The University of Birmingham
School of Computer Science


# *Solutions to Exercise Sheet 1*

## Exercise 1: Introduction to the lambda calculus

a. *In what ways does Java's way of defining functions (i.e. methods) differ from that of the $\lambda$-calculus?*

There are several differences:

- A Javafunction has a name, a $\lambda$-calculusfunction not.
- Javafunctions and their arguments have types.
- In the $\lambda$-calculuswe also allow functions as arguments, in Java, we do not.

b. *Translate the following Java expressions into $\lambda$-calculus notation:*

i. `sin(x+3)`: $\sin(+\ x\ 3)$

ii. `length(y)+z`: $+\ (\mathit{length}\ y)\ z$

iii.
```
public static int quot(double x, double n)
          { return (int)(x/n); }
```
The translation is $\lambda x.\, \lambda n.\, \mathit{int}\ (/\ x\ n)$

## Exercise 2: Parsing lambda terms

*Draw the syntax trees for the following $\lambda$-terms:*

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

b. $\lambda x.\, \lambda y.\, \lambda z.\, xyz$

c. $(\lambda x.\, xx)(\lambda x.\, xx)$

## Exercise 3: beta reduction

a. *Consider the* $\lambda$*-term*$(\lambda f.\,\lambda x.\,f(fx))\;(\lambda y.\,*\;y\;2)\;5$*.* $\beta$*-reduce the term as far as possible.*

$$\begin{array}{lll} (\lambda f.\,\lambda x.\,f(fx))\;(\lambda y.\,*\;y\;2)\;5 & & \\ \longrightarrow_\beta & (\lambda x.\,(\lambda y.\,*\;y\;2)\,((\lambda y.\,*\;y\;2)\,x))\;5 & \text{by applying the } \lambda f \text{ function to its argument} \\ \longrightarrow_\beta & (\lambda y.\,*\;y\;2)\,((\lambda y.\,*\;y\;2)\,5) & \text{by applying the } \lambda x \text{ function to its argument} \\ \longrightarrow_\beta & (\lambda y.\,*\;y\;2)\,(*\;5\;2) & \text{by applying the second } \lambda y \text{ function to 5} \\ \longrightarrow_\beta & (*\,(*\;5\;2)\;2) & \text{by applying the } \lambda y \text{ function to its argument} \end{array}$$

b. *Let* $S$ *be the term* $\lambda x.\,\lambda y.\,\lambda z.\,(xz)(yz)$ *and* $K$ *the term* $\lambda x.\,\lambda y.\,x$*. Reduce* $SKK$ *to normal form.*

$$\begin{array}{ll} (\lambda x.\,\lambda y.\,\lambda z.\,(xz)\;(yz))\;K\;K & \\ \longrightarrow_\beta & (\lambda y.\,\lambda z.\,(Kz)\;(yz))\;K \\ \longrightarrow_\beta & \lambda z.\,(Kz)(Kz) = \lambda z.\,((\lambda x.\,\lambda y.\,x)\;z)\;(Kz) \\ \longrightarrow_\beta & \lambda z.\,(\lambda y.\,z)\;(Kz) \\ \longrightarrow_\beta & \lambda z.\,z \end{array}$$

c. *Consider the following lambda terms:*

$$\begin{array}{rcl} M & = & \lambda x.\,((\lambda z.\,z\;x)\;((\lambda r.\,\lambda s.\,s\;r)\;y\;f)) \\ N & = & \lambda x.\,((\lambda z.\,\lambda u.\,\lambda v.\,u\;v\;z)\;x\;f\;y) \end{array}$$

*Use* $\beta$*-reduction to show that* $M$ *and* $N$ *are* $\beta$*-equivalent.*

$$\begin{array}{lll} M = \lambda x.\,((\lambda z.\,z\;x)\;((\lambda r.\,\lambda s.\,s\;r)\;y\;f)) & & \\ \longrightarrow_\beta & \lambda x.\,((\lambda z.\,z\;x)\;(f\;y)) & \text{by applying the } \lambda r \text{ and } \lambda s \text{ abstraction terms} \\ \longrightarrow_\beta & \lambda x.\,(f\;y)\;x & \\ N = \lambda x.\,((\lambda z.\,\lambda u.\,\lambda v.\,u\;v\;z)\;x\;f\;y) & & \\ \longrightarrow_\beta & \lambda x.\,f\;y\;x & \text{by applying the } \lambda z,\ \lambda u \text{ and } \lambda v \text{ abstraction terms} \end{array}$$

Since $M$ and $N$ are $\beta$-equivalent to the same normal form, they are $\beta$-equivalent themselves.