RWTH RHEINISCH-WESTFÄLISCHE TECHNISCHE HOCHSCHULE AACHEN

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

Prof. Dr. Jürgen Giesl
Darius Dlugosz

Exercises *Functional Programming* – Sheet 9

Solutions will be collected until Thursday, July 19, 2001.

Exercise 1 (4 points)

Translate the following Haskell-expressions into lambda terms and reduce them with WHNO-reduction using the $\rightarrow_{\beta\delta}$-relation, where δ is the set of rules from definition 3.3.5:

(a) `let not' = \x -> if x == True then False else True in not' True`

(b) `let half = \x -> if x <= 1 then 0 else half (minus x 1) in half 3`

(c) `let double = \x -> if x == 0 then 0 else (plus 2 (double (x - 1)))`
`in double 2`

(d) `let fact = \x -> if x <= 0 then 1 else (times (fact (x - 1)) x) in fact 2`

Hint: You can write `(minus x 1)` instead of `(x - 1)`, `(equal x 0)` instead of `x == 0`, etc.

Exercise 2 (2 points)

Give the representation of the constants $\texttt{isa}_{\texttt{Zero}}$, $\texttt{isa}_{\texttt{Succ}}$, $\texttt{isa}_{\texttt{True}}$, $\texttt{isa}_{\texttt{False}}$ in pure lambda calculus.

Exercise 3 (2 points)

Define the functions `plus` and `times` in pure lambda calculus.

Exercise 4 (2 points)

Show that $Y\,y \leftrightarrow_\beta^* y\,(Y\,y)$ holds, where Y denotes the term $\lambda f.(\lambda x.f(x\,x))(\lambda x.f(x\,x))$.

Exercise 5 (2 points)

The relation $\rightarrow_\eta \subseteq \Lambda \times \Lambda$ is the smallest relation with

(1) $\lambda x.t\,x \;\rightarrow_\eta\; t$, if $x \notin \text{free}(t)$

(2) if $t_1 \;\rightarrow_\eta\; t_2$, then also $(t_1\,r) \;\rightarrow_\eta\; (t_2\,r)$, $(r\,t_1) \;\rightarrow_\eta\; (r\,t_2)$ and $\lambda y.t_1 \;\rightarrow_\eta\; \lambda y.t_2$ for all $r \in \Lambda$, $y \in \mathcal{V}$.

Please answer the questions:

(a) What are the reasons for the requirement "*if* $x \notin \text{free}(t)$" in (1)?

(b) Why don't we use the η-reduction rule when implementing HASKELL using the lambda calculus?

Remark:
Definition 3.3.5 (δ-rules for HASKELL-programs) Let Con be the set of constructors of a HASKELL-program, where Con_n denotes the set of all constructors of arity n. Let δ_0 be the rules of the predefined operators in HASKELL, i.e., δ_0 contains rules such as $\texttt{plus}\,1\,2 \;\rightarrow\; 3$, $\texttt{not}\,\texttt{True} \;\rightarrow\; \texttt{False}$, etc. All these rules have the form $f\,t_1 \,\ldots\, t_n \;\rightarrow\; r$, where $f \in \mathcal{C}_0$ and $t_1, \ldots, t_n, r \in \mathsf{Con}_0$. Because there are not two rules with the same left-hand side, δ_0 is a delta-rule set. The entire set δ is defined as follows:

$$
\begin{array}{rcl}
\delta & = & \delta_0 \cup \\
 & & \{\texttt{bot} \;\rightarrow\; \texttt{bot}, \\
 & & \texttt{isa}_{()}\,\texttt{tuple}_0 \;\rightarrow\; \texttt{True}, \\
 & & \texttt{if}\,\texttt{True} \;\rightarrow\; \lambda xy.x, \\
 & & \texttt{if}\,\texttt{False} \;\rightarrow\; \lambda xy.y, \\
 & & \texttt{fix} \;\rightarrow\; \lambda f.f(\texttt{fix}f)\} \cup \\
 & & \{\texttt{isa}_{\underline{\mathsf{constr}}}\,(\underline{\mathsf{constr}}\,t_1 \,\ldots\, t_n) \;\rightarrow\; \texttt{True} \,|\, \underline{\mathsf{constr}} \in \mathsf{Con}_n,\, n \geq 0, \\
 & & \qquad\qquad t_1, \ldots, t_n \in \Lambda,\, t_j \text{ closed}\} \cup \\
 & & \{\texttt{isa}_{\underline{\mathsf{constr}}}\,(\underline{\mathsf{constr}'}\,t_1 \,\ldots\, t_m) \;\rightarrow\; \texttt{False} \,|\, \underline{\mathsf{constr}'} \in \mathsf{Con}_m,\, \underline{\mathsf{constr}} \neq \underline{\mathsf{constr}'},\, m \geq 0, \\
 & & \qquad\qquad t_1, \ldots, t_m \in \Lambda,\, t_j \text{ closed}\} \cup \\
 & & \{\texttt{argof}_{\underline{\mathsf{constr}}}\,(\underline{\mathsf{constr}}\,t_1 \,\ldots\, t_n) \;\rightarrow\; \texttt{tuple}_{\mathsf{n}}\,t_1 \,\ldots\, t_n \,|\, \underline{\mathsf{constr}} \in \mathsf{Con}_n,\, n \geq 0, \\
 & & \qquad\qquad t_1, \ldots, t_n \in \Lambda,\, t_j \text{ closed}\} \cup \\
 & & \{\texttt{sel}_{n,i}\,(\texttt{tuple}_n\,t_1 \,\ldots\, t_n) \;\rightarrow\; t_i \,|\, n \geq 2,\, i \leq n,\, t_1, \ldots, t_n \in \Lambda,\, t_j \text{ closed}\}
\end{array}
$$