

Präsenzübungen zur Vorlesung
Deklarative Programmierung: Sommersemester 2018
Nr. 9

Aufgabe 9.1: Marvel at Prolog

Entwickeln Sie ein Prolog-programm zur Lösung des folgenden Rätsels:

- Tony, Steve und Clint sind Personen
- Tony mag Sokovia-Accords und Freiheit
- Steve hasst Sokovia-Accords
- Avengers lieben Freiheit
- Freiheitskämpfer hassen Sokovia-Accords
- Wer Freiheit liebt, hasst die Sokovia-Accords
- **Wer ist Avenger und wer ist Freiheitskämpfer?**

Aufgabe 9.2: Ein Beispiel sagt mehr als 1000 Worte

Erklären Sie die folgenden Begriffe anhand eines Beispiels:

- Funktor
- Logische Variable
- Instanz eines Terms
- Existenz-Abfrage (existenziell quantifizierte Abfrage)
- Universeller Fakt
- Konjunktive Abfrage
- Horn-Klausel

Aufgabe 9.3: Old McDonald hat 'ne Farm ...

Die folgenden Prolog-Fakten definieren, von welchen Tieren und Pflanzen, welche Rohstoffe gewonnen werden können:

```
animal_gives (sheep, wool) .  
animal_gives (alpaca, wool) .  
animal_gives (sheep, meat) .
```

```
animal_gives(cow, meat) .  
plant_gives(cotton, wool) .  
plant_gives(wheat, flour) .
```

Schreiben Sie die folgenden Prozeduren:

- a) `resource(Resource)` : ist erfüllt für alle Rohstoffe `Resource`, die aus von einem Tier oder einer Pflanze gewonnen werden können.
- b) `farm_animal(Animal)` : ist erfüllt für alle Tiere `Animal`, von denen ein Rohstoff gewonnen werden kann.
- c) `crop_plant(Plant)` : ist erfüllt für alle Pflanzen `Plant`, von denen ein Rohstoff gewonnen werden kann.
- d) `herbal_animal(Resource)` : ist erfüllt für alle Rohstoffe `Resource`, für die es entweder ein Tier oder eine Pflanze gibt, wovon sie gewonnen werden können.
- e) `farmable(Source, Resource)` : ist erfüllt für alle Quellen (Tiere oder Pflanzen) `Source` Rohstoffe `Resource`, wenn der Rohstoff von der Quelle gewonnen werden kann.

Aufgabe 9.4: Setzende Schlussfigur

Erweitern Sie das Programm aus der vorige Aufgabe um die folgenden zwei Fakten und die Regel (die in der vorigen Aufgabe zu definierenden Prozeduren werden nicht benötigt):

```
food(meat) .  
food(flour) .
```

```
animal_gives_food(X, Y) :- animal_gives(X, Y), food(Y) .
```

Zeigen Sie mithilfe des Gesetzes des universellen Modus Ponens, dass die Abfrage
`?- animal_gives_food(X, meat) .` aus dem Program hergeleitet werden kann.