Optimising Typed Programs – Exercises

Note: There is an error in the notes; you need to replace the requirement $\vec{\alpha} = \mathrm{ftv}(\tau) \setminus \mathrm{ftv}(\Gamma)$ in rules (6) and (9) with the requirement $\mathrm{ftv}(\vec{\alpha}) \cap \mathrm{ftv}(\Gamma) = \emptyset$. Otherwise, Lemma 2.1 does not hold.

Exercise 1 Give an example showing that simultaneous substitution is different from compositional substitution, i.e., show that there exist types τ_1, τ_2 and τ, such that

$$\tau\{\alpha_1 \mapsto \tau_1, \alpha_2 \mapsto \tau_2\} \neq (\tau\{\alpha_1 \mapsto \tau_1\})\{\alpha_2 \mapsto \tau_2\}$$

□

Exercise 2 Prove Lemma 2.1. □

Exercise 3 Show that $\longrightarrow_{\mathrm{proj}}$ is type preserving. □

Exercise 4 Show that $\longrightarrow_{\mathrm{dce}}$ is type preserving. □

Exercise 5 Assume that the typed lambda language is extended to support integers. What is the result of applying $\longrightarrow_{\mathrm{spec2}}$ and $\longrightarrow_{\mathrm{inl1}}$ to the expression

$$
\begin{array}{l}
\texttt{letrec}\ \mathit{adder} : (\texttt{int} \to \texttt{int}) \to (\texttt{int} \to \texttt{int})\ g\ \texttt{=} \\
\quad \lambda n : \texttt{int}.\ \texttt{if}\ n\ =\ 0\ \texttt{then}\ 0 \\
\qquad\qquad\quad \texttt{else}\ g\ n\ +\ \mathit{adder}\ g\ (n\ -\ 1) \\
\texttt{in}\ \mathit{adder}\ (\lambda x : \texttt{int}.\ x\ +\ 1)\ 10
\end{array}
$$

□

Exercise 6 What is the result of applying value propagation to the expression

$$
\begin{array}{l}
\lambda y : \texttt{bool}.\ \lambda x : \texttt{bool}. \\
\quad \texttt{if}\ x\ \texttt{then} \\
\qquad \texttt{if}\ x\ \texttt{then}\ y\ \texttt{else}\ \texttt{true} \\
\quad \texttt{else}\ y
\end{array}
$$

□