

Division in Prolog

Übungsblatt 8

ERIC KUNZE — 8. JUNI 2021

Dieses Werk ist lizenziert unter einer Creative Commons “Namensnennung – Nicht-kommerziell – Weitergabe unter gleichen Bedingungen 4.0 International” Lizenz.



Keine Garantie auf Vollständigkeit und/oder Korrektheit!

Aufgabe 2

Aufgabe. Natürliche Zahlen stellen wir in Prolog⁻ als Terme über dem einstelligen Funktionsymbol s und dem nullstelligen Funktionssymbol 0 dar:

```
nat(0).  
nat(s(X)) :- nat(X).
```

Dabei kürzen wir wie in der Vorlesung den Term für die natürliche Zahl n mit $\langle n \rangle$ ab, z.B. $s(s(s(0))) = \langle 3 \rangle$. Weiterhin wurde das Prädikat `sum` besprochen:

```
sum(0, Y, Y) :- nat(Y).  
sum(s(X), Y, s(S)) :- sum(X, Y, S).
```

- (a) Geben Sie ein Prädikat `even` an, das für alle geraden natürlichen Zahlen gilt und für alle ungeraden natürlichen Zahlen nicht.
- (b) Geben Sie eine zweistellige Relation `div2` an, die für jede natürliche Zahl n das Paar $(\langle n \rangle, \lfloor \frac{n}{2} \rfloor)$ enthält und sonst nichts.
- (c) Geben Sie eine dreistellige Relation `div` an, die für jedes Paar von natürlichen Zahlen n und m , wobei $m \neq 0$, das Tripel $(\langle n \rangle, \langle m \rangle, \lfloor \frac{n}{m} \rfloor)$ enthält und sonst nichts. Nutzen Sie dafür die dreistellige Relation `sum` aus der Vorlesung.

Hinweis: Definieren Sie zunächst eine zweistellige Relation `1t`, die $\langle \rangle$ modelliert.

Teilaufgabe (a)

Die Geradheit einer natürlichen Zahl bleibt unverändert, wenn ich 2 addiere oder subtrahiere, d.h. beispielsweise

$$\begin{aligned} 1 \text{ ungerade} &\Rightarrow 3 \text{ ungerade} \Rightarrow \dots \\ 0 \text{ gerade} &\Rightarrow 2 \text{ gerade} \Rightarrow \dots \end{aligned}$$

Genau dies wollen wir nun ausnutzen und insbesondere die zweite Zeile für das Prädikat `even` nutzen. Wenn eine Zahl N gerade ist, dann ist auch $N + 2$ gerade. Nun können wir also notieren

`even(s(s(N))) :- even(N)`.. Damit erhalten wir eine schöne Reduktion des Problems: das “Argument” von `even` wird immer kleiner. Nun merkt der Haskell-Kenner schon was fehlt: ein “Basisfall”. Den finden wir hier ganz einfach mit der kleinsten natürlichen, geraden Zahl: die Null. Also fügen wir noch den Fakt `even(0)` hinzu und sind fertig.

```
even(0).
even(s(s(N))) :- even(N).
```

Teilaufgabe (b)

Wir betrachten die Division durch 2 und machen folgende Beobachtung, die uns zum Ziel einer Reduktion führen soll:

$$\frac{N}{2} = Q \Leftrightarrow \frac{N-2}{2} = Q-1$$

Wenn ich also von Zähler zwei abziehe, dann verringert sich das Ergebnis um 1. Nun wandeln wir die Subtraktion noch in eine Addition um und erhalten mit

$$\frac{N+2}{2} = Q+1 \Leftrightarrow \frac{N}{2} = Q$$

quasi die definierende Aussage für unsere Prolog-Relation. Nun haben wir noch ein Problem: die Abrundung. Uns fehlt aber genauso noch ein Basisfall – suchen wir ihn doch. Und wie so oft nehmen wir doch einfach mal die Null und überlegen, was ich für die Division $\frac{0}{2}$ erhalte. Natürlich $\frac{0}{2} = 0$ und wir können den Fakt `div2(0,0)` hinzufügen. Aber wir können durch die Reduktion um 2, wie wir sie uns überlegt haben auch in den Fall laufen, dass wir nicht in der Null rauskommen, sondern in der Eins. Aber was ist denn $\lfloor \frac{1}{2} \rfloor$? Richtig, ebenso Null. Also kommt noch der Fakt `div2(s(0),0)` hinzu und wir erhalten insgesamt:

```
div2(0, 0).
div2(s(0), 0).
div2(s(s(N)), s(Q)) :- div2(N, Q).
```

Damit reduzieren wir also gemäß der Regel immer weiter den Zähler um zwei und das Ergebnis um eins, am Ende landen wir, sofern die richtigen Zahlen eingegeben wurden, bei einem der beiden Fakten.

Teilaufgabe (c)

Wir folgen dem Hinweis und überlegen uns die $<$ -Relation. Dafür nutzen wir folgende Idee:

$$N < M \Leftrightarrow N + 1 < M + 1$$

Reduzieren wir so eine Anfrage immer weiter, sollten wir am Ende in der ersten Komponente eine Null erhalten. Würde die zweite Komponente eher Null werden, würde das ja heißen, die zweite Komponente wäre die kleinere, aber das widerspricht ja gerade $N < M$. Also erhalten wir:

$$\begin{aligned} \text{lt}(0, \text{s}(M)) & :- \text{nat}(M). \\ \text{lt}(\text{s}(N), \text{s}(M)) & :- \text{lt}(N, M). \end{aligned}$$

So jetzt können wir uns mal überlegen, wann denn bei der Division immer Null rauskommt: Fall 1 - wenn der Zähler Null ist. Die einzige Einschränkung ist dann noch, dass der Nenner nicht Null sein darf, das können wir mit $0 < \text{Nenner}$ abfangen. Also:

$$\text{div}(0, M, 0) :- \text{lt}(0, M).$$

Da wir wieder Abrunden sollen, haben wir noch einen zweiten Fall, bei dem immer Null rauskommt. Nämlich wenn der Zähler kleiner als der Nenner ist, dann ist das Ergebnis (im Reellen) $0, \dots$ und das wird abgerundet zu Null. Also:

$$\text{div}(N, M, 0) :- \text{lt}(N, M).$$

So jetzt der allgemeine, komplizierte Fall: Wir versuchen mal die Idee von `div2` zu verallgemeinern: Dort haben wir durch 2 dividiert und dementsprechend vom Zähler immer zwei abgezogen um das Ergebnis um 1 zu verringern. Jetzt dividieren wir nicht mehr durch 2 sondern durch ein beliebiges M . Also müssen wir nicht mehr 2 im Zähler subtrahieren, sondern M .

$$\begin{aligned} \frac{N}{M} = Q & \Leftrightarrow \frac{N - M}{M} = Q - 1 \\ \frac{N + M}{M} = Q + 1 & \Leftrightarrow \frac{N}{M} = Q \end{aligned}$$

Nennen wir die Differenz mal V , also $N - M = V$ oder als Summe geschrieben (da wir das Prädikat `sum` schon haben): $M + V = N$. Dann haben wir bei `div2` mit dem um zwei verringerten Ergebnis weiter dividiert, also müssen wir nun mit dem um M verringerten Ergebnis weiter dividieren: das ist aber gerade das V . Teilen wir also weiter $\frac{V}{M}$, dann erhalten wir das um eins verringerte Ergebnis. Vorher war das Ergebnis `s(Q)` - um eins verringert also Q . Natürlich sollte der Nenner auch hier immer positiv sein, also $0 < M$. Und dann erhalten wir schließlich:

$$\text{div}(N, M, \text{s}(Q)) :- \text{lt}(0, M), \text{sum}(M, V, N), \text{div}(V, M, Q).$$

Alles zusammen sieht dann also so aus:

```
1  nat(0).
2  nat(s(X)) :- nat(X).
3
4  sum(0, Y, Y) :- nat(Y).
5  sum(s(X), Y, s(S)) :- sum(X, Y, S).
6
7  even(0).
8  even(s(s(N))) :- even(N).
9
10 div2(0, 0).
11 div2(s(0), 0).
12 div2(s(s(N)), s(M)) :- div2(N, M).
13
14 lt(0, s(M)) :- nat(M).
15 lt(s(N), s(M)) :- lt(N, M).
16
17 div(0, M, 0) :- lt(0, M).
18 div(N, M, 0) :- lt(N, M).
19 div(N, M, s(Q)) :- lt(0, M), sum(M, V, N), div(V, M, Q).
```