

### Aufgabe 13 (7 Punkte)

Gegeben sei die folgenden, auch von der Veranstaltungsseite herunterladbaren Fakten in einer neuen Datei AufgabeErsteAnfragen2.pl, mit den später genannten Tests. Bei den Anfragen ist es kein Problem, wenn einzelne Ausgaben mehrfach erscheinen. Beachten Sie, dass nicht nur die Tests laufen müssen sondern Ihre Prädikate auch Lösungen berechnen können sollen.

```
%film(Film-Id, Filmname, FSK ab)
film(42, 'Wombat', 16).
film(43, 'Schnuddel', 0).
film(44, 'Hase', 18).
film(45, 'Wut', 0).
film(46, 'Wurmi', 16).
film(47, 'Schnuddel', 18).

%kino(Name, Ort, Film-Id eines aktuell gezeigten Films)
kino(gloria, os, 42).
kino(gloria, os, 43).
kino(gloria, os, 47).
kino(gloria, ol, 42).
kino(gloria, ol, 44).
kino(apollo, os, 43).
kino(apollo, os, 45).
```

- Schreiben Sie eine Anfrage die unter anderem die Ids unterschiedlicher Filme ausgibt, die den gleichen Namen haben. („unter anderem, da noch weitere Werte mit ausgegeben werden)
- Schreiben Sie eine Anfrage, die die Namen aller Orte ausgibt, an denen mindestens zwei Kinos existieren. Sie können davon ausgehen, dass es an einem Ort nicht zwei Kinos mit dem gleichen Namen gibt.
- Schreiben Sie ein Prädikat `alleFilme/2`, das alle Kinonamen und die Namen der dort gespielten Filme verknüpft. Folgende Tests sollen ebenfalls laufen.

```
?- alleFilme(X, Y).
X = gloria,
Y = 'Wombat' ;
X = gloria,
Y = 'Schnuddel' . % gibt weitere Ergebnisse

testAlleFilme :-
alleFilme(gloria, 'Wombat'),
alleFilme(gloria, 'Schnuddel'),
alleFilme(apollo, 'Wut'),
not(alleFilme(gloria, 'Wut')).
```

Bei der Ausführung des Tests sollte einfach das Ergebnis true, eventuell mehrfach hintereinander erscheinen.

```
?- testAlleFilme.
true ;
true ;
true ;
true ;
false.
```

- d) Schreiben Sie ein Prädikat `laeuftIn/2`, das Ids von Filmen und Orte verknüpft, wobei der zur Id gehörende Film in diesem Ort läuft. Folgende Tests sollen laufen.

```
?- laeuftIn(X, Y).  
X = 42,  
Y = os ;  
X = 43,  
Y = os . % gibt weitere Ergebnisse
```

```
testLaeuftIn :-  
    laeuftIn(42,os),  
    laeuftIn(42,ol),  
    laeuftIn(47,os),  
    not(laeuftIn(43,ol)).
```

- e) Schreiben Sie ein Prädikat `hatKinderfilm/2`, die einen Kinonamen und den zum Kino gehörenden Ort enthält, in dem mindestens ein Film mit FSK 0 läuft. Folgende Tests sollen laufen.

```
?- hatKinderfilm(A, B).  
A = gloria,  
B = os ;  
A = apollo,  
B = os . % gibt weiteres Ergebnis
```

```
testHatKinderfilm :-  
    hatKinderfilm(gloria,os),  
    hatKinderfilm(apollo,os),  
    not(hatKinderfilm(gloria,ol)).
```

- f) Schreiben Sie ein Prädikat `gleicheFilme/2`, das zwei Kinonamen von unterschiedlichen Kinos verknüpft, in denen zumindest ein gleicher Film läuft. Folgende Tests sollen laufen. (Sie können die Annahme, dass es nicht zwei Kinos mit dem gleichen Namen in einer Stadt gibt, nutzen.)

```
?- gleicheFilme(A, B).  
A = B, B = gloria . % gibt weitere Ergebnisse
```

```
testGleicheFilme :-  
    gleicheFilme(gloria, gloria), % unterschiedliche Kinos  
    not(gleicheFilme(apollo, apollo)).
```

- g) Schreiben Sie ein Prädikat `laeuftNicht/1`, das die Ids der Filme enthält, die in keinem Kino laufen. Dabei kann der Anfang der Ergebnisausgabe z. B. wie folgt aussehen.

```
?- laeuftNicht(A).  
A = 46 .
```

Folgende Tests sollen laufen.

```
testLaeuftNicht :-  
    laeuftNicht(46),  
    not(laeuftNicht(47)),  
    not(laeuftNicht(42)).
```

- h) Schreiben Sie ein Prädikat `laeuftNichtIn/2`, das Ids von Filmen und Orte verknüpft, wobei der zur Id gehörende Film in diesem Ort nicht läuft. Dabei kann der Anfang der Ergebnisausgabe z. B. wie folgt aussehen.

```
?- laeuftNichtIn(A, B).  
A = 43,  
B = ol ; % gibt weitere Ergebnisse
```

Folgende Tests sollen laufen.

```
testLaeuftNichtIn :-  
    laeuftNichtIn(47, ol),  
    laeuftNichtIn(45, ol),  
    laeuftNichtIn(46, os),  
    not(laeuftNichtIn(44, ol)),  
    not(laeuftNichtIn(45, os)).
```

- i) Schreiben Sie ein Prädikat `hatNurKinderfilme/2`, das einen Kinonamen und den zum Kino gehörenden Ort enthält, in dem nur Filme mit FSK 0 laufen. Folgende Tests sollen laufen. Denken Sie daran, dass sich `not/1` nur auf ein Prädikat bezieht. Sollten sie eine Konjunktion mehrerer Prädikate negieren wollen, muss die Konjunktion in Klammern stehen. Dabei kann der Anfang der Ergebnisausgabe z. B. wie folgt aussehen.

```
?- hatNurKinderfilme(A, B).  
A = apollo,  
B = os . % gibt evtl. weiteres Ergebnis
```

Folgende Tests sollen laufen.

```
testHatNurKinderfilme :-  
    not(hatNurKinderfilme(gloria,os)),  
    hatNurKinderfilme(apollo,os),  
    not(hatNurKinderfilme(gloria,ol)).
```

### Aufgabe 14 (2 Punkte)

Die Tests zu folgenden Aufgaben können von der Veranstaltungsseite geladen werden.

- a) Schreiben Sie ein Prädikat `fak/2`, dass zu einer gegebenen Zahl die Fakultät berechnet.

```
?- fak(42, Y).  
Y = 140500611775287989854314260624451156993638400000000
```

Folgende Tests sollen laufen.

```
testFak :-  
    fak(0,1),  
    fak(1,1),  
    fak(4, 24),  
    fak(40, 81591528324789773434561126959611589427200000000) .
```

- b) Schreiben Sie ein Prädikat `fakk/2`, dass zu einer gegebenen Zahl die Fakultät berechnet, dazu aber ein weiteres Prädikat mit Akkumulator nutzt, mit dem die Fakultät aufsteigend berechnet wird. Folgende Tests sollen laufen.

```
testFakk :-  
    fakk(0,1),  
    fakk(1,1),  
    fakk(4, 24),  
    fakk(40, 81591528324789773434561126959611589427200000000) .
```

- c) Die Funktion `collatz` ist für natürliche Zahlen  $n > 0$  wie folgt definiert.

$$\text{collatz}(n) = \begin{cases} \text{collatz}(n / 2), & \text{falls } n \text{ gerade} \\ \text{collatz}(3*n + 1), & \text{falls } n \text{ ungerade} \end{cases}$$

Schreiben Sie ein Prädikat `collatz/1`, das die Funktion umsetzt und alle berechneten Zahlen nacheinander ausgibt. Die Berechnung soll enden, wenn eine der Zahlen 1, 2 oder 4 erreicht wird. Nutzen Sie folgende Beispielausgaben zur Überprüfung.

```
?- collatz(3).  
10 5 16 8 4  
true .
```

```
?- collatz(7).  
22 11 34 17 52 26 13 40 20 10 5 16 8 4  
true .
```

```
?- collatz(19).  
58 29 88 44 22 11 34 17 52 26 13 40 20 10 5 16 8 4  
true .
```

Zur Berechnung von Modulo-Werten wird `mod` genutzt, z. B. `Tmp is N mod 2`.  
Schauen Sie sich als Hintergrundinformation das Video <https://www.youtube.com/watch?v=094y1Z2wpJg> an.