universität innsbruck

Functional Programming	WS 2018/2019	LVA 703024
EXAM 2		March 1, 2019

> 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 Haskell functions:

```
map :: (a -> b) -> [a] -> [b]
map f []     = []
map f (x:xs) = f x : map f xs

takeChain :: (a -> a -> Bool) -> [a] -> [a]
takeChain p (x:y:ys) | p x y     = x : takeChain p (y:ys)
                     | otherwise = [x]
takeChain p xs = xs
```

[9] (a) Use equational reasoning to evaluate `takeChain (<) [1,2,3,1]` and `map (\x -> 0) [1,2,3,1]`, giving each intermediate step.

[6] (b) Determine whether `map` and `takeChain` are tail recursive and guardedly recursive, respectively.

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

[25] **2** Consider `map` from Exercise 1. Use structural induction to prove `map` $(\backslash x$ `->` $x)\ xs = xs$ for all finite lists xs.

[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] **3** Consider the λ-term $t = ((\lambda x.\, x)\ (\lambda x.\, (\lambda y.\, y)))\ (((\lambda x.\, x)\ (\lambda x.\, (\lambda y.\, y)))\ a)$

[4] (a) Write t as compactly as possible, using the notational conventions (for omitting parentheses etc.).

[10] (b) Stepwise reduce t to normal form using applicative order reduction.

[6] (c) Only using the variables x and y, give three *different* λ-terms that are α-equivalent to $(\lambda xy.\, y)\ y$.

[5] (d) Give a λ-term that has a β-normal form but also admits an infinite β-reduction.

[25] **4** Consider the typing environment $E = P \cup \{f :: \mathsf{Int} \to \mathsf{Int}, x :: \mathsf{Int}\}$.

[9] (a) Use type checking to show that $E \vdash f\ (x + 1) :: \mathsf{Int}$.

[8] (b) If possible, solve the unification problem $\alpha_0 \to \mathsf{List}(\alpha_1) \approx \alpha_1 \to \alpha_0$ and compute the resulting mgu.

[8] (c) Apply the typing constraint rules to transform $E \triangleright f\ f :: \alpha_0$ into a unification problem.