

---

## Aufgabe: Eine Logelei

Die folgende Aufgabe ist dem Buch

*99 Logeleien von Zweistein*

entnommen, das 1968 von *Thomas von Randow* unter dem Pseudonym "Zweistein" im Verlag *Christian Wegner* veröffentlicht worden ist.

Die Herren Amann, Bemann, Cemann und Demann heissen — nicht unbedingt in derselben Reihenfolge — mit Vornamen Erich, Fritz, Gustav und Heiner. Sie sind alle verheiratet. Ausserdem weiss man über sie und ihre Ehefrauen noch dies:

1. Entweder ist Amanns Vorname Heiner, oder Bemanns Frau heisst Inge.
2. Wenn Cemann mit Josefa verheiratet ist, dann — und nur in diesem Falle — heisst Klaras Mann nicht Fritz.
3. Wenn Josefus Mann nicht Erich heisst, dann ist Inge mit Fritz verheiratet.
4. Wenn Luises Mann Fritz heisst, dann ist der Vorname von Klaras Mann nicht Gustav.
5. Wenn die Frau von Fritz Inge heisst, dann ist Erich nicht mit Josefa verheiratet.
6. Wenn Fritz nicht mit Luise verheiratet ist, dann heisst Gustavs Frau Klara.
7. Entweder ist Demann mit Luise verheiratet, oder Cemann heisst Gustav.

Wie heissen die Herren mit vollem Namen, wie ihre Ehefrauen mit Vornamen?

Verwenden Sie zur Lösung des Problems die Vorlage, die Sie im Netz unter der Adresse

[github.com/karlstroetmann/Logik/blob/master/Aufgaben/Blatt-8/logelei-frame.stlx](https://github.com/karlstroetmann/Logik/blob/master/Aufgaben/Blatt-8/logelei-frame.stlx)

finden. Dieser Rahmen enthält bereits eine Implementierung der Funktion

`parseKNF(s),`

die eine Formel, die als String *s* gegeben ist, in eine Menge von Klauseln umwandelt. Weiter enthält der Rahmen die Implementierung der Funktion

`davisputnam(clauses, literals),`

die für eine gegebene Menge von Klauseln eine Lösung berechnet. Wenn Sie diese Funktion verwenden, sollten Sie für den Parameter *literals* die leere Menge einsetzen.

Bei der Lösung dieser Aufgabe sollten Sie die folgenden Terme als aussagenlogische Variablen verwenden:

1.  $\text{Name}(x, y)$  ist genau dann wahr, wenn der *x* der Vorname und *y* der Nachname einer der Herren ist. Beispielsweise ist

`Name("Heiner", "Amann")`

genau dann wahr, wenn Herr Amann mit Vornamen *Heiner* heisst.

2.  $\text{Ehe}(x, y)$  ist genau dann wahr, wenn der Mann, der mit Vornamen *x* heisst, mit der Frau, die mit Vornamen *y* heisst, verheiratet ist. Beispielsweise wäre

`Ehe("Heiner", "Luise")`

genau dann wahr, wenn Heiner mit Luise verheiratet ist.

---

Zur Lösung des Problems sollten Sie die folgenden Teilaufgaben bearbeiten:

- (a) Implementieren Sie eine Funktion `exactlyOne` mit der Sie ausdrücken können, dass zu jedem Vornamen genau ein Nachname gehört und dass jeder Mann mit genau einer Frau verheiratet ist. Die Funktion `exactlyOne` wird in der Form

$$\text{exactlyOne}(a, b, fct)$$

aufgerufen. Hierbei sind  $a$  und  $b$  Mengen von Strings. Beispielsweise könnte  $a$  die Menge der männlichen Vornamen und  $b$  die Menge der weiblichen Vornamen sein. In diesem Fall wäre  $fct$  der String "Ehe". Die Funktion `exactlyOne` würde dann eine Menge von Klauseln erzeugen, die ausdrückt, dass jeder Mann aus der Menge  $a$  mit genau einer Frau aus der Menge  $b$  verheiratet ist.

**Hinweis:** Mit Hilfe der SETLX-Funktion `makeTerm` können Sie die Terme erzeugen, mit denen wir die aussagenlogischen Variablen darstellen. Beispielsweise erzeugt der Aufruf

$$\text{makeTerm}(\text{"Ehe"}, [\text{"Heiner"}, \text{"Luise"}])$$

den Term

$$\text{Ehe}(\text{"Heiner"}, \text{"Luise"}),$$

der ausdrückt, dass Heiner mit Luise verheiratet ist.

Sie finden in Zeile 195 das Gerüst der Funktion `exactlyOne`.

- (b) Implementieren Sie eine Funktion `isWifeOf`, so dass der Aufruf

$$\text{isWifeOf}(y, z)$$

eine Formel berechnet, die ausdrückt, dass  $y$  die Frau von  $z$  ist. Hierbei ist  $z$  der Nachname eines Mannes. Diese Formel soll als String dargestellt werden.

Sie finden in Zeile 205 das Gerüst der Funktion `isWifeOf`.

- (c) Implementieren Sie eine Funktion `exclusiveOr`, so dass der Aufruf

$$\text{exclusiveOr}(a, b)$$

für zwei aussagenlogische Formeln  $a$  und  $b$ , die als Strings vorliegen, eine Formel berechnet, die genau dann wahr ist, wenn entweder  $a$  oder  $b$  wahr ist. Die resultierende Formel soll in konjunktiver Normalform zurück gegeben werden.

Sie finden in Zeile 214 das Gerüst der Funktion `exclusiveOr`.

- (d) Vervollständigen Sie nun die Implementierung der Funktion `computeClauses` in Zeile 220. Wenn Sie alles richtig gemacht haben, berechnet der Aufruf der Funktion `solve` in Zeile 250 die Lösung des Problems.