CS 4110

Programming Languages & Logics

Lecture 7
Denotational Semantics Proofs

12 September 2016

Announcements

- My office hours: today after class (after coffee)
- Homework #1 graded
  - out of 50, $\bar{x} = 46.1$, $\sigma = 4.6$, median 48

Determinism in Small-Step Semantics

Determinism: every configuration has at most one successor

$$\forall e \in \textbf{Exp}.\ \forall \sigma, \sigma', \sigma'' \in \textbf{Store}.\ \forall e', e'' \in \textbf{Exp}.$$
$$\text{if } \langle \sigma, e \rangle \to \langle \sigma', e' \rangle \text{ and } \langle \sigma, e \rangle \to \langle \sigma'', e'' \rangle$$
$$\text{then } e' = e'' \text{ and } \sigma' = \sigma''.$$

A different property, which you can call confluence:

$$\text{If } \langle \sigma, e \rangle \to^* \langle \sigma', e' \rangle \text{ and } \langle \sigma, e \rangle \to^* \langle \sigma'', e'' \rangle$$
$$\text{and neither } \langle \sigma', e' \rangle \text{ nor } \langle \sigma'', e'' \rangle \text{ can take a step}$$
$$\text{then } e' = e'' \text{ and } \sigma' = \sigma''.$$

Kleene Fixed-Point Theorem

Definition (Scott Continuity)

A function F is *Scott-continuous* if for every chain $X_1 \subseteq X_2 \subseteq \dots$ we have $F(\bigcup_i X_i) = \bigcup_i F(X_i)$.

Kleene Fixed-Point Theorem

Definition (Scott Continuity)

A function F is *Scott-continuous* if for every chain $X_1 \subseteq X_2 \subseteq \ldots$ we have $F(\bigcup_i X_i) = \bigcup_i F(X_i)$.

Theorem (Kleene Fixed Point)

Let F be a Scott-continuous function. The least fixed point of F is $\bigcup_i F^i(\emptyset)$.

Denotational Semantics for IMP Commands

$$\mathcal{C}[\![\textbf{skip}]\!] = \{(\sigma, \sigma)\}$$

$$\mathcal{C}[\![x := a]\!] = \{(\sigma, \sigma[x \mapsto n]) \mid (\sigma, n) \in \mathcal{A}[\![a]\!]\}$$

$$\mathcal{C}[\![c_1; c_2]\!] = \\ \{(\sigma, \sigma') \mid \exists \sigma''. ((\sigma, \sigma'') \in \mathcal{C}[\![c_1]\!] \land (\sigma'', \sigma') \in \mathcal{C}[\![c_2]\!])\}$$

$$\mathcal{C}[\![\textbf{if } b \textbf{ then } c_1 \textbf{ else } c_2]\!] = \\ \{(\sigma, \sigma') \mid (\sigma, \textbf{true}) \in \mathcal{B}[\![b]\!] \land (\sigma, \sigma') \in \mathcal{C}[\![c_1]\!]\} \cup \\ \{(\sigma, \sigma') \mid (\sigma, \textbf{false}) \in \mathcal{B}[\![b]\!] \land (\sigma, \sigma') \in \mathcal{C}[\![c_2]\!]\}$$

$$\mathcal{C}[\![\textbf{while } b \textbf{ do } c]\!] = \mathit{fix}(f)$$

where $F(f) = \{(\sigma, \sigma) \mid (\sigma, \textbf{false}) \in \mathcal{B}[\![b]\!]\} \cup$
$\{(\sigma, \sigma') \mid (\sigma, \textbf{true}) \in \mathcal{B}[\![b]\!] \land \exists \sigma''. ((\sigma, \sigma'') \in \mathcal{C}[\![c]\!] \land (\sigma'', \sigma') \in f)\}$

Exercises

skip; *c* and *c*; **skip** are equivalent.

$$\mathcal{C}[\![\text{skip}; c]\!] = \{(\sigma, \sigma'') \mid \exists \sigma'.\ (\sigma, \sigma') \in \mathcal{C}[\![\text{skip}]\!] \wedge (\sigma', \sigma'') \in \mathcal{C}[\![c]\!]\}$$

$$\sigma = \sigma'$$

Exercises

skip; *c* and *c*; **skip** are equivalent.

$\mathcal{C}$⟦**while false do** *c*⟧ is equivalent to...

$$= \mathrm{fix}(F)$$

$$F\,f = \{(\sigma,\sigma) \mid (\sigma,\mathrm{false}) \in \mathcal{B}[\![b]\!]\} \cup \{(\sigma,\sigma'') \mid (\sigma,\mathrm{true}) \in \mathcal{B}[\![b]\!] \land (\sigma,\sigma') \in [\![c]\!] \land (\sigma',\sigma'') \in f\}$$

$$\mathrm{fix}(F) = \bigcup_i F^i(\emptyset)$$

Exercises

skip; *c* and *c*; **skip** are equivalent.

$\mathcal{C}[\![$**while false do** $c]\!]$ is equivalent to...

$\mathcal{C}[\![$**while true do skip**$]\!]$

$$P(F^i(\emptyset))$$

$$P(F^0(\emptyset))$$

$$P(F^i(\emptyset)) \Rightarrow P(F^{i+1}(\emptyset))$$

$$= F(F^i(\emptyset))$$