universität innsbruck

| Functional Programming | WS 2017/2018 | LVA 703024 |
|---|---|---|
| EXAM 2 | | March 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:

```
length []     = 0
length (x:xs) = 1 + length xs

map f []     = []
map f (x:xs) = f x : map f xs
```

Use structural induction to prove that `length (map` $f$ $xs$`) = length` $xs$ for all lists $xs$.

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

[10] (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 $\lambda$-terms $t_1 = \lambda x.\, x\ x$, $t_2 = \lambda x.\, x\ y$, $t_3 = \lambda y.\, y\ x$, and $t_4 = \lambda y.\, y\ y$.

[9] (a) Use normal order reduction to reduce $t_1\ t_2\ t_3\ t_4$ to normal form, giving each intermediate $\beta$-step.

[8] (b) Determine for $1 \leqslant i \leqslant 4$ if $t_i\ t_i$ has a normal form or not.

[8] (c) Determine for each pair of terms from $\{t_1, \ldots, t_4\}$ if they are $\alpha$-equivalent or not.

[25] **3** Consider the Haskell function `map` from Exercise 1.

[10] (a) Use equational reasoning to evaluate `map ((-) 1) [1,2]`.

[5] (b) Explain why `map` is not tail recursive and why it is guardedly recursive or not.

[10] (c) Give a tail recursive implementation of `map`. Attention: Do not use any function that is not tail recursive.

[25] **4** Consider core FP.

[5] (a) Find a core FP expression that is not typeable with respect to the primitive environment $P$.

[10] (b) Consider the typing environment $E = \{z :: \mathsf{Int}\}$. Apply the typing constraint rules to transform $E \rhd z\ (\lambda x.\, x) :: \alpha_0$ into a unification problem.

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