

Proseminar Logische Programmierung
Elmar Eder, Universität Salzburg

Blatt 3, Aufgaben für 22.3.2021

Stellen Sie die Anfragen “?- A=5, A1=A+1.” und “?- A=5, A1 is A+1.”! In Prolog wird \leq als $=<$ geschrieben. Zum Beispiel gelingt die Anfrage “?- A=3, B=5, A=<B.”. Lassen Sie sich mit der Anfrage “?- X is 100 mod 7.” den Rest $X=2$ bei der ganzzahligen Division “100 : 7 = 14 Rest 2” ausgeben! Man sagt, eine natürliche Zahl t *teilt* eine natürliche Zahl n oder ist ein *Teiler* von n , wenn es eine natürliche Zahl x gibt mit $n = t \cdot x$. Beispiel: 5 ist ein Teiler von 100, weil $100 = 5 \cdot 20$. Testen Sie mit Prolog, ob 5 beziehungsweise 7 ein Teiler von 100 ist, indem Sie die Anfrage “?- 100 mod 5 =:= 0.” beziehungsweise “?- 100 mod 7 =:= 0.” stellen! Eine *zusammengesetzte* Zahl ist eine Zahl $x \cdot y$, wobei x und y natürliche Zahlen mit $x > 1$ und $y > 1$ ist. Eine *Primzahl* ist eine natürliche Zahl größer als 1, die nicht zusammengesetzt ist. In einigen Prologs gibt es ein eingebautes Prädikat `between/3`. Man ruft `between(A,B,X)` mit gegebenen ganzen Zahlen A und B auf. `between(A,B,X)` gilt dann, wenn X eine ganze Zahl ist mit $A \leq X \leq B$. Probieren Sie die Anfragen “?- between(5,8,7).” und “?- between(5,8,X).” und lassen Sie sich dabei jeweils alle Lösungen ausgeben!

Aufgabe 1 Sie sollen das nun selbst als Prädikat `zwischen/3` implementieren ohne das eingebaute Prädikat `between/3` zu verwenden. Warum liefert das Programm “`zwischen(A,B,X):- A=<X, X=<B.`” zwar bei der Anfrage “?- zwischen(5,8,7).” das richtige Ergebnis, aber bei der Anfrage “?- zwischen(5,8,X).” eine Fehlermeldung? Man kann das Prädikat `zwischen/3` induktiv durch zwei Regeln charakterisieren:

- A ist zwischen A und B , wenn $A \leq B$ ist.
- X ist zwischen A und B , wenn $A < B$ ist und X zwischen $A + 1$ und B ist.

Übersetzen Sie diese zwei Regeln in ein Prologprogramm und testen Sie Ihr Programm mit Anfragen wie “?- zwischen(5,8,7).” und “?- zwischen(5,8,X).”!

Aufgabe 2 Definieren Sie ein Prädikat `teilt/2` zum Testen, ob eine Zahl t eine Zahl n teilt! Die Anfrage “?- teilt(5,100).” sollte gelingen, also die Antwort “yes” liefern. Die Anfrage “?- teilt(7,100).” sollte fehlschlagen, also die Antwort “no” liefern. Verwenden Sie dieses Prädikat und das Prädikat `between/3` oder `zwischen/3` zur Definition eines Prädikats `teiler/2`, das das gleiche leistet wie das Prädikat `teilt/2`, aber zusätzlich auch verwendet werden kann, um sich alle Teiler einer Zahl n anzeigen zu lassen! Die Anfrage “?- teiler(T,100).” sollte bei wiederholter Eingabe eines Strichpunkts alle Teiler von 100 liefern.

Aufgabe 3 Definieren Sie ein Prädikat `zusammengesetzt/1` zum Testen, ob eine Zahl zusammengesetzt ist! Fügen Sie Ihrem Programm die Regel

```
primzahl(P):- P>1, \+ zusammengesetzt(P).
```

hinzu! Für ein Ziel (goal) G ist $\backslash+G$ die “negation as failure” von G . Beim Aufruf von $\backslash+G$ ruft Prolog zunächst das Ziel G auf. Wenn G gelingt, schlägt $\backslash+G$ fehl und umgekehrt. Definieren Sie ein Prädikat zur Ausgabe z.B. aller Primzahlen bis 100 mit “?- primzahl_kleinergleich(P,100).” und wiederholte Eingabe eines Strichpunkts! Probieren Sie auch “?- findall(P, primzahl_kleinergleich(P,100), Ps).”!