RHEINISCH-
WESTFÄLISCHE
TECHNISCHE
HOCHSCHULE
AACHEN

LEHR- UND FORSCHUNGSGEBIET
INFORMATIK 2
RWTH Aachen · D-52056 Aachen · GERMANY
http://www-i2.informatik.rwth-aachen.de/lufgi2

LuFG Informatik II

Prof. Dr. Jürgen Giesl
Peter Schneider-Kamp
Stephan Swiderski

Exercises *Functional Programming* – Sheet 11

Solutions will be collected until Friday, July 13, 2007 in the written exam. Exercises can be solved both in English and in German.

Important:

The written exam (Übungsscheinklausur) will take place on Friday, July 13, 10am in AH II instead of the last lecture.

To get a certificate for this course (Übungsschein) you must pass this test *and* reach at least 50% of the points on the exercise sheets. We strongly recommend the acquisition of this certificate, since this is a very good opportunity to prepare for the diploma or master examination.

Exercise 1 (2 + 2 points)

The constant function symbol `fix` can be translated into the following pure lambda term:

$$\overline{fix} = (\lambda x\, y.y\,(x\,x\,y))(\lambda x\, y.y\,(x\,x\,y))$$

For this translation, $\overline{fix}\ z \rightarrow_\beta^* z(\overline{fix}\ z)$ holds.

There exists another fixpoint combinator called Y with $Y = \lambda f.(\lambda x.f\,(x\,x))(\lambda x.f\,(x\,x))$. Please show:

(a) $Y\ z \not\rightarrow_\beta^* z(Y\ z)$

(b) $Y\ z \leftrightarrow_\beta^* z(Y\ z)$

Here, $\leftrightarrow_\beta$ is the union of $\rightarrow_\beta$ and $\leftarrow_\beta$.

In the next two exercises, please use the initial type assumption A_0 as presented in the lecture. This type assumption contains at least the following:

$$\begin{array}{lcl} A_0(\texttt{1}) & = & \texttt{Int} \\ A_0(\texttt{True}) & = & \texttt{Bool} \\ A_0(\texttt{False}) & = & \texttt{Bool} \\ A_0(\texttt{plus}) & = & \texttt{Int} \to \texttt{Int} \to \texttt{Int} \\ A_0(\texttt{Nil}) & = & \forall a.\ \texttt{List}\ a \\ A_0(\texttt{Cons}) & = & \forall a.\ a \to \texttt{List}\ a \to \texttt{List}\ a \\ A_0(\texttt{tuple}_2) & = & \forall a, b.\ a \to b \to (a, b) \\ A_0(\texttt{fix}) & = & \forall a.\ (a \to a) \to a \\ A_0(\texttt{isa}_{\texttt{True}}) & = & \texttt{Bool} \to \texttt{Bool} \\ A_0(\texttt{if}) & = & \forall a.\ \texttt{Bool} \to a \to a \to a \end{array}$$

Exercise 2 (6 points)

Give the most general shallow type schema for each term if it exists. For the remaining terms you should say why they cannot be typed using a shallow type schema.

(a) $\lambda x\ y.\ \texttt{if}\ x\ y$

(b) $\lambda x\ y.\ \texttt{tuple}_2\ x\ (\texttt{Cons}\ \texttt{Nil}\ y)$

(c) $\lambda x.\ \texttt{Cons}\ \texttt{if}\ (\texttt{Cons}\ \texttt{plus}\ x)$

(d) $\lambda x\ y.\ \texttt{plus}\ (x\ y)\ (y\ 1)$

Exercise 3 (3 + 3 + 4 points)

Use the type inference algorithm $\mathcal{W}$ to determine the most general type of the following λ-terms. Show the results of all sub-computations and unifications, too. If the term is not well-typed, show at what step and why the $\mathcal{W}$-algorithm detects this.

(a) $\lambda y.\ \texttt{if}\ \texttt{True}\ (y\ \texttt{True})\ (y\ \texttt{1})$

(b) $\texttt{tuple}_2\ (x\ \texttt{True})\ (x\ \texttt{1})$

(c) $\texttt{fix}\ (\lambda n\ x.\ \texttt{plus}\ \texttt{1}\ x)$