University of Innsbruck **Institute of Computer Science**
Test 2B **January 15, 2010**

Functional Programming	**WS 2009/2010**	**LVA 703018**

Name: **Matr.Nr.:**

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

[8] **1.** Reduce the λ-term

$$(\lambda p.p\ (\lambda x\ y.y))\ ((\lambda x\ y\ f.f\ x\ y)\ \mathsf{Y}\ (\lambda y.y))$$

to normal form using the leftmost outermost reduction strategy. Here, Y denotes the λ-term $\lambda f.(\lambda x.f\ (x\ x))\ (\lambda x.f\ (x\ x))$.

[8] **2.** Consider the OCaml functions

```
let rec (@) xs ys = match xs with []     -> ys
                                | x::xs -> x::(xs@ys)

let rec rev = function []     -> []
                     | x::xs -> rev xs @ [x]

let rec len = function []     -> 0
                     | _::xs -> 1 + len xs
```

Prove by induction that

$$\texttt{len}(\texttt{rev}\ xs) = \texttt{len}\ xs$$

for all lists xs. You may use the equation

$$\texttt{len}(xs\ \texttt{@}\ ys) = \texttt{len}\ xs + \texttt{len}\ ys \qquad (\star)$$

[5] **3.** (a) Consider the type inference problem $E \rhd \mathsf{fst}\ (\mathsf{pair}\ 0\ 0) : \alpha_4$ using the typing environment $E = \{\mathsf{fst} : \mathsf{pair}(\alpha_0, \alpha_1) \to \alpha_0, \mathsf{pair} : \alpha_2 \to \alpha_3 \to \mathsf{pair}(\alpha_2, \alpha_3), 0 : \mathsf{int}\}$. Apply the type inference rules of $\mathcal{I}$, to compute a resulting unification problem. Give all the intermediate computation steps.

[4] (b) Solve the following unification problem.

$$\begin{aligned} \mathsf{pair}(\alpha_0, \alpha_1) \to \alpha_0 &\approx \alpha_5 \to \alpha_4 \\ \mathsf{int} &\approx \alpha_6 \\ \alpha_2 \to \alpha_3 \to \mathsf{pair}(\alpha_2, \alpha_3) &\approx \alpha_7 \to \alpha_6 \to \alpha_5 \\ \mathsf{int} &\approx \alpha_7 \end{aligned}$$