbeck Michael Kaps Tobias Häußer
Dominik Bergen Kowsikan Sathiyamoorthy

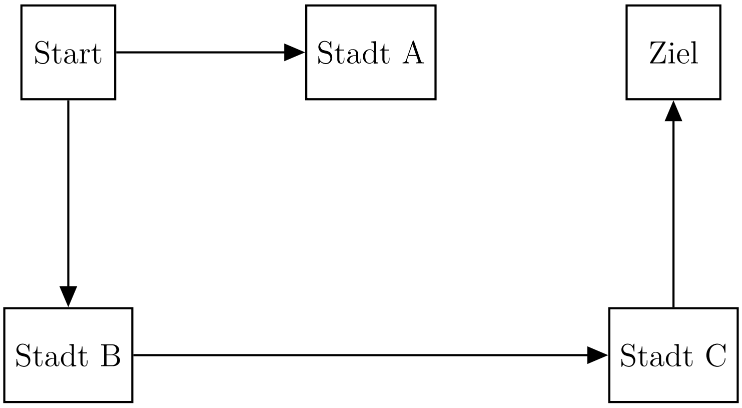
2. Juli 2016, Aalen

Inhaltsverzeichnis

1 Aufgabe

1.1 Erreicht der Agent immer sein Ziel?

Nein, da er so unter Umständen in eine Sackgasse gelangen könnte und nicht mehr herausfinden würde (s. untenstehende Skizze).



So würde der Agent immer zwischen den Städten „Start“ und „Stadt A“ pendeln und nie sein Ziel erreichen.

1.2 Ordnen Sie dem Agenten einen Agententyp zu

→ Reflexagent

1.3 Arbeitsumgebung

Performance Der Agent muss auf dem kürzesten Weg das Ziel erreichen
→ z.B. zurückgelegte Strecke wird negativ bewertet, höchster Wert (kleinste Strecke) ist der höchste Wert

Environment Städte, Straßen zwischen Städten

Actuators Möglichkeit, sich in eine Stadt zu bewegen, die mit der aktuellen Stadt über eine Straße verbunden ist

Sensors
- spezieller Kompass, der immer auf das Ziel zeigt
- erkennt von der aktuellen Stadt ausgehende Straßen

1.4 Eigenschaften der Arbeitsumgebung

teilweise beobachtbar Es ist nicht sofort bekannt, welche Städte und Straßen existieren (nur die Start-Stadt und davon ausgehende Straßen).

deterministisch Kein Zufall, Umgebung ändert sich nicht stategisch bzw. gar nicht

eqisodisch Für den Agenten hängen seine Entscheidungen nicht von den vorherigen Entscheidungen ab.

statisch Es ändern sich z.B. keine Verbindungen zwischen Städten zur Laufzeit, alles ist statisch.

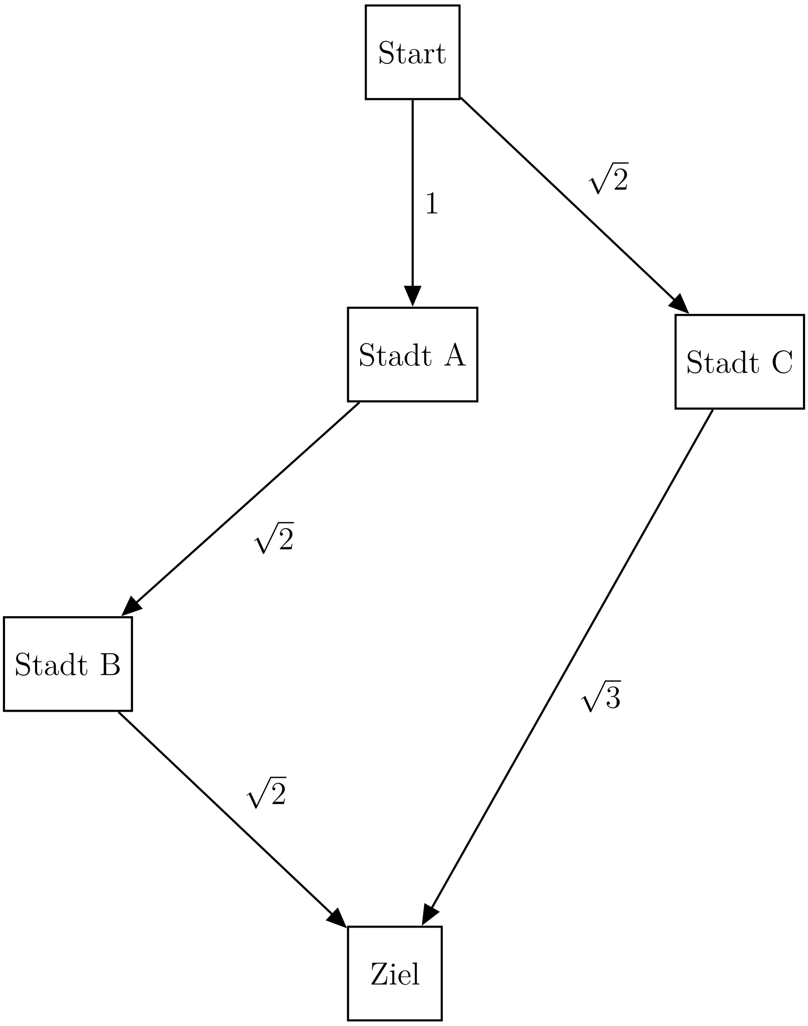
diskret Der Agent kann sich nur in diskreten Zuständen befinden (in Städten), nicht etwa zwischen Städten. Die Zeitintervalle zwischen Aktionen sind auch diskret.

Einzelagent Es existieren keine weiteren Agenten.

1.5 Optimale Landkarte

Agent bewegt sich über Stadt A direkt zum Ziel. Dies ist die optimale Lösung.

1.6 Nicht optimale Landkarte



Weg des Agenten (Start → Stadt A → Stadt B → Ziel):

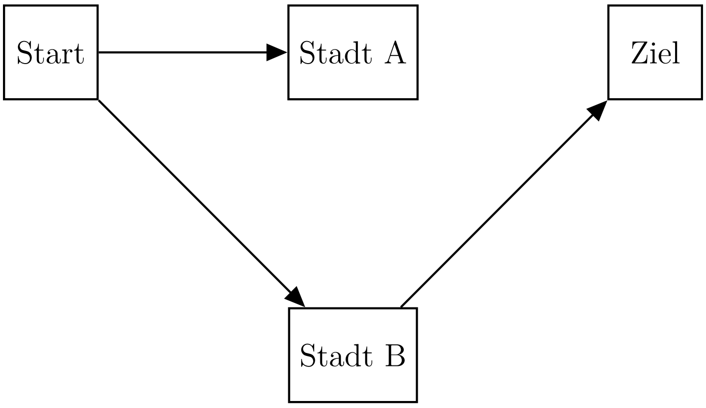
$$w_1 = 1 + 2 \cdot \sqrt{2} \approx 3.828427125$$

Optimaler Weg (Start → Stadt C):

$$w_o = \sqrt{2} + \sqrt{3} \approx 3.14626437$$

Die Karte aus Kap. 1.7 passt hier auch, da der Agent nie ins Ziel kommt, was ebenfalls nicht optimal ist.

1.7 vs. Zufalls-Agent



Der Agent würde hier nie zum Ziel kommen (s. Kap. 1.1), ein zufällig agierender Agent hätte hier zumindest eine Wahrscheinlichkeit ungleich Null, irgendwann zum Ziel zu kommen. Dies ist jedoch nicht garantiert, da auch hier bei ungünstigen Entscheidungen der Agent in eine Endlosschleife gelangen könnte, ohne je ins Ziel zu kommen.

2 Aufgabe

2.1 Zulässigkeit

$h_1(n)$**: Luftlinienentfernung des zu n gehörenden Knotens zum Startknoten** Zulässig, da von jedem Knoten aus für einen Rundweg noch mindestens die Distanz zum Startknoten zurückgelegt werden muss. Die Luftlinienentfernung zum Startknoten ist dabei immer kleiner oder gleich der tatsächlich benötigten Distanz zum Startknoten (evtl. über mehrere Zwischenknoten).

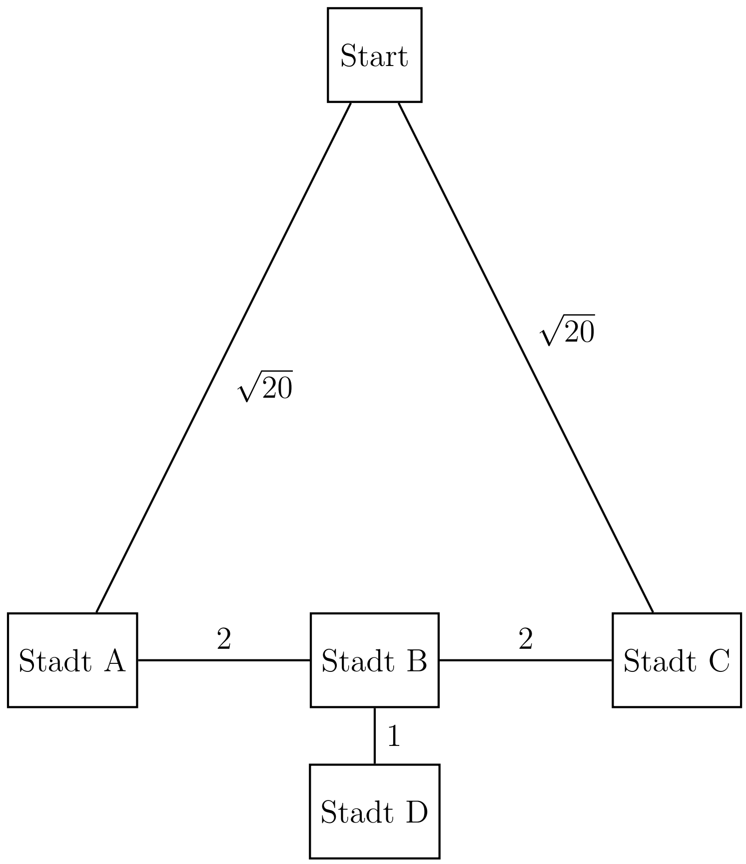
$h_2(n)$**: kürzester Pfad des zu n gehörenden Knotens zum Startknoten** Zulässig, da von jedem Knoten aus für einen Rundweg noch mindestens die Distanz zum Startknoten zurückgelegt werden muss. Der kürzeste Pfad von diesem Knoten entspricht genau dieser Distanz.

$h_3(n)$**: Summe der Entfernungen der im Zustand n noch nicht besuchten Knoten zu dem Nachbarknoten mit der geringsten Entfernung**

Zulässig, da die Strecke zu jedem noch unbesuchten Knoten noch zurückgelegt werden muss und diese im günstigsten Fall nur jene zum Nachbarknoten ist.

$h_4(n)$**: Summe der Luftlinienentfernungen aller im Zustand n noch nicht besuchten Knoten zum Startknoten**

Nicht zulässig, Gegenbeispiel (noch keine Stadt besucht):



$$h_4(\text{Stadt A}) = 4 + 5 + \sqrt{20}$$
$$k(\text{Stadt A}) = 2 + 1 + 1 + 2 + \sqrt{20}$$

2.2 Dominanz

- $h_1(n) \leq h_2(n)$ weil Luftlinienentfernung zu einem Knoten nie größer als Pfad zu einem Knoten (Dreiecksungleichung)
- $h_3(n)$ kann nicht eingeordnet werden:
  - $h_3(n) < h_1(n)$ wenn n und die anderen nicht besuchten Knoten dicht zusammen liegen, aber weit vom Startknoten entfernt sind.
  - $h_3(n) > h_2(n)$ wenn n nahe am Startknoten, die anderen nicht besuchten Knoten aber weit von einander entfernt sind.
- $h_4(n)$ ist nicht zulässig $\rightarrow$ Voraussetzung für Dominanz nicht erfüllt.

3 Aufgabe

3.1 Variablen, Wertebereiche

- Variablen:
  - $\text{Feld}_{i,j} \rightarrow$ Zahl im Feld in Zeile i und Spalte j
  - $i, j \in \{0, 1, 2, 3\}$
  - $\text{Feld}_{0,0}$ befindet sich in der oberen linken Ecke.
- (Basis-)Wertebereiche:
  - $\text{Feld}_{i,j} \in \{1, 2, 3, 4\}, \quad i, j \in \{0, 1, 2, 3\}$
- Constraints (nicht verlangt):
  - $\text{alldiff}(\text{Feld}_{0,j}, \text{Feld}_{1,j}, \text{Feld}_{2,j}, \text{Feld}_{3,j}), \quad j \in \{0, 1, 2, 3\}$
  - $\text{alldiff}(\text{Feld}_{i,0}, \text{Feld}_{i,1}, \text{Feld}_{i,2}, \text{Feld}_{i,3}), \quad j \in \{0, 1, 2, 3\}$
  - $\text{alldiff}(\text{Feld}_{m,n}, \text{Feld}_{m,n+1}, \text{Feld}_{m+1,n}, \text{Feld}_{m+1,n+1}), \quad m, n \in \{0, 2\}$

3.2 aktuelle Wertebereiche der Variablen

- $\text{Feld}_{0,0} \in \{2, 4\}$
- $\text{Feld}_{0,1} \in \{2, 4\}$
- $\text{Feld}_{0,2} \in \{1, 4\}$
- $\text{Feld}_{0,3} \in \{3\}$ (bereits zugewiesen)
- $\text{Feld}_{1,0} \in \{2, 3, 4\}$
- $\text{Feld}_{1,1} \in \{1\}$ (bereits zugewiesen)
- $\text{Feld}_{1,2} \in \{4\}$
- $\text{Feld}_{1,3} \in \{2\}$
- $\text{Feld}_{2,0} \in \{1, 3\}$
- $\text{Feld}_{2,1} \in \{3\}$

- $\text{Feld}_{2,2} \in \{2\}$ (bereits zugewiesen)
- $\text{Feld}_{2,3} \in \{4\}$ (bereits zugewiesen)
- $\text{Feld}_{3,0} \in \{1, 2, 3, 4\}$
- $\text{Feld}_{3,1} \in \{2, 3, 4\}$
- $\text{Feld}_{3,2} \in \{1, 3\}$
- $\text{Feld}_{3,3} \in \{1\}$

3.3 Lösen des Sudokus

$\rightarrow$ Anwenden der Minimum-Remaining-Values-Heuristik (MRV):
1. Iteration:

- Wählen der noch nicht zugewiesenen Variable mit dem kleinsten Wertebereich, z.B. $\text{Feld}_{3,3} \in \{1\}$.
- Probieren des ersten Wertes (hier der einzige Wert): $\text{Feld}_{3,3} = 1$
- Forward-Checking ändert Wertebereiche:
  - $\text{Feld}_{3,2} \in \{3\}$
  - $\text{Feld}_{3,0} \in \{2, 3, 4\}$

2. Iteration:

- Wählen der noch nicht zugewiesenen Variable mit dem kleinsten Wertebereich, z.B. $\text{Feld}_{3,2} \in \{3\}$.
- Probieren des ersten Wertes (hier der einzige Wert): $\text{Feld}_{3,2} = 3$
- Forward-Checking ändert Wertebereiche:
  - $\text{Feld}_{3,0} \in \{2, 4\}$
  - $\text{Feld}_{3,1} \in \{2, 4\}$

3. Iteration:

- Wählen der noch nicht zugewiesenen Variable mit dem kleinsten Wertebereich, z.B. $\text{Feld}_{1,2} \in \{4\}$.
- Probieren des ersten Wertes (hier der einzige Wert): $\text{Feld}_{1,2} = 4$
- Forward-Checking ändert Wertebereiche:
  - $\text{Feld}_{0,2} \in \{1\}$

   - $\text{Feld}_{1,0} \in \{2, 3\}$

4. Iteration:

   - Wählen der noch nicht zugewiesenen Variable mit dem kleinsten Wertebereich, z.B. $\text{Feld}_{0,2} \in \{1\}$
   - Probieren des ersten Wertes (hier der einzige Wert): $\text{Feld}_{0,2} = 1$

4 Aufgabe

4.1 Beschreibung des Problems

→ atomare Sätze sind entweder

- z.B. `Antilope_rot` ist wahr, wenn die Antilope die rote Karte gezogen hat und `Antilope_blau` ist wahr, wenn die Antilope die blaue Karte gezogen hat oder
- leichter zu vereinfachen:

  Es existiert nur z.B. `Antilope_rot`, denn `Antilope_blau` = ¬`Antilope_rot`

4.2 Sätze für Giraffe und Elefant

- Antilope sagt: Papagei rot, Hyäne blau

  $(\texttt{Antilope\_rot} \Rightarrow ((\texttt{Papagei\_rot} \land \texttt{Hyäne\_blau}) \lor (\neg\texttt{Papagei\_rot} \land \neg\texttt{Hyäne\_blau})))\land$
  $(\texttt{Antilope\_blau} \Rightarrow ((\neg\texttt{Papagei\_rot} \land \texttt{Hyäne\_blau}) \lor (\texttt{Papagei\_rot} \land \neg\texttt{Hyäne\_blau})))$

  Mit $\texttt{*\_blau} \Leftrightarrow \neg\texttt{*\_rot}$:

  $(\texttt{Antilope\_rot} \Rightarrow ((\texttt{Papagei\_rot} \land \neg\texttt{Hyäne\_rot}) \lor (\neg\texttt{Papagei\_rot} \land \texttt{Hyäne\_rot})))\land$
  $(\neg\texttt{Antilope\_rot} \Rightarrow ((\neg\texttt{Papagei\_rot} \land \neg\texttt{Hyäne\_rot}) \lor (\texttt{Papagei\_rot} \land \texttt{Hyäne\_rot})))$

- Elefant sagt: Papagei blau, Schlange rot

  $(\texttt{Elefant\_rot} \Rightarrow ((\texttt{Papagei\_blau} \land \texttt{Schlange\_rot}) \lor (\neg\texttt{Papagei\_blau} \land \neg\texttt{Schlange\_rot})))\land$
  $(\texttt{Elefant\_blau} \Rightarrow ((\neg\texttt{Papagei\_blau} \land \texttt{Schlange\_rot}) \lor (\texttt{Papagei\_blau} \land \neg\texttt{Schlange\_rot})))$

  Mit $\texttt{*\_blau} \Leftrightarrow \neg\texttt{*\_rot}$:

  $(\texttt{Elefant\_rot} \Rightarrow ((\neg\texttt{Papagei\_rot} \land \texttt{Schlange\_rot}) \lor (\texttt{Papagei\_rot} \land \neg\texttt{Schlange\_rot})))\land$
  $(\neg\texttt{Elefant\_rot} \Rightarrow ((\texttt{Papagei\_rot} \land \texttt{Schlange\_rot}) \lor (\neg\texttt{Papagei\_rot} \land \neg\texttt{Schlange\_rot})))$

4.3 Resolution

Können Sie die Fragestellung an das Faultier durch einmalige Anwendung der Resolution lösen? → Nein, da mit dem Algorithmus immer nur versucht werden kann, ein einzelnes Literal oder dessen Negation aus den vorhandenen Sätzen abzuleiten.

Zur Anwendung des Resolutionsalgorithmus wird die Wissensbasis zunächst in die Konjunktive Normalform gebracht (hier nicht explizit gefordert):

$$
\begin{aligned}
\mathrm{KB} = {}&(\texttt{Antilope\_rot} \Rightarrow ((\texttt{Papagei\_rot} \land \lnot\texttt{Hyäne\_rot}) \lor (\lnot\texttt{Papagei\_rot} \land \texttt{Hyäne\_rot})))\land\\
&(\lnot\texttt{Antilope\_rot} \Rightarrow ((\lnot\texttt{Papagei\_rot} \land \lnot\texttt{Hyäne\_rot}) \lor (\texttt{Papagei\_rot} \land \texttt{Hyäne\_rot})))\land\\
&(\texttt{Elefant\_rot} \Rightarrow ((\lnot\texttt{Papagei\_rot} \land \texttt{Schlange\_rot}) \lor (\texttt{Papagei\_rot} \land \lnot\texttt{Schlange\_rot})))\land\\
&(\lnot\texttt{Elefant\_rot} \Rightarrow ((\texttt{Papagei\_rot} \land \texttt{Schlange\_rot}) \lor (\lnot\texttt{Papagei\_rot} \land \lnot\texttt{Schlange\_rot})))\\
= {}&(\lnot\texttt{Antilope\_rot} \lor ((\texttt{Papagei\_rot} \land \lnot\texttt{Hyäne\_rot}) \lor (\lnot\texttt{Papagei\_rot} \land \texttt{Hyäne\_rot})))\land\\
&(\texttt{Antilope\_rot} \lor ((\lnot\texttt{Papagei\_rot} \land \lnot\texttt{Hyäne\_rot}) \lor (\texttt{Papagei\_rot} \land \texttt{Hyäne\_rot})))\land\\
&(\lnot\texttt{Elefant\_rot} \lor ((\lnot\texttt{Papagei\_rot} \land \texttt{Schlange\_rot}) \lor (\texttt{Papagei\_rot} \land \lnot\texttt{Schlange\_rot})))\land\\
&(\texttt{Elefant\_rot} \lor ((\texttt{Papagei\_rot} \land \texttt{Schlange\_rot}) \lor (\lnot\texttt{Papagei\_rot} \land \lnot\texttt{Schlange\_rot})))\\
= {}&(\lnot\texttt{Antilope\_rot} \lor (\\
&\quad((\texttt{Papagei\_rot} \land \lnot\texttt{Hyäne\_rot}) \lor \lnot\texttt{Papagei\_rot})\land\\
&\quad((\texttt{Papagei\_rot} \land \lnot\texttt{Hyäne\_rot}) \lor \texttt{Hyäne\_rot})\\
&\quad))\land\\
&(\texttt{Antilope\_rot} \lor (\\
&\quad((\lnot\texttt{Papagei\_rot} \land \lnot\texttt{Hyäne\_rot}) \lor \texttt{Papagei\_rot})\land\\
&\quad((\lnot\texttt{Papagei\_rot} \land \lnot\texttt{Hyäne\_rot}) \lor \texttt{Hyäne\_rot})\\
&\quad))\land\\
&(\lnot\texttt{Elefant\_rot} \lor (\\
&\quad((\lnot\texttt{Papagei\_rot} \land \texttt{Schlange\_rot}) \lor \texttt{Papagei\_rot})\land\\
&\quad((\lnot\texttt{Papagei\_rot} \land \texttt{Schlange\_rot}) \lor \lnot\texttt{Schlange\_rot})\\
&\quad))\land\\
&(\texttt{Elefant\_rot} \lor (\\
&\quad((\texttt{Papagei\_rot} \land \texttt{Schlange\_rot}) \lor \lnot\texttt{Papagei\_rot})\land\\
&\quad((\texttt{Papagei\_rot} \land \texttt{Schlange\_rot}) \lor \lnot\texttt{Schlange\_rot})\\
&\quad))\\
= {}&(\lnot\texttt{Antilope\_rot} \lor (\\
&\quad((\underbrace{\texttt{Papagei\_rot} \lor \lnot\texttt{Papagei\_rot}}_{\text{True}}) \land (\lnot\texttt{Hyäne\_rot} \lor \lnot\texttt{Papagei\_rot}))\land\\
&\quad((\texttt{Papagei\_rot} \lor \texttt{Hyäne\_rot}) \land (\underbrace{\lnot\texttt{Hyäne\_rot} \lor \texttt{Hyäne\_rot}}_{\text{True}}))\land\\
&\quad)\land\\
&(\texttt{Antilope\_rot} \lor (\\
&\quad((\lnot\texttt{Papagei\_rot} \lor \texttt{Papagei\_rot}) \land (\lnot\texttt{Hyäne\_rot}) \lor \texttt{Papagei\_rot})\land\\
&\quad((\lnot\texttt{Papagei\_rot} \lor \texttt{Hyäne\_rot}) \land (\lnot\texttt{Hyäne\_rot} \lor \texttt{Hyäne\_rot}))\\
&\quad)\land\\
&(\lnot\texttt{Elefant\_rot} \lor (\\
&\quad((\lnot\texttt{Papagei\_rot} \lor \texttt{Papagei\_rot}) \land (\texttt{Schlange\_rot}) \lor \texttt{Papagei\_rot})\land\\
&\quad((\lnot\texttt{Papagei\_rot} \lor \lnot\texttt{Schlange\_rot}) \land (\texttt{Schlange\_rot} \lor \lnot\texttt{Schlange\_rot}))\\
&\quad)\land\\
&(\texttt{Elefant\_rot} \lor (\\
&\quad((\texttt{Papagei\_rot} \lor \lnot\texttt{Papagei\_rot}) \land (\texttt{Schlange\_rot} \lor \lnot\texttt{Papagei\_rot}))\land\\
&\quad((\texttt{Papagei\_rot} \lor \lnot\texttt{Schlange\_rot}) \land (\texttt{Schlange\_rot} \lor \lnot\texttt{Schlange\_rot}))\\
&\quad)
\end{aligned}
$$

$$
\begin{aligned}
&= (\neg\texttt{Antilope\_rot} \lor ((\neg\texttt{Hyäne\_rot} \lor \neg\texttt{Papagei\_rot}) \land (\texttt{Papagei\_rot} \lor \texttt{Hyäne\_rot})))\land \\
&\quad (\texttt{Antilope\_rot} \lor ((\neg\texttt{Hyäne\_rot} \lor \texttt{Papagei\_rot}) \land (\neg\texttt{Papagei\_rot} \lor \texttt{Hyäne\_rot})))\land \\
&\quad (\neg\texttt{Elefant\_rot} \lor ((\texttt{Schlange\_rot} \lor \texttt{Papagei\_rot}) \land (\neg\texttt{Papagei\_rot} \lor \neg\texttt{Schlange\_rot})))\land \\
&\quad (\texttt{Elefant\_rot} \lor ((\texttt{Schlange\_rot} \lor \neg\texttt{Papagei\_rot}) \land (\texttt{Papagei\_rot} \lor \neg\texttt{Schlange\_rot}))) \\
&= (\neg\texttt{Antilope\_rot} \lor \neg\texttt{Hyäne\_rot} \lor \neg\texttt{Papagei\_rot})\land && \text{(A)} \\
&\quad (\neg\texttt{Antilope\_rot} \lor \texttt{Papagei\_rot} \lor \texttt{Hyäne\_rot})\land && \text{(B)} \\
&\quad (\texttt{Antilope\_rot} \lor \neg\texttt{Hyäne\_rot} \lor \texttt{Papagei\_rot})\land && \text{(C)} \\
&\quad (\texttt{Antilope\_rot} \lor \neg\texttt{Papagei\_rot} \lor \texttt{Hyäne\_rot})\land && \text{(D)} \\
&\quad (\neg\texttt{Elefant\_rot} \lor \texttt{Schlange\_rot} \lor \texttt{Papagei\_rot})\land && \text{(E)} \\
&\quad (\neg\texttt{Elefant\_rot} \lor \neg\texttt{Papagei\_rot} \lor \neg\texttt{Schlange\_rot})\land && \text{(F)} \\
&\quad (\texttt{Elefant\_rot} \lor \texttt{Schlange\_rot} \lor \neg\texttt{Papagei\_rot})\land && \text{(G)} \\
&\quad (\texttt{Elefant\_rot} \lor \texttt{Papagei\_rot} \lor \neg\texttt{Schlange\_rot}) && \text{(H)}
\end{aligned}
$$

Resolutionsalgorithmus: Um zu zeigen, dass `Elefant_rot` gilt, nehmen wir ¬`Elefant_rot` (I) an. Dann leiten wir damit neue Sätze ab (hier nicht explizit gefordert, nur zur Demonstration des Prinzips):

- Aus Kombination von I und E folgt: `Schlange_rot` ∨ `Papagei_rot` (J)
- Aus Kombination von J und G folgt: `Elefant_rot` ∨ `Schlange_rot` (K)
- Aus Kombination von K und H folgt: `Elefant_rot` ∨ `Papagei_rot` (L)
- Aus Kombination von L und I folgt: `Papagei_rot` (M)
- Aus Kombination von M und D folgt: `Antilope_rot` ∨ `Hyäne_rot` (N)

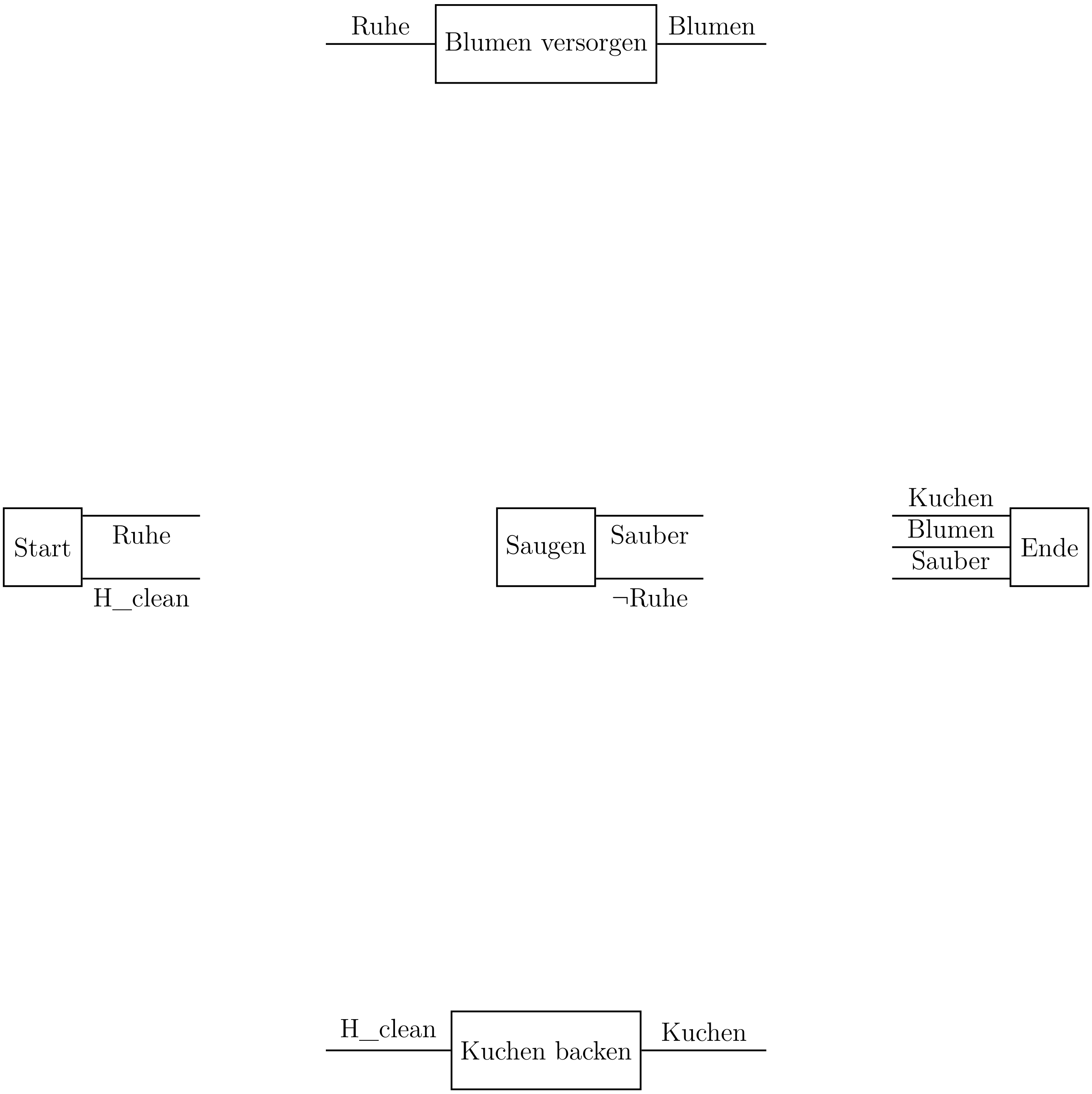
Dies führt hier bei allen absehbaren Kombinationsmöglichkeiten nicht zu einem Widerspruch, d.h. die Aussage $texttt Elefant\_rot$ ist nicht aus der Wissensbasis ableitbar. Wäre dies der Fall, müsste es bei wiederholter Anwendung der Resolution zu einer leeren Klausel kommen (Wahrheitswert False). Dies würde die Wissensbasis insgesamt False machen (aufgrund der Konjunktiven Normalform), was einen Widerspruch darstellt, da die Wissensbasis wahr sein muss. Dann wäre der gewünschte Satz bewiesen.

4.4 Problemart

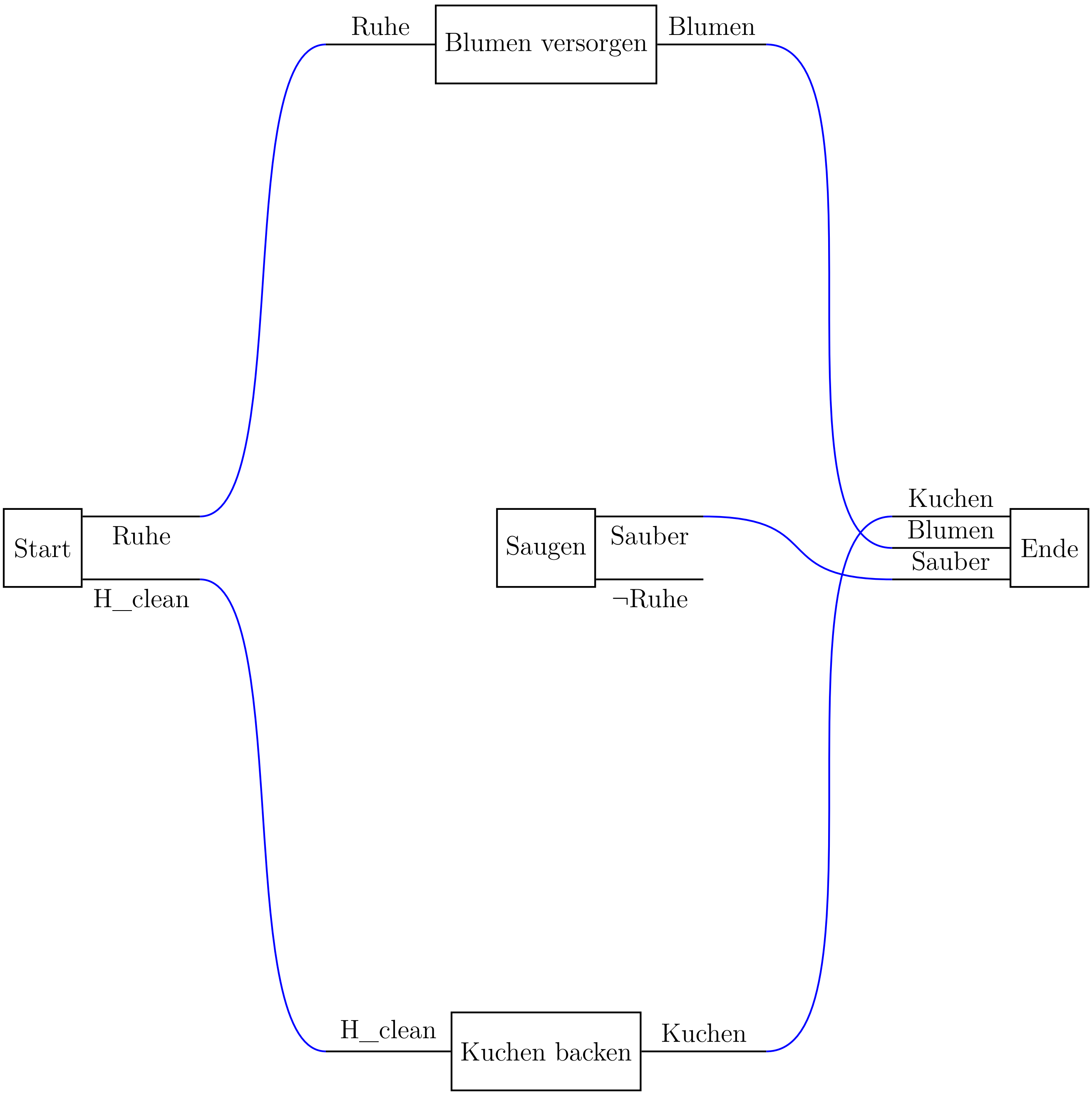
→ Constraint Problem bzw. Erfüllbarkeitsproblem der Aussagenlogik (da jede Variable nur 2 Werte besitzen kann).

5 Aufgabe

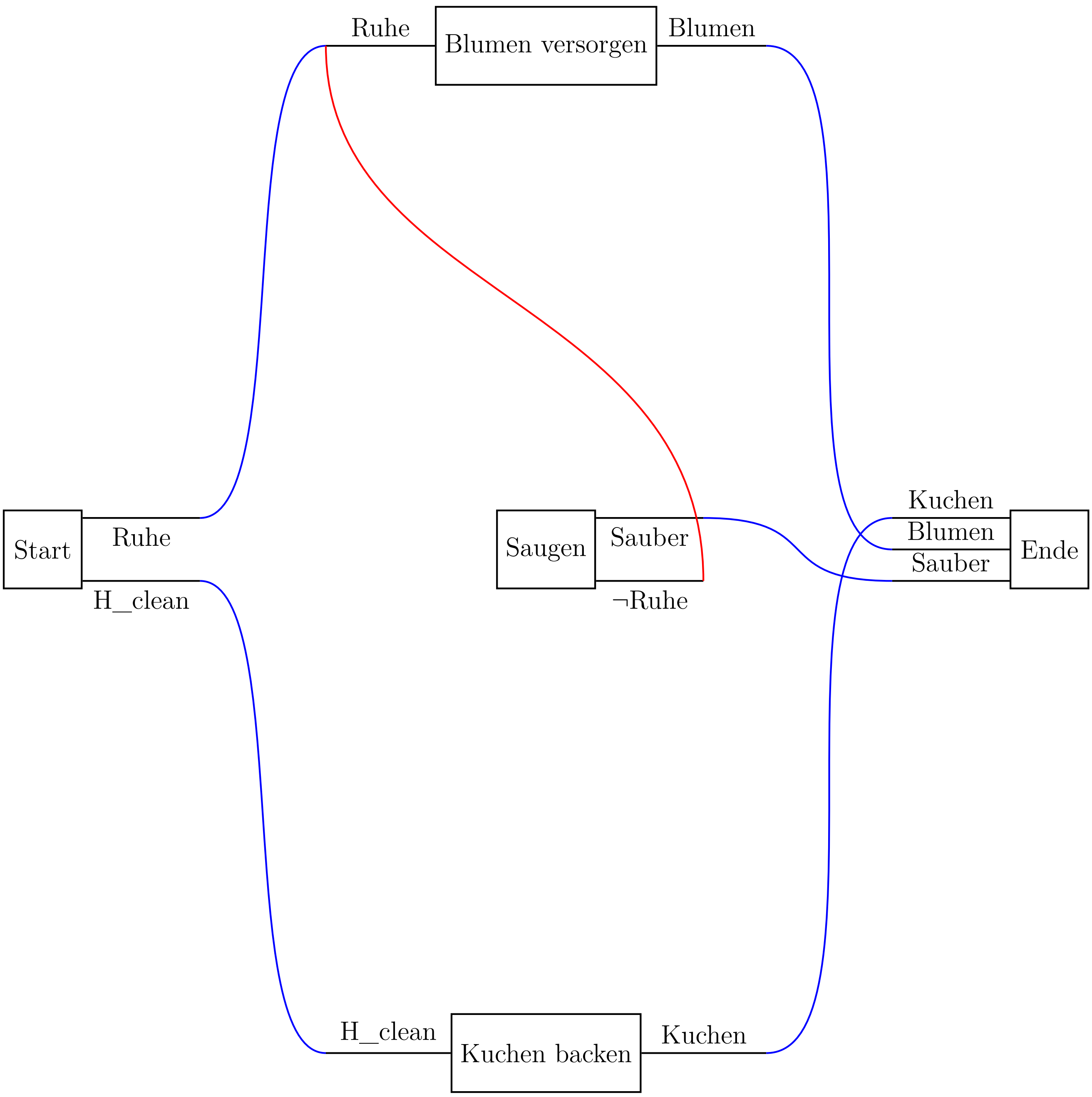
5.1 Einzeichnen von Vorbedingungen und Effekten



5.2 Causal Links



5.3 Threads



5.4 Ordnungsrelationen

- Blumen versorgen < Saugen

5.5 Linearisierungen

- Blumen versorgen → Saugen → Kuchen backen

- Kuchen Backen → Blumen versorgen → Saugen
- Blumen versorgen → Kuchen Backen → Saugen