CS 520
Theory of Programming Language

04/19 – 04/28, 2021

$\boxed{\text{Chap 1 : Transition Semantics.}}$ ····· small-step operational semantics.

(big-step op. semantics.)

1. Motivation / Big Picture.

① Denotational semantics. ····· programs into math. entities.

+ rigorous, reveals math. structures.

− too complex. doesn't say much about intermediate steps.

② Operational semantics. (Transition semantics).

↳ simple abstract impl. of an interpreter.

rigorous.

③ e.g. $x := 3 \, ; \, y := x + 4$

(1) $[\![x := 3 \, ; \, y := x + 4]\!] : \Sigma \to \Sigma_\perp$

$[\![x := 3 \, ; \, y := x + 4]\!] \, 6 = [6 \mid x : 3 \mid y : 7]$.

(2) $\langle \underbrace{x := 3 \, ; \, y := x + 4}_{\text{command}} , 6 \rangle \xrightarrow{} \langle y := x + 4 , [6 \mid x : 3] \rangle$

pc

transition relation.

$\downarrow$

$[\, [6 \mid x : 3] \mid y : 7] = [6 \mid x : 3 \mid y : 7]$.

Note: $\langle x := 3 \, ; \, \text{skip} \, ; \, y := x + 4 , 6 \rangle \xrightarrow{x := 3} \cdots \xrightarrow{\text{skip}} \cdots \xrightarrow{y := x + 4} [6 \mid x : 3 \mid y : 7]$

2. General recipe for defining small-step op. semantics (trans. semantics).

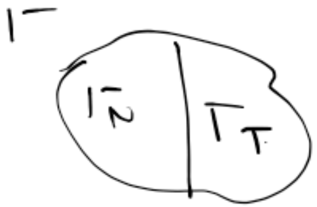
Goal: Define $\to$ ... relation that describes what happens in a single comp. step.

Define the set of:

① Configurations. $\Gamma = \Gamma_N \cup \Gamma_T$ $\quad (\Gamma_N \cap \Gamma_T = \phi)$

Γ_T: conf. for completed computations

Γ_N: conf. for incomplete comput.



Define transition relation.

② $\to \subseteq \Gamma_N \times \Gamma$ $\quad \gamma \in \Gamma_N, \ \gamma' \in \Gamma$

$(\gamma, \gamma') \in \to$ $\quad \gamma \to \gamma'$

γ unfinished.

$\vdots$

$\left(\to \subseteq \Gamma_N \times \Lambda \times \Gamma \quad \gamma \xrightarrow{\lambda} \gamma' \quad ((\gamma, \lambda, \gamma') \in \to) \right)$

3. Simple imp. PL.

$$\langle comm \rangle ::= \text{skip} \mid \langle var \rangle := \langle int\,exp \rangle \mid \langle comm \rangle ; \langle comm \rangle$$

$$\mid \text{if } \langle boolexp \rangle \text{ then } \langle comm \rangle \text{ else } \langle comm \rangle.$$

$$\mid \text{while } \langle boolexp \rangle \text{ do } \langle comm \rangle$$

1 $\Gamma_N \stackrel{def}{=} \langle comm \rangle \times \Sigma.$ $\left(\Sigma \stackrel{def.}{=} \langle var \rangle \to \mathbb{Z} \right)$

$\Gamma_T \stackrel{def.}{=} \Sigma.$, $\Gamma \stackrel{def}{=} \Gamma_N \cup \Gamma_T$

② $\longrightarrow \;\subseteq\; \Gamma_N \times \Gamma$

$$\frac{}{\langle \mathrm{fail}, \sigma \rangle \xrightarrow{\varepsilon} \langle \mathrm{abort}, \sigma \rangle}$$

$$\frac{\langle c_1, \sigma \rangle \xrightarrow{\lambda} \langle \mathrm{abort}, \sigma' \rangle}{\langle c_1 ; c_2, \sigma \rangle \xrightarrow{\lambda} \langle \mathrm{abort}, \sigma' \rangle}$$

$$\frac{}{\langle \mathrm{skip}, \sigma \rangle \xrightarrow{\varepsilon} \sigma}$$

$$\frac{}{\langle x := e, \sigma \rangle \xrightarrow{\varepsilon} [\sigma \mid x : \llbracket e \rrbracket \sigma]}$$

$$\frac{}{\langle ?x, \sigma \rangle \xrightarrow{?n} \sigma[\sigma \mid x : n]}$$

$$\frac{\langle c_1, \sigma \rangle \xrightarrow{\lambda} \sigma'}{\langle c_1 ; c_2, \sigma \rangle \xrightarrow{\lambda} \langle c_2, \sigma' \rangle}$$

$$\frac{\langle c_1, \sigma \rangle \xrightarrow{\lambda} \langle c_1', \sigma' \rangle}{\langle c_1 ; c_2, \sigma \rangle \xrightarrow{\lambda} \langle c_1' ; c_2, \sigma' \rangle}$$

$$\frac{}{\langle !e, \sigma \rangle \xrightarrow{!\llbracket e \rrbracket \sigma} \sigma}$$

$$\frac{\llbracket b \rrbracket \sigma = tt}{\langle \mathrm{if}\ b\ \mathrm{then}\ c_1\ \mathrm{else}\ c_2, \sigma \rangle \xrightarrow{\varepsilon} \langle c_1, \sigma \rangle}$$

$$\frac{\llbracket b \rrbracket \sigma = ff}{\langle \mathrm{if}\ b\ \mathrm{then}\ c_1\ \mathrm{else}\ c_2, \sigma \rangle \xrightarrow{\varepsilon} \langle c_2, \sigma \rangle}$$

$$\frac{\llbracket b \rrbracket \sigma = tt}{\langle \mathrm{while}\ b\ \mathrm{do}\ c, \sigma \rangle \xrightarrow{\varepsilon} \langle c ; \mathrm{while}\ b\ \mathrm{do}\ c, \sigma \rangle}$$

$$\frac{\llbracket b \rrbracket \sigma = ff}{\langle \mathrm{while}\ b\ \mathrm{do}\ c, \sigma \rangle \xrightarrow{\varepsilon} \sigma}$$

③ (1) $\xrightarrow{*} = \bigcup_{n=0}^{\infty} (\rightarrow)^n$

(2) $\rightarrow$ .... deterministic. $(\gamma \rightarrow \gamma'$ and $\gamma \rightarrow \gamma'' \Rightarrow \gamma' = \gamma'')$

$\rightarrow$ .... progress. $\forall \gamma \in \Gamma_N.\ \exists \gamma' \in \Gamma.\ \gamma \rightarrow \gamma'$.

(3) $\gamma \in \Gamma$ [poss 1] $\gamma \rightarrow \gamma_1 \rightarrow \gamma_2 \rightarrow \gamma_3 \rightarrow \cdots\cdots$ infinite. $(\gamma \uparrow)$.

[poss 2] $\gamma \rightarrow \gamma_1 \rightarrow \gamma_2 \rightarrow \cdots \rightarrow \gamma_n$. $(\gamma \xrightarrow{*} \gamma_n)$.

$\gamma_n \in \Gamma_T$.

$$F(\gamma) = \begin{cases} \bot & \text{if the first case } (\gamma \uparrow) \\ \gamma_n & \text{otherwise.} \end{cases}$$

$\gamma = \langle c, \sigma \rangle$, $\gamma_n = \sigma'$

[thm] c ... command, $\sigma \in \Sigma$.

$\Rightarrow$. $[\![c]\!]\sigma = F(\langle c, \sigma \rangle)$.

3. Extension: Include fail.

$$\hat{\Sigma} = \Sigma \cup \{abort\} \times \Sigma.$$

$$[\![fail]\!] \sigma = \langle abort, \sigma \rangle.$$

$$[\![fail ; c]\!] \sigma = [\![c]\!]_{\perp\!\!\perp} ([\![fail]\!] \sigma) = [\![c]\!]_{\perp\!\!\perp} (\langle abort, \sigma \rangle) = \langle abort, \sigma \rangle.$$

① $\Gamma = \underbrace{\Gamma_N}_{\text{not changed.}} \cup \Gamma_T$

$\qquad \Gamma_T = \Sigma \cup \{abort\} \times \Sigma.$

② → … How to define it? Very similar to what we did.

(1) fail. (2) $\dfrac{\ldots \to \ldots}{\ldots \to \ldots}$ add one more case.

4. Extension: Input/output.

$$\langle \mathrm{comm} \rangle ::= \dots \mid \mathbf{fail} \mid ?\langle \mathrm{var} \rangle \mid !\langle \mathrm{intexp} \rangle.$$

① $\underbrace{\Gamma = \Gamma_N \cup \Gamma_T}_{\text{same as Extension 3.}}$

② $\rightarrow \; \subseteq \Gamma_N \times \Lambda \times \Gamma$

$$\Lambda = (\{\varepsilon\} \cup \{!n \mid n \in \mathbb{Z}\} \cup \{?n \mid n \in \mathbb{Z}\}).$$

$$\gamma \xrightarrow{\varepsilon} \gamma', \quad \gamma \xrightarrow{!3} \gamma', \quad \gamma \xrightarrow{?4} \gamma'$$

③

$$\gamma \xrightarrow{\varepsilon} \gamma_1 \xrightarrow{\varepsilon} \gamma_2 \xrightarrow{!3} \gamma_3 \xrightarrow{\varepsilon} \gamma_4 \xrightarrow{\varepsilon} \gamma_5 \xrightarrow{?4} \gamma_6 \xrightarrow{\varepsilon} \sigma'$$

$$\gamma_5 \xrightarrow{?5} \gamma_6' \rightarrow \sigma''$$

$$\gamma_5 \xrightarrow{?-} \gamma_6'' \rightarrow \sigma''$$

$$\omega \in \Omega$$

$$J_{out}\left(\left\langle 3, J_{in}\left(\lambda n.\ \begin{cases} \sigma' & \text{if } n = 4 \\ \sigma'' & \text{otherwise.} \end{cases}\right)\right\rangle\right)$$