RHEINISCH-
WESTFÄLISCHE
TECHNISCHE
HOCHSCHULE
AACHEN

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

Prof. Dr. Jürgen Giesl
Peter Schneider-Kamp

Exercises *Functional Programming* – Sheet 10

Solutions will be collected until Wednesday, Jul 06, 2005 in the exercise course.
Exercises can be solved both in English and in German.

Exercise 1 (2+2+2 points)

Identify the free variables of the following lambda terms and also apply the three substitutions $\sigma_1 = [x/\lambda x.x]$, $\sigma_2 = [y/y\,y]$, and $\sigma_3 = [z/\lambda z.x\,z]$ to each of these terms:

(a) $(\lambda x.y\,x)\,(\lambda y.x)$

(b) $\lambda x\,y.(\lambda z.z\,(\lambda x.y))\,z$

(c) $(\lambda x\,y.z\,(y\,z)\,x)\,(\lambda x.y\,(\lambda y.y))$

Exercise 2 (2 + 3 + 3 points)

Mark the sequences that correspond to leftmost-innermost reductions and leftmost-outermost reductions respectively.

(a) Give all possible reduction sequences with $\rightarrow_\beta$ starting with the following term:

$$(\lambda x\,y.\mathsf{plus}\,x\,y)\,((\lambda z.\mathsf{plus}\,3\,z)\,4)\,\mathsf{5}$$

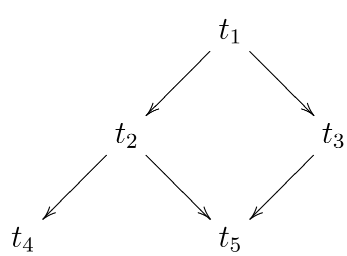
(b) Give the leftmost-outermost and the leftmost-innermost reduction sequences with $\rightarrow_\beta$ starting with the following term:

$$(\lambda g.g\,\mathsf{times}\,(g\,\mathsf{plus}\,(g\,\mathsf{times}\,\mathsf{3})))\,(\lambda f\,x.f\,x\,x)$$

(c) Give all reduction sequences with $\rightarrow_\beta$ with at most 5 steps starting with the following term:

$$(\lambda x\,y.y(x\,x\,y))(\lambda x\,y.y(x\,x\,y))(\lambda x.x)$$

Hint: You can save space by representing the reduction sequences as (directed) graphs. Example for $t_1 \rightarrow_\beta t_2 \rightarrow_\beta t_4$, $t_1 \rightarrow_\beta t_2 \rightarrow_\beta t_4$, and $t_1 \rightarrow_\beta t_3 \rightarrow_\beta t_5$.



Exercise 3 (2 points)

The notion of "δ-reduction" is defined as follows:

A set δ of rules of the form $c\,t_1 \dots t_n \;\to\; r$ with $c \in \mathcal{C}$, $t_1, \dots, t_n, r \in \Lambda$ is called a delta-rule set if

(1) $t_1, \dots, t_n, r$ are closed lambda terms

(2) all t_i are in $\to_\beta$-normal form

(3) the t_i do not contain any left-hand side of a rule from δ

(4) in δ there exist no two rules $c\,t_1 \dots t_n \;\to\; r$ and $c\,t_1 \dots t_m \;\to\; r'$ with $m \geq n$

For such a set δ we define the relation $\to_\delta$ as the least relation with

- $l \;\to_\delta\; r$ for all $l \to r \;\in\; \delta$
- if $t_1 \;\to_\delta\; t_2$, then also $(t_1\, r) \;\to_\delta\; (t_2\, r)$, $(r\, t_1) \;\to_\delta\; (r\, t_2)$ and $\lambda y.t_1 \;\to_\delta\; \lambda y.t_2$ for all $r \in \Lambda$, $y \in \mathcal{V}$.

We denote the combination of β- and δ-reduction by $\to_{\beta\delta}$, i.e., $\to_{\beta\delta} = \to_\beta \cup \to_\delta$.

The four conditions (1), (2), (3), (4) are required in order to ensure that $\to_{\beta\delta}$ is confluent. Would $\to_{\beta\delta}$ still be confluent if we replace condition (4) by the following condition:

> in δ there exist no two rules $c\,t_1 \dots t_n \;\to\; r$ and $c\,t_1 \dots t_m \;\to\; r'$
> with $m \geq n$ <u>and $r \neq r'$</u>

Please explain your answer.

Important:

> **Those of you who are interested in obtaining the exercise certificate should send me an e-mail (psk@informatik.rwth-aachen.de) until Wednesday, July 6, 11:00am. The e-mail is not a binding registration, but will only be used to see how many of you want to participate in the exercise exam. Further details will be discussed in the global exercise that Wednesday.**
>
> **Thanks, Peter.**