universität innsbruck

Functional Programming	WS 2017/2018	LVA 703024
EXAM 1		February 2, 2018

This exam consists of **four** exercises. The available points for each item are written in the margin. You need at least 50 points to pass.

[25] **1** Consider the two Haskell functions:

```
[] ++ ys      =  ys
(x:xs) ++ ys  =  x : (xs ++ ys)

concat []     =  []
concat (x:xs) =  x ++ concat xs
```

Use structural induction to prove that `concat` $(xs$ `++` $ys) =$ `concat` xs `++` `concat` ys for all lists xs and ys. Apart from the defining equations above, you may assume associativity of `++`.

[6] (a) Prove the base case, applying a single equation at a time.

[9] (b) State the induction hypothesis and the statement you have to show in the step case.

[10] (c) Prove the step case, applying a single equation at a time.

[25] **2** Consider the four λ-terms $A = \lambda f x.\, f\ x$, $K = \lambda x y.\, x$, $\Omega = (\lambda x.\, x\ x)\ (\lambda x.\, x\ x)$, and y.

[9] (a) Use normal order reduction to reduce $A\ K\ y\ \Omega$ to normal form, giving each intermediate β-step.

[8] (b) State which of A, K, Ω, and y from above are in weak head normal form and which are not.

[8] (c) Determine for each of the following λ-terms if they are α-equivalent to A, K, Ω, and y, respectively.

$$t_1 = y \qquad t_2 = \lambda x y.\, y \qquad t_3 = \lambda g y.\, g\ y \qquad t_4 = \lambda y x.\, y$$

[25] **3** Consider the Haskell functions:

```
diffs []     = 0
diffs (x:xs) = diffs xs - x

rev acc [] = acc
rev acc (x:xs) = rev (x:acc) xs

downto m n = if m < n then [] else m : downto (m - 1) n
```

[9] (a) Use equational reasoning to evaluate `diffs [1,2]`, `rev [1,2,3] [1,2,3]`, and `downto 1 2`.

[6] (b) Determine for each of the above functions whether it is tail recursive and/or guardedly recursive.

[10] (c) Give a tail recursive implementation of `downto`.

[25] **4** Consider the typing environment $E = \{1 :: \mathsf{Int}\}$.

[10] (a) Use type checking to prove the typing judgment $E \vdash (\lambda x y.\, 1) :: \mathsf{Int} \to \mathsf{Int} \to \mathsf{Int}$.

[6] (b) Apply the typing constraint rules to transform $E \rhd \lambda x y.\, 1 :: \alpha_0$ into a unification problem.

[9] (c) Solve the unification problem $\alpha_0 \approx \alpha_1 \to \alpha_2; \alpha_2 \approx \alpha_3 \to \alpha_4; \mathsf{Int} \approx \alpha_4$ and compute the resulting mgu.