

Übungen zur „Deskriptiven Programmierung“ Blatt 5

Aufgabe 8. *Formeln erster Stufe* lassen sich in Haskell wie folgt darstellen:

```
data Formula a = F
                | T
                | Rel Symbol [Term a]
                | Not (Formula a)
                | And (Formula a) (Formula a)
                | Or (Formula a) (Formula a)
                | Then (Formula a) (Formula a)
                | Iff (Formula a) (Formula a)
                | Forall a (Formula a)
                | Exists a (Formula a)
```

Definiere folgende Funktionen:

$$\begin{aligned} free &:: Formula\ a \rightarrow [a] \\ closed &:: Formula\ a \rightarrow Bool \end{aligned}$$

Die Funktion *free* bestimmt die freien Variablen einer gegebenen Formel; *closed* überprüft, ob eine Formel geschlossen ist.

Aufgabe 9. Definiere eine Funktion, die eine Formel in eine semantisch äquivalente *rektifizierte* Formel überführt.

$$rectify \quad :: \quad (Eq\ a) \Rightarrow Formula\ a \rightarrow Formula\ a$$

Aufgabe 10. Definiere Funktionen, die eine gegebene Formel in die Pränex-Normalform bzw. in die konjunktive Normalform überführen.

$$\begin{aligned} pnf &:: Formula\ a \rightarrow Formula\ a \\ cnf &:: Formula\ a \rightarrow Formula\ a \end{aligned}$$

Wie lassen sich Formeln in konjunktiver Normalform ‘kompakter’ darstellen?