

## Übungen zur „Deskriptiven Programmierung“ Blatt 6

### Aufgabe 11.

*Logische Programme* lassen sich in Haskell wie folgt darstellen:

```
data Literal a    = Rel Symbol [Term a]  
data Clause a   = Literal a :<= [Literal a]  
type Program a = [Clause a]
```

Entwerfe eine konkrete Syntax für Logische Programme (in EBNF). Folgende Punkte sind mindestens zu klären: Wie werden Konstanten von Variablen unterschieden? Dürfen Funktions- oder Relationssymbole infix notiert werden (wie wird die Assoziativität und die Bindungsstärke bestimmt)?

Schreibe Funktionen, die die konkrete Syntax in die abstrakte Syntax überführen und umgekehrt.

*Tipp:* Es gibt verschiedene Haskell Bibliotheken, die einem bei der Programmierung von *Parseern* (konkrete in abstrakte Syntax) und *Pretty-Printern* (abstrakte in konkrete Syntax) unterstützen.