University of Innsbruck — Institute of Computer Science
Test 2B — 11 January 2008

Functional Programming — WS 2007/2008 — LVA 703018

Solutions

1.

$$
\begin{aligned}
(\lambda uv.u)\ (\underline{(\lambda w.w)\ (\lambda xy.y)})\ (\lambda z.z) &\to_\beta \underline{(\lambda uv.u)\ (\lambda xy.y)}\ (\lambda z.z)\\
&\to_\beta \underline{(\lambda vxy.y)\ (\lambda z.z)}\\
&\to_\beta \lambda xy.y
\end{aligned}
$$

2. *Proof.* The property to prove is

$$P(xs) = (\mathsf{prod}\ xs = \mathsf{fold}\ (\times)\ 1\ xs).$$

Base Case ($\mathsf{xs} = []$). To show: $P([])$, i.e.,

$$\mathsf{prod}\ [] = \mathsf{fold}\ (\times)\ 1\ [].$$

Starting from the lhs, following derivation concludes the base case:

$$
\begin{aligned}
\mathsf{prod}\ [] &= 1 && \text{(definition of prod)}\\
&= \mathsf{fold}\ (\times)\ 1\ [] && \text{(definition of fold)}
\end{aligned}
$$

Step Case ($\mathsf{xs} = y :: ys$). To show: $P(ys) \to P(y :: ys)$. Therefor assume the IH $P(ys)$, i.e.,

$$\mathsf{prod}\ ys = \mathsf{fold}\ (\times)\ 1\ ys.$$

It remains to show that $P(y :: ys)$. This is done as follows:

$$
\begin{aligned}
\mathsf{prod}\ (y :: ys) &= y \times \mathsf{prod}\ ys && \text{(definition of prod)}\\
&= y \times \mathsf{fold}\ (\times)\ 1\ ys && \text{(IH)}\\
&= \mathsf{fold}\ (\times)\ 1\ (y :: ys) && \text{(definition of fold)}
\end{aligned}
$$

□

3. (a) Yes, e and o are tail recursive since the last function calls in their respective recursive cases are to each other.

(b) No, prod is not tail recursive since the last function call in the recursive case of its definition is to (*). Here is a tail recursive alternative:

```
let prod xs =
 let rec prod acc = function
  | [] -> acc
  | x :: xs -> prod (x * acc) xs
 in prod 1 xs
;;
```