

CS 4110

Programming Languages & Logics

Lecture 8

Denotational Semantics Proofs



Determinism in Small-Step Semantics

Determinism: every configuration has at most one successor

$\forall e \in \mathbf{Exp}. \forall \sigma, \sigma', \sigma'' \in \mathbf{Store}. \forall e', e'' \in \mathbf{Exp}.$
if $\langle \sigma, e \rangle \rightarrow \langle \sigma', e' \rangle$ and $\langle \sigma, e \rangle \rightarrow \langle \sigma'', e'' \rangle$
then $e' = e''$ and $\sigma' = \sigma''$.

A different property, which you can call **confluence**:

If $\langle \sigma, e \rangle \rightarrow^* \langle \sigma', e' \rangle$ and $\langle \sigma, e \rangle \rightarrow^* \langle \sigma'', e'' \rangle$
and neither $\langle \sigma', e' \rangle$ nor $\langle \sigma'', e'' \rangle$ can take a step
then $e' = e''$ 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 $X_1 \subseteq X_2 \subseteq \dots$ 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

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

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

$$\begin{aligned} \mathcal{C}[c_1; c_2] = \\ \{(\sigma, \sigma') \mid \exists \sigma''. ((\sigma, \sigma'') \in \mathcal{C}[c_1] \wedge (\sigma'', \sigma') \in \mathcal{C}[c_2])\} \end{aligned}$$

$$\begin{aligned} \mathcal{C}[\mathbf{if } b \mathbf{ then } c_1 \mathbf{ else } c_2] = \\ \{(\sigma, \sigma') \mid (\sigma, \mathbf{true}) \in \mathcal{B}[b] \wedge (\sigma, \sigma') \in \mathcal{C}[c_1]\} \cup \\ \{(\sigma, \sigma') \mid (\sigma, \mathbf{false}) \in \mathcal{B}[b] \wedge (\sigma, \sigma') \in \mathcal{C}[c_2]\} \end{aligned}$$

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

$$\begin{aligned} \text{where } F(f) = \{(\sigma, \sigma) \mid (\sigma, \mathbf{false}) \in \mathcal{B}[b]\} \cup \\ \{(\sigma, \sigma') \mid (\sigma, \mathbf{true}) \in \mathcal{B}[b] \wedge \exists \sigma''. ((\sigma, \sigma'') \in \mathcal{C}[c] \wedge \\ (\sigma'', \sigma') \in f)\} \end{aligned}$$

Exercises

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

Exercises

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

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

Exercises

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

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

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