HW 8 **Prof. Caldwell**
Due: 20 September 2011 **COSC 3015**

Recall the list induction principle to prove a property for finite lists of type a.

$$\begin{array}{l}[P([\,])\wedge \\ \;\forall x :: a.\, \forall xs :: [a].\, P(xs) \Rightarrow P(x : xs) \\ \;\;\Rightarrow \forall ys :: [a].P(ys)\end{array}$$

Thus, for a property P of lists, to show that $\forall ys :: [a].\ P(ys)$ it is enough to show two things:

$$\begin{array}{rl}\textbf{i.)} & P([\,]) \\ \textbf{ii.)} & \forall x :: a.\, \forall xs :: [a].\, P(xs) \Rightarrow P(x : xs)\end{array}$$

Here are some definitions (note that $\perp$ means loop forever).

$$\begin{array}{l}
head(x : xs) = x \\
head[\,] = \perp \\
\\
last[x] = x \\
last(x : xs) = last\ xs \\
last[\,] = \perp \\
\\
reverse[\,] = [\,] \\
reverse(x : xs) = (reverse\ xs) + +[x] \\
\\
map\ f\,[\,] = [\,] \\
map\ f\ (x : xs) = (f\ x)\ :\ map\ f\ xs \\
\\
(f\,.\,g)\ x = f\ (g\ x)
\end{array}$$

Two useful lemmas for problem 3 and 4 are as follows:

Lemma 1. $\forall ys, xs :: [a].\ xs \neq [] \Rightarrow last(ys + +xs) = last\ xs$
Lemma 2. $\forall ys, xs :: [a].\ xs \neq [] \Rightarrow head(xs + +ys) = head\ xs$

One proof technique you also might need for 3 or 4 is case analysis. For any list $xs :: [a]$ you can say $xs = []$ or $xs = y : ys$ for some arbitrary $y : a$ and $ys :: [a]$. This can be captured in a lemma as follws:

Lemma 3. $\forall xs :: [a].\ xs = [] \vee \exists y :: a, ys :: [a].\ xs = (y : ys)$

Prove the following by finite list induction [1].

1.) $\forall xs :: [a].\ map\,(\backslash x \rightarrow x)\ xs = xs$
2.) $\forall xs :: [a].\ map\,(f\,.\,g)\ xs = ((map\ f)\,.\,(map\ g))\ xs$
3.) $\forall xs :: [a].\ head\,(reverse\ xs) = last\ xs$
4.) $\forall xs :: [a].\ last\,(reverse\ xs) = head\ xs$

[1] For problem 2 you can assume $g :: a \rightarrow b$ and $f :: b \rightarrow c$.