

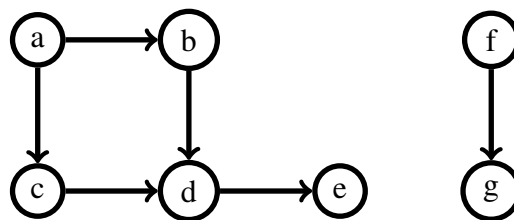
**Übungen zur Vorlesung**  
**Deklarative Programmierung: Sommersemester 2018**

Nr. 10, Abgabe bis 9.7.2017 11:59 Uhr

**Aufgabe 10.1:** Count Richtula

4 Punkte

In Racket haben Sie bereits einen Algorithmus kennengelernt, der überprüft, ob in einem gerichteten Graphen ein Weg zwischen zwei Knoten existiert. Das gleiche Problem möchten wir nun in Prolog betrachten.



- Der Graph soll lediglich durch eine Kantenmenge repräsentiert werden. Modellieren Sie den abgebildeten Graphen mithilfe einer binären Relation.
- Schreiben Sie ein Prädikat `connected(N1, N2)`, das erfüllt ist, falls ein Weg von N1 zu N2 existiert.
- Schreiben Sie ein Prädikat `path(N1, N2, PATH)`, das erfüllt ist, falls PATH eine Liste von Knoten ist, über die N1 und N2 verbunden sind.

**Aufgabe 10.2:** Zahlenspiele

4 Punkte

Betrachten Sie das folgende Programm zur Verarbeitung natürlicher Zahlen:

```
nat(0).  
nat(s(X)) :- nat(X).  
  
add(0, X, X) :- nat(X).  
add(s(X), Y, s(Z)) :- add(X, Y, Z).
```

Erweitern Sie das Programm um die folgenden Prädikate (verwenden Sie für die Implementierung die oben genannten Prädikate):

- `even(X)` und `odd(X)`, welche überprüfen, ob die übergebene Zahl gerade bzw. ungerade ist.
- `mult(X, Y, Z)` und `pow(X, N, Z)`, welche die Relationen  $X * Y = Z$  bzw.  $X^N = Z$  abbildet.

- c) `fib(N, F)`, welche n-te Fibonacci-Zahl (F) ermittelt

### Aufgabe 10.3: Listige Prädikate

4 Punkte

Schreiben Sie folgende Listenprädikate:

- a) `without(List, Element, Result)`, das erfüllt ist, falls Result durch Entfernen des übergebenen Elements aus List hervorgeht.
- b) `double(List, LL)`, das erfüllt ist, wenn jedes Element aus List zwei mal in LL enthalten ist, z.B.: `double([1, 2, 3], [1, 1, 2, 2, 3, 3])`.
- c) `sum(ListOfNat, Sum)`, das erfüllt ist, falls die Summe der Listenelements gleich Sum sind. Verwenden Sie die Darstellung der natürlichen Zahlen aus Aufgabe 10.2.

### Aufgabe 10.4: Deutschland ist BUNT...

4 Punkte

Wir wollen nun mit Prolog überprüfen, ob wir die Deutschlandkarte so einfärben können, dass zwei benachbarte Bundesländer unterschiedliche Farben haben.

- a) Überlegen Sie sich zunächst ein Format für Ihre Bundesländer. Zum Lösen der Aufgabe sollten Sie beachten, dass Sie den Namen des Bundeslandes, die Farbe, sowie die benachbarten Bundesländer benötigen.
- b) Schreiben Sie ein `pick(Col, ColList, Rest)`-Prädikat, welches sich aus der Liste der Farben eine Farbe aussucht und die Liste der Restfarben zurückliefert und ein `grouplist(L1, L2)`-Prädikat, welches alle Elemente der ersten Liste paarweise mit allen Elementen der zweiten Liste verknüpft.
- c) Schreiben Sie nun ein `colReg(Land, Farben, Zuordnung)`-Prädikat, welches eine Region und dessen Nachbarn einfärbt.
- d) Schreiben Sie nun ein `col(Laender, Farben, Zuordnung)`-Prädikat, welches alle oberen Hilfsprädikate verwendet, um das Farbproblem der Deutschlandkarte mit lediglich 4 Farben zu lösen.