CSE526: Principles of Programming Languages (Spring 2003)
Scott Stoller
hw1 (version 12:10, 14feb2003), due in class on 6 Feb 2003

Each problem is worth 10 points. Please justify your answers.

Problem 1

Exercise 1.5, parts (a)-(c).
Answer: (a) Abstract syntax: $\langle\text{intexp}\rangle ::= \Sigma\langle\text{var}\rangle : \langle\text{intexp}\rangle \texttt{ to } \langle\text{intexp}\rangle.\ \langle\text{intexp}\rangle$
(b)

$$[\![\Sigma v : e_0 \texttt{ to } e_1.\ e_2]\!] = \sum_{k=[\![e_0]\!]_{\text{intexp}}\sigma}^{[\![e_1]\!]_{\text{intexp}}\sigma} [\![e_2]\!]_{\text{intexp}}[\sigma|v:k]$$

(c) Free vars: $\text{FV}_{\text{intexp}}(\Sigma v : e_0 \texttt{ to } e_1 .e_2) = \text{FV}_{\text{intexp}}(e_0) \cup \text{FV}_{\text{intexp}}(e_1) \cup (\text{FV}_{\text{intexp}}(e_2) \setminus \{v\})$.
Substitition: $(\Sigma v : e_0 \texttt{ to } e_1.\ e_2)/\delta = (\Sigma v' : e_0/\delta \texttt{ to } e_1/\delta.\ e_2/[\delta|v:v'])$
where $v' \notin \cup_{w \in \text{FV}_{\text{intexp}}(e_2)\setminus\{v\}} \text{FV}_{\text{intexp}}(\delta w)$

Problem 2

Exercise 2.2.
Answer: (a) Abstract syntax: $\langle\text{comm}\rangle ::= \textbf{repeat } \langle\text{comm}\rangle \textbf{ until } \langle\text{boolexp}\rangle$
(b) Semantics: $[\![\textbf{repeat } c \textbf{ until } b]\!]_{\text{comm}}\sigma = (Y_{\Sigma\to\Sigma_\perp} F)([\![c]\!]_{\text{comm}}\sigma)$
where $F f \sigma = \textbf{if } \neg[\![b]\!]_{\text{boolexp}}\sigma \textbf{ then } f_{\perp\!\!\perp}([\![c]\!]_{\text{comm}}\sigma) \textbf{ else } \sigma$
(c) Syntactic sugar: **repeat** c **until** b can be re-written as $c; \textbf{while } \neg b \textbf{ do } c$.
Proof of equivalence: start with $[\![c; \textbf{while } \neg b \textbf{ do } c]\!]_{\text{comm}}\sigma$, and expand it by unfolding the definitions of the meanings of sequential composition and while loops. it should be easy to show that the resulting expression is equivalent to the right side of the equation in part (b).

Problem 3

Exercise 2.4. Hint: Let $f_0, f_1, f_2, \ldots$ be a chain in $\Sigma \to \Sigma_\perp$. Show that $\sqcup_i F(f_i) = F(\sqcup_i f_i)$ by starting with $\sqcup_i F(f_i)$ and pushing the $\sqcup_i$ inwards through terms that do not depend on i.
Answer: I will use the characterization on continuity on the last line of page 31 and the first few lines of page 32. We need to show that F is monotonic and that F commutes with limits.
First, we show that F is monotonic. By definition, we need to show: if $f \sqsubseteq f'$ then $Ff \sqsubseteq Ff'$, *i.e.*,

$$\begin{aligned} & \textbf{if } [\![b]\!]_{\text{boolexp}}\sigma \textbf{ then } f_{\perp\!\!\perp}([\![c]\!]_{\text{comm}}\sigma) \textbf{ else } \sigma \\ \sqsubseteq\ & \textbf{if } [\![b]\!]_{\text{boolexp}}\sigma \textbf{ then } f'_{\perp\!\!\perp}([\![c]\!]_{\text{comm}}\sigma) \textbf{ else } \sigma \end{aligned}$$

It is easy to see that this follows from the fact that, for all states σ', $f(\sigma') \sqsubseteq f'(\sigma')$.
Now we show that F commutes with limits.

$$\begin{aligned} \sqcup_i F(f_i) &= \sqcup_i \textbf{if } [\![b]\!]_{\text{boolexp}}\sigma \textbf{ then } (f_i)_{\perp\!\!\perp}([\![c]\!]_{\text{comm}}\sigma) \textbf{ else } \sigma \\ &= \sqcup_i \textbf{if } [\![b]\!]_{\text{boolexp}}\sigma \textbf{ then } (\textbf{if } [\![c]\!]_{\text{comm}}\sigma = \perp \textbf{ then } \perp \textbf{ else } \sqcup_i f_i([\![c]\!]_{\text{comm}}\sigma)) \textbf{ else } \sigma \\ &= \sqcup_i \textbf{if } [\![b]\!]_{\text{boolexp}}\sigma \textbf{ then } (\textbf{if } [\![c]\!]_{\text{comm}}\sigma = \perp \textbf{ then } \perp \textbf{ else } (\sqcup_i f_i)([\![c]\!]_{\text{comm}}\sigma)) \textbf{ else } \sigma \\ &= F(\sqcup_i f_i) \end{aligned}$$

If we instead use the definition of continuity in the third-to-last paragraph on page 31, we don't need to show that F is monotonic, but we instead need to show that $\{Ff_0, Ff_1, \ldots\}$ has a least upper bound in $\Sigma \to \Sigma_\perp$. (An arbitrary set of elements of a domain D does not necessarily have a least upper bound in D.)