

Übungsblatt 4

Bearbeiten Sie alle Aufgaben aller Übungsblätter – soweit nicht anders angegeben – in festen Zweiergruppen. Beide Gruppenmitglieder müssen die Aufgabe gemeinsam bearbeiten und die Lösung alleine erläutern können.

Das sogenannte Einstein-Rätsel, obwohl es nicht von Einstein ist ...

- 1) There are five houses **in a row**, each of a **different color** (red, green, ivory, blue, yellow) and inhabited by men of **different nationalities** (Englishman, Spaniard, Ukrainian, Norwegian, Japanese), with **different pets** (dog, snake, zebra, fox horse), **drinks** (tea, orange juice, milk, water, coffee), and **cigarettes** (Old Gold, Kools, Chesterfield, Lucky Strike, Parliament).
- 2) The Englishman lives in the red house.
- 3) The Spaniard owns the dog.
- 4) Coffee is drunk in the green house.
- 5) The Ukrainian drinks tea.
- 6) The green house is immediately to the right of the ivory house.
- 7) The Old Gold smoker owns the snake.
- 8) Kools are smoked in the yellow house.
- 9) Milk is drunk in the middle house.
- 10) The Norwegian lives in the first house on the left.
- 11) The man who smokes Chesterfields lives in the house next to the man with the fox.
- 12) Kools are smoked in the house next to the house where the horse is kept.
- 13) The Lucky Strike smoker drinks orange juice.
- 14) The Japanese smoke Parliaments.
- 15) The Norwegian lives next to the blue house.

Dieses Rätsel ist auf 2 Wegen (komplett) zu lösen:

- eine generate & test - Lösung
- eine weitere Lösung, bei der Sie Ihre Prolog- und Constraint-Kenntnisse nutzen sollen. Bitte benutzen Sie für die Constraint-Programmierung die Prolog Bibliothek clpfd.

Hinweis:

das Rätsel ist **komplett** zu lösen, d.h. es werden **keine** Anfragen gestellt wie „Wer besitzt die Schlange?“.

Es muss demzufolge eine „ordentliche“ Ausgabe erzeugt werden.

Beispielsweise startet der Aufruf von `?-solve.` die Suche und erzeugt eine Ausgabe derart:

Haus1: Red Norwegian Snake Water Chesterfield

Haus2:

Die Prolog-Constraint-Referenzlösung benötigt 10151 Inferenzen (inklusive der programmierten Ausgabe), die gilt es zu schlagen. Vorrang (vor einer "auf Biegen und Brechen" implementierten Minimierung der Inferenzen) soll aber auf jeden Fall der deklarative Charakter Ihrer Lösung haben und alles, was Sie bisher über „gutes Programmieren“ gelernt haben, die Zahl der Inferenzen ist nur zweitrangig.

Abnahme:

- Erläuterung der generate & test – Lösung
- Vorstellung und Erläuterung der Constraint-Lösung