University of Innsbruck | **Institute of Computer Science**
Test 1B | **November 26, 2010**

Functional Programming | **WS 2010/2011** | **LVA 703018**

Solutions

This test consists of three exercises. *Explain your answers*. The available points for each item are written in the margin.

(8) **1.** Given the Haskell function

```
foldl f b []     = b
foldl f b (x:xs) = foldl f (f b x) xs
```

use *equational reasoning* to evaluate the function call `foldl (*) 1 [4,3,2,1]` and give at least four intermediate steps.

Solution.

```
foldl (*) 1 [4,3,2,1]  = foldl (*) (1 * 4) [3,2,1]
                       = foldl (*) ((1 * 4) * 3) [2,1]
                       = foldl (*) (((1 * 4) * 3) * 2) [1]
                       = foldl (*) ((((1 * 4) * 3) * 2) * 1) []
                       = ((((1 * 4) * 3) * 2) * 1)
                       = 24
```

(8) **2.** Implement a function `dup : [a] -> [(a,a)]`, duplicating every element of a list in a pair, e.g.,

```
dup [2,3,1,5] = [(2,2),(3,3),(1,1),(5,5)]
```

It is not allowed to use any predefined functions.

Solution.

```
dup xs = map (\x -> (x,x)) xs
  where map f []     = []
        map f (x:xs) = f x : map f xs
```

3. Consider the lambda-term $t = (\lambda xy.\, x)\ ((\lambda x.\, y)\ z)$.

(5) (a) Reduce t to normal form by applying the beta-rule repeatedly.

Solution.

$$\begin{aligned} &\underline{(\lambda xy.\, x)\ ((\lambda x.\, y)\ z)} \\ &\to_\beta \lambda y_1.\, \underline{(\lambda x.\, y)\ z} \\ &\to_\beta \lambda y_1.\, y \end{aligned}$$

(4) (b) Give the three sets $\mathcal{FV}\mathsf{ar}(t)$, $\mathcal{BV}\mathsf{ar}(t)$, and $\mathcal{V}\mathsf{ar}(t)$ of the (free/bound) variables of t.

Solution.

$$\begin{aligned} \mathcal{FV}\mathsf{ar}(t) &= \{y, z\} \\ \mathcal{BV}\mathsf{ar}(t) &= \{x, y\} \\ \mathcal{V}\mathsf{ar}(t) &= \{x, y, z\} \end{aligned}$$