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

$$\forall e \in \mathbf{Exp}. \ \forall \sigma, \sigma', \sigma'' \in \mathbf{Store}. \ \forall e', e'' \in \mathbf{Exp}.$$
 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 $\sigma' = \sigma''$.

3

Kleene Fixed-Point Theorem

Definition (Scott Continuity)

A function F is Scott-continuous if for every chain $X_1 \subseteq X_2 \subseteq ...$ 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 $X_1 \subseteq X_2 \subseteq ...$ 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}\llbracket \mathsf{skip} \rrbracket = \{(\sigma, \sigma)\}
C[\![x := a]\!] = \{(\sigma, \sigma[x \mapsto n]) \mid (\sigma, n) \in A[\![a]\!]\}
  \mathcal{C}\llbracket c_1; c_2 \rrbracket =
                                                                     \{(\sigma,\sigma')\mid \exists \sigma''. ((\sigma,\sigma'')\in \mathcal{C}\llbracket c_1\rrbracket \land (\sigma'',\sigma')\in \mathcal{C}\llbracket c_2\rrbracket)\}
  \mathcal{C}\llbracket \text{if } b \text{ then } c_1 \text{ else } c_2 \rrbracket =
                                                                     \{(\sigma, \sigma') \mid (\sigma, \mathsf{true}) \in \mathcal{B}\llbracket b \rrbracket \land (\sigma, \sigma') \in \mathcal{C}\llbracket c_1 \rrbracket \} \cup
                                                                     \{(\sigma, \sigma') \mid (\sigma, \mathsf{false}) \in \mathcal{B}\llbracket b \rrbracket \land (\sigma, \sigma') \in \mathcal{C}\llbracket c_2 \rrbracket \}
  \mathcal{C}[\![\mathbf{while}\ b\ \mathbf{do}\ c]\!] = fix(f)
  where F(f) = \{(\sigma, \sigma) \mid (\sigma, \mathbf{false}) \in \mathcal{B}[\![b]\!]\} \cup
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         talse
                                                                   \{(\sigma, \sigma') \mid (\sigma, \mathsf{true}) \in \mathcal{B}[b] \land (\sigma, \sigma'') \in \mathcal{C}[c] \land (\sigma, \sigma'') \in \mathcal{
                                                                                                                                                                                                                                                                                                                                                                                                                                                        F (4) = $
```

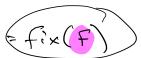
Exercises

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

Exercises

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

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



Exercises

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

C while false do c is equivalent to...

C [while true do skip]

$$P(F(\delta))$$

$$P(F(\delta)) \Rightarrow P(F(\delta))$$

$$= F(F(\delta))$$