Errata for the *Semantics of Programming Languages* Notes

Peter Sewell

Time-stamp: <2003-05-19 15:08:54 pes20>

The updated notes available on the course web page includes all these corrections, together with other improvements.

1. Slide 51, *Executing L1 in mosml*, omitted a definition of \texttt{skip = ()}. It should read:

   \begin{verbatim}
   let val skip = ()
   and l1 = ref n1
   and l2 = ref n2
   \ldots
   and lk = ref nk
   in
   e
   end;
   \end{verbatim}

   where \( s \) is the store \( \{ l_1 \mapsto n_1, \ldots, l_k \mapsto n_k \} \) and all locations that occur in \( e \) are contained in \( \{ l_1, \ldots, l_k \} \).

   (watch out for \( \sim 1 \) and \( -1 \)).

2. The printing process misaligned the \( \leftrightarrow \) arrows used for writing partial maps. They should be rendered with the short vertical at the left end of the arrow. Negated turnstiles \( \nrightarrow \) were also misprinted; they should render as a turnstile \( \rightarrow \) with a slash through it.

3. The \texttt{l1} code omitted a clause for \texttt{if}.

4. On Slide 92, the second clause of the definition of substitution should read

   \begin{verbatim}
   \{e/z\}(\texttt{fn }x:T \Rightarrow e_1) = \texttt{fn }x:T \Rightarrow (\{e/z\}e_1)
   \text{ if } x \neq z \text{ (*)}
   \text{ and } x \notin \text{fv}(e) \text{ (*)}
   \end{verbatim}

   if (*) is not true, we first have to pick an alpha-variant of \texttt{fn }x:T \Rightarrow e_1 to make it so (always can)

5. On Page 13, the two \texttt{N} should be \texttt{Z}.

6. On Slide 112, the last expression should be \texttt{let val rec }f:T_1 \rightarrow T_2 = \texttt{fn }x:T_1 \Rightarrow e_1 \texttt{ in } e_2 \texttt{ end}.

7. On Page 73, two \texttt{()} should be \texttt{skip}.

8. One of the L3 rules for records was omitted – add:

   \begin{verbatim}
   (record3) \quad \langle e, s \rangle \rightarrow \langle e', s' \rangle
   \frac{\langle \#lab_i e, s \rangle \rightarrow \langle \#lab_i e', s' \rangle}{\langle e, s \rangle \rightarrow \langle e', s' \rangle}
   \end{verbatim}

   (a) (an errata erratum) That rule has \( e' \) not \( e \) on the right of the conclusion

9. One of the L3 rules for references was omitted – add:

   \begin{verbatim}
   (loc) \quad \Gamma(\ell) = T \texttt{ ref}
   \frac{\Gamma \vdash \ell: T \texttt{ ref}}{\Gamma \vdash \ell: T \texttt{ ref}}
   \end{verbatim}

10. The statement of store typing for L3 was missing a \texttt{ref}; it should be

    \( \Gamma \vdash s \text{ if } \forall \ell \in \text{dom}(s). \exists T. \Gamma(\ell) = T \texttt{ ref} \wedge \Gamma \vdash s(\ell): T. \)

11. In the proof of Determinacy, the start of the \( e_1 \texttt{ op } e_2 \) case had a bogus \( l := \). It should read:

    Case \( e_1 \texttt{ op } e_2 \). Suppose \( \Phi(e_1) \) and \( \Phi(e_2) \).

    Take arbitrary \( s, e', s', e'', s'' \) such that \( \langle e_1 \texttt{ op } e_2, s \rangle \rightarrow \langle e', s' \rangle \wedge \langle e_1 \texttt{ op } e_2, s \rangle \rightarrow \langle e'', s'' \rangle \).
12. The precise definition of the Ordered 2PL discipline should be as below

Now can make the Ordered 2PL Discipline precise

Say $e$ obeys the discipline if for any (finite or infinite)

$e \xrightarrow{a_1} e_1 \xrightarrow{a_2} e_2 \xrightarrow{a_3} ...$

- if $a_i$ is $(l_j := n)$ or $(l_j = n)$ then for some $k < i$ we have $a_k = \text{lock } m_j$ without an intervening unlock $m_j$.

- for each $j$, the subsequence of $a_1, a_2, ...$ with labels lock $m_j$ and unlock $m_j$ is a prefix of $((\text{lock } m_j)(\text{unlock } m_j))^*$. Moreover, if $\neg(e_k \xrightarrow{a} )$ then the subsequence does not end in a lock $m_j$.

- if $a_i = \text{lock } m_j$ and $a_{i'} = \text{unlock } m_{j'}$ then $i < i'$

- if $a_i = \text{lock } m_j$ and $a_{i'} = \text{lock } m_{j'}$ and $i < i'$ then $j < j'$

13. The precise definition of deadlock-freedom should have an $s'$ instead of an $s$ in the final state.