Errata in *Programming Language Foundations*

Aaron Stump  
Computer Science  
The University of Iowa

October 14, 2014

**Corrections below**

Despite many previous readings, attentive readers have found some errors in the book. I correct these below, sometimes underlining changes to the text. Thanks to the following people for reporting bugs: Ryan Brummet, Junyang Chen, Wyatt Kaiser, Tingting Liu, Talal Riaz, John Bodeen.

**Chapter 1**

1. page 28, second to last case of the proof. The sentence should say “So by the induction hypothesis, we have that $[t_1]_{\sigma'}$ and $[t_2]_{\sigma'}$ are both defined and equal to $[t_1]_{\sigma}$ and $[t_2]_{\sigma}$, respectively.”

2. page 30, Section 1.14.2, problem 2. The theorem cannot be proved without some additional assumption. So the revised problem is:

   2. Let us temporarily define $\sigma \leq \sigma'$ for assignments $\sigma$ and $\sigma'$ to mean that for all variables $x$, if $\sigma(x)$ is defined then so is $\sigma'(x)$ and we have $\sigma(x) \leq \sigma'(x)$. Suppose that $t$ is a term which does not contain the negation or subtraction symbols, and suppose that $\sigma$ is an assignment where $0 \leq \sigma(x)$ for every variable $x$. Prove by induction on the structure of $t$ that if $\sigma \leq \sigma'$, then $0 \leq [t]_{\sigma} \leq [t]_{\sigma'}$. (You can use the proof of Theorem 1.11.1 as a guide.)

**Chapter 2**

1. page 35, top of the page: it says we can “easily define the syntax for all commands except while-commands”, but it should say “easily define the semantics for all commands except while-commands”.

2. page 45, proof of Theorem 2.5.4. In the proof, I wrote “$f(c(n)) \sqsubseteq f(\sqcup c)$”. It should say “$f(c(n)) \sqsubseteq f(\sqcup c)$” instead (we are not using the $\sqsubseteq$ symbol in this chapter).

**Chapter 3**

1. page 92, start of Section 3.8. It should say “This means that the strongest formula is *False* and the weakest is *True*.” (The book has *False* and *True* reversed.)
Chapter 4

1. page 100. The multi-step derivation at the bottom of the page has a bug in the left branch (a proof rule is not being correctly applied). The correct derivation is:

\[
\begin{align*}
    x &:= 1; \sigma \leadsto \sigma[x \mapsto 1] \\
    x &:= 1; y := 2; \sigma \leadsto \sigma[x \mapsto 1] \\
    y &:= 2; \sigma[x \mapsto 1] \leadsto \sigma[x \mapsto 1, y \mapsto 2] \\
    x &:= 1; y := 2; \sigma \leadsto \star \sigma[x \mapsto 1, y \mapsto 2] \\
\end{align*}
\]

2. page 120, part 1 of problem 4.5.1. The final state should be \(\{ x \mapsto -10, y \mapsto 20, z \mapsto -1 \}\), not \(\sigma[z \mapsto -1]\).

3. page 120, part 1 of problem 4.5.2. The problem should be asking for a reduction sequence (using the small-step operational semantics), not a single small-step derivation. So the problem can be changed to:

Write a sequence of reduction steps (with the small-step semantics) which show how to reduce the following command and starting state to the final state \(\{ x \mapsto 90 \}\). You do not need to give derivations proving your individual small steps.

\[
\text{if } x < 100 \text{ then } x := x * 10 \text{ else skip, } \{ x \mapsto 9 \}\n\]

4. page 120, part 2 of problem 4.5.2. The final state should be \(\{ x \mapsto 14, y \mapsto 3 \}\), not \(\{ x \mapsto 14, y \mapsto 1 \}\).

5. page 121, part 4 of problem 4.5.2. The problem asks you to find a command \(c'\) making a particular small-step judgment is provable, but that judgment mentions \(c\) rather than \(c'\). It should mention \(c'\).

Chapter 8

1. page 218, example command using \texttt{await}: to get correct behavior, the command should await \(y + y' = 0\), rather than \(y * y' = 0\). Also, we should initialize \(z\) and \(z'\) to 1 before the concurrently executing commands begin. So change the definition of \(exp_{z,y,n}\) to be

\[
exp_{z,y,n} = (\texttt{while } y > 0 \texttt{ do } y := y - 1; z := z + n)
\]

and then use this for the command:

\[
\begin{align*}
    y &:= x; y' := x; z := 1; z' := 1; \\
    (exp_{z,y,1} || exp_{z',y',1} || \texttt{await } y + y' = 0 \texttt{ then } z := z + z')
\end{align*}
\]

2. page 221, Figure 8.8. The third rule in the figure (the one on the right) should have stars on all the arrows:

\[
\begin{align*}
    & P \xrightarrow{\gamma} P' \quad P'' \xrightarrow{\gamma'} P'' \\
    \overrightarrow{\gamma^*} & \quad P \xrightarrow{\gamma^*} P'
\end{align*}
\]

3. page 223. Several arrows – the ones for multi-step reduction – are missing their stars (as in the issue noted just previously).

4. page 227, part 1 of problem 8.8.3. The judgment to prove should have \(\leadsto^*\) instead of \(\leadsto\).