



Technische Universität Berlin

Software and Embedded Systems Engineering Group

Prof. Dr. Sabine Glesner

www.sese.tu-berlin.de Sekr. TEL 12-4 Ernst-Reuter-Platz 7 10587 Berlin



Softwaretechnik und Programmierparadigmen WiSe 2022/2023

Prof. Dr. Sabine Glesner
Milko Monecke
Simon Schwan

Übungsblatt 05

Aufgabe 1: Prolog – Variablen und Unifikation

Überlegt, ob die folgenden Ausdrücke gültig sind, ob sie unifizierbar sind und welches mögliche Variablenbelegungen sein könnten.

🗉 `?- X=1.`
 `?- x=1.`
 `?- _="unterstrich".`
 `?- mensch(X) == mensch(ich).`
 `?- val(X,Y)=val(A,B).`
 `?- val(X,Y,_)=val(X,Y).`
 `?- strcat(A," teil2 ",C)=strcat(" teil1 ",B," teil3 ").`

▶ `?- 1+1=2.`
 `?- 2.0 is 1.0+1.0.`
 `?- Erg is 1+1, Erg==2.`
 `?- A=a,a==A.`
 `?- a==A,a=A.`

Schlüssel:

- ▶ Ein ergänzendes Video wird zur Vor- oder Nachbereitung veröffentlicht.
- 🗉 Wird im Tutorium besprochen.

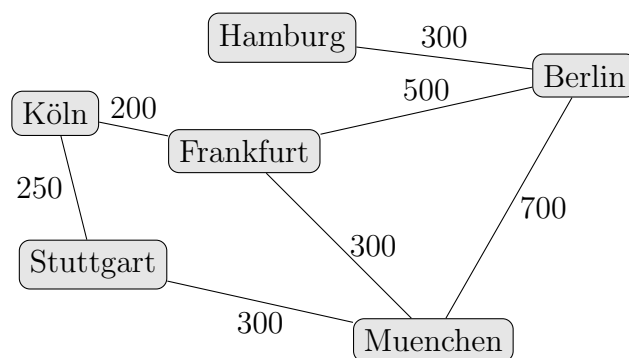
```

?- A(1,2,3)=funktion(1,2,3).
?- X=f(X).
?- (X=1;X=2;X=3), Y=X.
?- (X=1;X=2;X=3),! , Y=X.
?- (X=1;X=2;X=3), \+(X=2).
?- \+ (X=1;X=2;X=3).

```

Aufgabe 2: Prolog – Fakten

Die folgende Abbildung zeigt sehr grob einige Straßen in Deutschland und ihre Strecken. Wir wollen simple Funktionen zur Navigation entwickeln. Stellt dafür die Straßen mit ihren Längen als Fakten in Prolog dar. Zunächst reicht eine beliebige Richtung.








Mit dieser Datenbasis können nun mithilfe von Anfragen die ersten einfachen Fragen beantwortet werden:

- Gibt es eine Verbindung von Hamburg nach Berlin, die 300km lang ist?
- Gibt es eine Verbindung von München nach Frankfurt?
- Wie lang ist die Verbindung von Berlin nach München?
- In welche Städte kann man von Berlin aus direkt fahren?



Aufgabe 3: Prolog – Regeln

Nun sollen komplexere Anfragen an die Datenbasis möglich sein. Die folgenden Regeln sollen implementiert werden:

- a) Es soll ein Prädikat `directConnection(X,Y,S)` erstellt werden, durch das die Straßen beidseitig befahrbar werden. Hier soll die Reihenfolge der Städte egal sein, also ergibt `directConnection(X,Y,S)` immer das gleiche wie `directConnection(Y,X,S)`. 
- b) Ein Prädikat `dayTrip(X,Y,Max,S)` soll es ermöglichen, nur direkte Verbindungen zwischen zwei Städten auszugeben, die eine Maximallänge nicht überschreiten. 
- c) `oneStopConnection(A,B,L)` prüft, ob zwei Städte über genau einen Zwischenstopp verbunden sind. Die Gesamtlänge ist der Variable L zugewiesen. 
- d) `shortestOneStop(A,B,VIA,L)` gibt die Stadt VIA zurück, über die die kürzeste Verbindung mit genau einem Zwischenstop geht. 
- e) Mit dem Prädikat `pretty(X,Y)` sollen Daten aus der gegebenen Faktenbasis in einer übersichtlichen Form auf die Konsole geschrieben werden. Dabei sollen X und Y die ersten beiden Sub-Termen der Fakten entsprechen. 
- f) Mit dem Prädikat `dump(X,Y)` sollen Daten aus der gegebenen Faktenbasis analog zu `pretty(...)` ausgegeben werden, dabei aber alle möglichen Belegungen auf einmal (Lösung ohne `findAll(...)` oder Listen). 