Prof. Martin Hofmann, PhD
Dr. Steffen Jost

Ludwig-Maximilians-Universität München
Institut für Informatik
16. Juni 2015

8. Übung zur Vorlesung

Programmierung und Modellierung

A8-1 ***Unifikation I*** Berechnen Sie jeweils den allgemeinsten Unifikator für die folgenden Typgleichungen mit Robinson's Unifikationsalgorithmus, Folie 08-48, falls möglich:

a) $\{\texttt{Int} \to (\alpha \to \beta) = \alpha \to (\beta \to \texttt{Int})\}$

b) $\{\alpha \to (\alpha \to \texttt{Int}) = (\texttt{Int} \to \beta) \to \beta\}$

c) $\{\alpha \to (\alpha \to \texttt{Int}) = (\texttt{Int} \to \beta) \to \gamma\}$

A8-2 ***Typherleitung I***

a) Erstellen Sie eine Typherleitung in Baum-Notation für folgendes Typurteil:

$$\{\,\} \vdash (\backslash x \texttt{ -> } (\backslash y \texttt{ -> } (\backslash z \texttt{ -> } x\ z\ y))) :: (\alpha \to \beta \to \delta) \to \beta \to \alpha \to \delta$$

Zusatzfrage: Welchen Namen hat diese Funktion in der Standardbibliothek von Haskell?

b) Erstellen Sie eine Typherleitung in linearer Notation für folgendes Typurteil:

$$\{x{::}\texttt{Bool}, y{::}\texttt{Double} \to \texttt{Int}\} \vdash (\backslash z \texttt{ -> } z\ x\ 4\ (y\ 7)) :: (\texttt{Bool} \to \texttt{Int} \to \texttt{Int} \to \beta) \to \beta$$

Hinweis: Beachten Sie dabei die übliche Klammerkonventionen für Funktionstypen und Funktionsanwendung!

c) Beweisen Sie durch eine saubere Typherleitung in der Notation Ihrer Wahl, dass die folgende Haskell Funktion den behaupteten Typ hat:

```
twice :: (a -> a) -> a -> a
twice f x = f (f x)
```

Hinweis: Da das in der Vorlesung behandelte Typsystem nicht mit benannten Funktionen umgehen kann, müssen wir zuerst den Funktionsrumpf in eine äquivalenten Term übersetzen, welcher eine anonyme Funktionsdefinition mit Lambda benutzt.

A8-3 ***Monadische Komposition*** In der Vorlesung am 15. Juni wurde das Prinzip demonstriert, wie man mehrere `IO`-Aktionen zu einer einzelnen Aktion verschmelzen kann. Anstatt echtem `IO` wurde der Zustand der Welt durch einen simplen `Integer` beschrieben.

Der Code wurde nun um ein paar einfache Funktionen erweitert, um die Anwendung der Funktionen `nacheinander` und `komposition` zu demonstrieren.

Lösen Sie diese Aufgabe mit Papier & Bleistift, also ohne GHCI einfach nach der Lösung zu fragen!

a) Zu welchem Wert wertet der Ausdruck `demo` aus? Warum und wie?

b) Schreiben Sie eine Funktion `tick2 :: IO Int`, welchen den Zustand der Welt bei Aufruf von `tick2` zurückgibt und gleichzeitig den nachfolgenden Zustand der Welt mit 3 multipliziert.

```
import Prelude hiding (IO)

type Welt = Integer
type IO a = Welt -> (a,Welt)

nacheinander :: IO a -> IO b -> IO b
nacheinander f = komposition f . const

komposition  :: IO a -> (a -> IO b) -> IO b
komposition f g w1 = let (x,w2) = f w1 in g x w2 -- x::a, w2::Welt

-- Ab hier NEUER CODE:
demo = snd $ demoSequenz makeWorld

demoSequenz :: IO ()
demoSequenz = nacheinander tick $
              nacheinander tick $
              nacheinander tick $
              komposition  lese addiere

makeWorld :: Welt
makeWorld = 0

lese :: IO Int
lese w = (fromIntegral w,w)

tick :: IO ()
tick w = ((), succ w)

addiere :: Int -> IO ()
addiere x w = ((),w+(fromIntegral x))
```

H8-1 ***Typherleitung II*** (4 Punkte; Abgabeformat: Text oder PDF)
Erstellen Sie eine Typherleitung für folgendes Typurteil:

$$\{f{::}(\alpha \to \alpha) \to \beta \to \beta\} \vdash (\backslash x \texttt{ -> } f\ x)\ (\backslash y \texttt{ -> } y) :: \beta \to \beta$$

Sie dürfen sich aussuchen, ob Sie die Herleitung in Baum-Notation oder in linearer Notation verfassen.

H8-2 ***Unifikation II*** (4 Punkte; Abgabeformat: Text oder PDF)

a) Berechnen Sie den allgemeinsten Unifikator für folgende Menge von Typgleichungen:

$$\{\texttt{Int} \to \alpha = \beta \to \texttt{Bool} \to \beta, \gamma \to \texttt{Int} = \alpha\}$$

b) Gegeben ist folgende Menge von Typgleichungen:

$$\alpha \to \beta = \delta \to \gamma, \beta \to \delta = \theta \to \eta, \eta = \alpha$$

Geben Sie eine nicht-leere Substitution an, welche alle Variablen durch konkrete Typen ersetzt, aber welche kein Unifikator für die Gleichungsmenge ist.

c) Gegeben ist folgende Menge von Typgleichungen:

$$\eta \to \beta = \delta \to \theta, \alpha = \eta, \alpha \to \beta = \delta \to \gamma$$

Geben Sie einen Unifikator für die Gleichungsmenge an, welche nicht der allgemeinste Unifikator ist.

Abgabe: Lösungen zu den Hausaufgaben können bis Dienstag, den 23.06.2015, 11:00 Uhr mit UniworX abgegeben werden.

Aufgrund des Klausurbonus müssen die Hausaufgaben von Ihnen alleine gelöst werden. Abschreiben bei den Hausaufgaben gilt als Betrug und kann zum Ausschluss von der Klausur zur Vorlesung führen.