Sample solution for Pierce 11.4.1(1).

We show that ascription can be formulated as a derived form by proving a theorem with the same structure as Thm. 11.3.1 (with errata corrected).

We write $\lambda^E$ for the simply-typed $\lambda$-calculus with the additional rules given in Figure 11-3, and $\lambda^I$ for pure simply-typed $\lambda$-calculus as given in Figure 9-1. We describe an elaboration function $e \in \lambda^E \rightarrow \lambda^I$, as follows:

$$
e(t \text{ as } T) = (\lambda x:T.x) \ e(t)$$
$$e(x) = x$$
$$e(\lambda x:T.t) = \lambda x:T.\ e(t)$$
$$e(t \ u) = e(t) \ e(u)$$

For convenience we will write $[t]$ for $e(t)$.

**Theorem 1** For each term $t$ of $\lambda^E$,

i. If $t \rightarrow_E t'$, then $[t] \rightarrow_I [t']$.

ii. If $[t] \rightarrow_I u$, then $\exists$ a $\lambda^E$ term $t'$ such that $t \rightarrow_E t'$ and $[t'] = u$.

iii. If $\Gamma \vdash^E t : T$, then $\Gamma \vdash^I [t] : T$.

iv. If $\Gamma \vdash^I [t] : T$, then $\Gamma \vdash^E t : T$.

**Proof.**

The proof of each result is by structural induction on $t$. In each instance, we consider only the interesting case where $t = t_1 \text{ as } T$, so $[t] = (\lambda x:T.x) \ [t_1]$. A really careful proof would consider all the other cases too.

i. By hypothesis, $t \rightarrow_E t'$. There are two cases, corresponding to the one-step rules that might apply to $t$:

- (E-ASCRIBE) In this case, $t_1 = v_1$ for some value $v_1$, and $t' = v_1$. So $[t] = (\lambda x:T.x)([v_1]) = (\lambda x:T.x)(v_1)$. By (E-APPABS), $(\lambda x:T.x)(v_1) \rightarrow_I v_1 = [v_1] = [t']$.

- (E-ASCRIBE1) In this case, $t_1 \rightarrow_E t'_1$. We have $[t] = (\lambda x:T.x)([t_1])$. By induction, $[t_1] \rightarrow_E [t'_1]$, so by (E-APP2), $(\lambda x:T.x)([t_1]) \rightarrow_I (\lambda x:T.x)([t'_1]) = [t']$.

ii. By hypothesis, $\exists u$ such that $[t] \rightarrow_I u$. There are two cases, corresponding to the one-step rules that might apply to $[t]$:

- (E-APPABS) In this case, $[t_1] = v_1$ for some value $v_1$, and $u = v_1$. It is easy to see that $t_1 = v_1$ as well, since only a variable can elaborate to a variable. Hence, by (E-ASCRIBE), we have $t \rightarrow_E v_1$ and $[v_1] = u$. 

1
• (E-App2) In this case, \( u = (\lambda x:T.x) \ u_1 \), where \([t_1] \rightarrow_I u_1\). By induction, \( \exists t'_1 \) such that \( t_1 \rightarrow_E t'_1 \) and \([t'_1] = u_1\). Hence, by (E-Ascribe1), we have \( t \rightarrow_E t'_1 \) as \( T \) and \([t'_1 as T] = (\lambda x:T.x) \ [t'_1] = (\lambda x:T.x) \ u_1 = u\).

iii. By hypothesis, \( \Gamma \vdash^E t : T \). The only applicable rule is (T-Ascribe), so we must have \( \Gamma \vdash t_1 : T \). By induction, \( \Gamma \vdash^I [t_1] : T \). So we can build the following deduction:

\[
\begin{align*}
\Gamma \vdash^I (\lambda x:T.x) : T & \rightarrow T & \text{T-Abs} \\
\Gamma \vdash^I [t_1] : T & \quad \text{T-App} \\
\Gamma \vdash^I (\lambda x:T.x) \ [x_1] : T
\end{align*}
\]

iv. By hypothesis, \( \Gamma \vdash^I [t] : T \). The only applicable rule is (T-App), which gives \( \Gamma \vdash^I (\lambda x:T.x) : T' \rightarrow T \) and \( \Gamma \vdash^I [t_1] : T' \). By the Inversion Lemma, \( T' = T \). By induction, \( \Gamma \vdash^E t_1 : T \). Finally, by (T-Ascribe), \( \Gamma \vdash^E T_1 \ as \ T : T \). \( \square \)