

4541.664A Homework 3

Semantics

Sungkeun Cho, Yongho Yoon

Exercise 1

Structural Operational Semantics

Semantic Domain

$$v \in Val = \mathbb{B}$$

Inference Rule

$$\perp \Rightarrow false$$

$$\top \Rightarrow true$$

$$\frac{C \Rightarrow v}{\neg C \Rightarrow !v}$$

$$\frac{C_1 \Rightarrow v_1 \quad C_2 \Rightarrow v_2}{\times C_1 C_2 \Rightarrow v_1 \wedge v_2}$$

$$\frac{C_1 \Rightarrow v_1 \quad C_2 \Rightarrow v_2}{+ C_1 C_2 \Rightarrow v_1 \vee v_2}$$

$$\frac{C_1 \Rightarrow v}{\oplus C_1 C_2 \Rightarrow v}$$

$$\frac{C_2 \Rightarrow v}{\oplus C_1 C_2 \Rightarrow v}$$

$$\frac{C_c \Rightarrow true \quad C_t \Rightarrow v_t}{\otimes C_c C_t C_f \Rightarrow v_t}$$

$$\frac{C_c \Rightarrow false \quad C_f \Rightarrow v_f}{\otimes C_c C_t C_f \Rightarrow v_f}$$

Evaluation Context Semantics

Semantic Domain

$$v \in Val = \mathbb{B}$$

Evaluation Context

$$\begin{array}{c} K \rightarrow [] \\ | \\ \neg K \\ | \\ \times K C \\ | \\ \times v K \\ | \\ + K C \\ | \\ + v K \\ | \\ \oplus K C \\ | \\ \oplus C K \\ | \\ \otimes K C C \end{array}$$

Rewriting Rule

$$\frac{K[C] \rightarrow K[C']}{C \rightarrow C'}$$

$$\begin{array}{l} \perp \rightarrow false \\ \top \rightarrow true \\ \neg v \rightarrow !v \\ \times v_1 v_2 \rightarrow v_1 \wedge v_2 \\ + v_1 v_2 \rightarrow v_1 \vee v_2 \\ \oplus v C \rightarrow v \\ \oplus C v \rightarrow v \\ \otimes true C_1 C_2 \rightarrow C_1 \\ \otimes false C_1 C_2 \rightarrow C_2 \end{array}$$

Transition Semantics

Semantic Domain

$$v \in Val = \mathbb{B}$$

Inference Rule

$$\overline{\perp \rightarrow false}$$

$$\overline{\top \rightarrow true}$$

$$\frac{C \rightarrow C'}{\neg C \rightarrow \neg C'}$$

$$\overline{\neg v \rightarrow !v}$$

$$\frac{C_1 \rightarrow C'_1}{\times C_1 C_2 \rightarrow \times C'_1 C_2}$$

$$\begin{array}{c}
\frac{C \rightarrow C'}{\times v C \rightarrow \times v C'} \\
\frac{}{\times v v' \rightarrow v \wedge v'} \\
\frac{C_1 \rightarrow C'_1}{+ C_1 C_2 \rightarrow + C'_1 C_2} \\
\frac{}{+ v C \rightarrow + v C'} \\
\frac{}{+ v v' \rightarrow v \vee v'} \\
\frac{C_1 \rightarrow C'_1}{\oplus C_1 C_2 \rightarrow \oplus C'_1 C_2} \\
\frac{C_2 \rightarrow C'_2}{\oplus C_1 C_2 \rightarrow \oplus C_1 C'_2} \\
\frac{}{\oplus v C \rightarrow v} \\
\frac{}{\oplus C v \rightarrow v} \\
\frac{C \rightarrow C'}{\otimes C C_t C_f \rightarrow \otimes C' C_t C_f} \\
\frac{}{\otimes true C_t C_f \rightarrow C_t} \\
\frac{}{\otimes false C_t C_f \rightarrow C_f}
\end{array}$$

Exercise 2

Evaluation Context Semantics

Semantic Domain

$$\begin{array}{llll}
x & \in & Var & = \text{Program Variable} \\
v & \in & Value & = \mathbb{Z} \\
M & \in & Memory & = Var \xrightarrow{\text{fin}} Value
\end{array}$$

Evaluation Context

$$K \rightarrow []
\begin{array}{l}
| \quad x := K \\
| \quad K ; C \\
| \quad \text{done} ; K \\
| \quad \text{if } K \text{ then } C \text{ else } C \\
| \quad \text{while } K C \\
| \quad K + E \\
| \quad v + K \\
| \quad - K
\end{array}$$

Rewriting Rule

$$\begin{array}{c}
 \frac{(M, C) \rightarrow (M', C')}{(M, K[C]) \rightarrow (M', K[C'])} \\
 \frac{(M, E) \rightarrow (M, E')}{(M, K[E]) \rightarrow (M, K[E'])} \\
 \begin{array}{lll}
 (M, x := v) & \rightarrow & (M\{x \mapsto v\}, \text{done}) \\
 (M, \text{done} ; \text{done}) & \rightarrow & (M, \text{done}) \\
 (M, \text{if } v \text{ then } C_1 \text{ else } C_2) & \rightarrow & (M, C_1) \quad (v \neq 0) \\
 (M, \text{if } 0 \text{ then } C_1 \text{ else } C_2) & \rightarrow & (M, C_2) \\
 (M, \text{while } 0 C) & \rightarrow & (M, \text{done}) \\
 (M, \text{while}_E v C) & \rightarrow & (M, C ; \text{while } E C) \quad (v \neq 0) \\
 (M, v_1 + v_2) & \rightarrow & (M, v) \quad (v = v_1 + v_2) \\
 (M, -v) & \rightarrow & (M, v') \quad (v' = -v)
 \end{array}
 \end{array}$$

Abstract Machine Semantics

Semantic Domain

$$\begin{array}{llll}
 x & \in & Var & = \text{Program Variable} \\
 v & \in & Value & = \mathbb{Z} \\
 M & \in & Memory & = Var \xrightarrow{\text{fin}} Value
 \end{array}$$

Machine

$$\langle S, M, C \rangle \in Stack \times Memory \times Cmd$$

$$\begin{array}{lll}
 S & \rightarrow & \epsilon \quad (\text{empty stack}) \\
 & | & n.S \quad (n \in \mathbb{Z}) \\
 C & \rightarrow & \epsilon \quad (\text{empty command}) \\
 & | & \text{nop}.C \\
 & | & \text{push}(n).C \\
 & | & \text{add}.C \\
 & | & \text{neg}.C \\
 & | & \text{jmpnz}(C, C).C \\
 & | & \text{store}(x).C \\
 & | & \text{load}(x).C \\
 & | & \text{loop}(C, C).C
 \end{array}$$

Translation

$$\begin{array}{lcl}
 \llbracket \text{skip} \rrbracket & \rightarrow & \text{nop} \\
 \llbracket x := E \rrbracket & \rightarrow & \llbracket E \rrbracket.\text{store}(x) \\
 \llbracket \text{if } E \text{ then } C_1 \text{ else } C_2 \rrbracket & \rightarrow & \llbracket E \rrbracket.\text{jmpnz}(\llbracket C_1 \rrbracket, \llbracket C_2 \rrbracket) \\
 \llbracket \text{while } E \text{ } C \rrbracket & \rightarrow & \llbracket E \rrbracket.\text{loop}(\llbracket E \rrbracket, \llbracket C \rrbracket) \\
 \llbracket C_1 ; C_2 \rrbracket & \rightarrow & \llbracket C_1 \rrbracket.[\llbracket C_2 \rrbracket] \\
 \\
 \llbracket n \rrbracket & \rightarrow & \text{push}(n) \\
 \llbracket E_1 + E_2 \rrbracket & \rightarrow & \llbracket E_1 \rrbracket.[\llbracket E_2 \rrbracket].\text{add} \\
 \llbracket -E \rrbracket & \rightarrow & \llbracket E \rrbracket.\text{neg}
 \end{array}$$

Transition

$$\begin{array}{lcl}
 \langle S, M, \text{nop}.C \rangle & \rightarrow & \langle S, M, C \rangle \\
 \langle S, M, \text{push}(n).C \rangle & \rightarrow & \langle n.S, M, C \rangle \\
 \langle v_2.v_1.S, M, \text{add}.C \rangle & \rightarrow & \langle v.S, M, C \rangle \quad (v = v_1 + v_2) \\
 \langle v.S, M, \text{neg}.C \rangle & \rightarrow & \langle -v.S, M, C \rangle \\
 \langle v.S, M, \text{jmpnz}(C_1, C_2).C \rangle & \rightarrow & \langle S, M, C_1.C \rangle \quad (v \neq 0) \\
 \langle 0.S, M, \text{jmpnz}(C_1, C_2).C \rangle & \rightarrow & \langle S, M, C_2.C \rangle \\
 \langle v.S, M, \text{store}(x).C \rangle & \rightarrow & \langle S, M\{x \mapsto v\}, C \rangle \\
 \langle S, M, \text{load}(x).C \rangle & \rightarrow & \langle M(x).S, M, C \rangle \\
 \langle 0.S, M, \text{loop}(C_1, C_2).C \rangle & \rightarrow & \langle S, M, C \rangle \\
 \langle v.S, M, \text{loop}(C_1, C_2).C \rangle & \rightarrow & \langle S, M, C_2.C_1.\text{loop}(C_1, C_2).C \rangle \quad (v \neq 0)
 \end{array}$$

Exercise 3

Structural Operational Semantics

Semantic Domain

$$\begin{array}{llll}
 l & \in & \text{Addr} \\
 \sigma & \in & \text{Env} & = \text{Var} \xrightarrow{\text{fin}} \text{Addr} \\
 M & \in & \text{Memory} & = \text{Addr} \xrightarrow{\text{fin}} \text{Val} \\
 v & \in & \text{Val} & = \mathbb{Z} + \text{Addr}
 \end{array}$$

$\sigma, M \vdash C \Rightarrow M'$

$$\frac{}{\sigma, M \vdash \text{skip} \Rightarrow M}$$

$$\frac{\sigma, M \vdash E \Rightarrow v}{\sigma, M \vdash x := E \Rightarrow M\{\sigma(x) \mapsto v\}}$$

$$\frac{\sigma, M \vdash E \Rightarrow v \quad \sigma, M \vdash x \Rightarrow l}{\sigma, M \vdash *x := E \Rightarrow M\{l \mapsto v\}}$$

$$\begin{array}{c}
\frac{\sigma, M \vdash C_1 \Rightarrow M' \quad \sigma, M' \vdash C_2 \Rightarrow M''}{\sigma, M \vdash C_1 ; C_2 \Rightarrow M''} \\
\frac{\sigma, M \vdash E \Rightarrow 0 \quad \sigma, M \vdash C_2 \Rightarrow M'}{\sigma, M \vdash \text{if } E \ C_1 \ C_2 \Rightarrow M'} \\
\frac{\sigma, M \vdash E \Rightarrow v \quad \sigma, M \vdash C_1 \Rightarrow M'}{\sigma, M \vdash \text{if } E \ C_1 \ C_2 \Rightarrow M'} \ v \neq 0 \\
\frac{\sigma, M \vdash E \Rightarrow 0}{\sigma, M \vdash \text{while } E \ C \Rightarrow M} \\
\frac{\sigma, M \vdash E \Rightarrow v \quad \sigma, M \vdash C \Rightarrow M' \quad \sigma, M' \vdash \text{while } E \ C \Rightarrow M''}{\sigma, M \vdash \text{while } E \ C \Rightarrow M''} \ v \neq 0 \\
\frac{\sigma, M \vdash E \Rightarrow v \quad \sigma\{x \mapsto l\}, M\{l \mapsto v\} \vdash C \Rightarrow M'}{\sigma, M \vdash \text{local } x := E \ \text{in } C \Rightarrow M'} \ l \notin \text{Dom } M \\
\boxed{\sigma, M \vdash E \Rightarrow v} \quad \overline{\sigma, M \vdash n \Rightarrow n} \\
\overline{\sigma, M \vdash x \Rightarrow M(\sigma(x))} \\
\frac{\sigma, M \vdash x \Rightarrow l}{\sigma, M \vdash *x \Rightarrow M(l)} \\
\overline{\sigma, M \vdash &x \Rightarrow \sigma(x)} \\
\frac{\sigma, M \vdash E_1 \Rightarrow v_1 \quad \sigma, M \vdash E_2 \Rightarrow v_2}{\sigma, M \vdash E_1 + E_2 \Rightarrow v_1 + v_2} \\
\frac{\sigma, M \vdash E \Rightarrow v}{\sigma, M \vdash -E \Rightarrow -v}
\end{array}$$

Transition Semantics(Small-Step Operational Semantics)

Semantic Domain

$$\begin{array}{llll}
l & \in & Addr \\
\sigma & \in & Env & = Var \times Addr \ list \\
M & \in & Memory & = Addr \xrightarrow{\text{fin}} Val \\
v & \in & Val & = \mathbb{Z} + Addr
\end{array}$$

$$lookup \in Env \rightarrow Var \rightarrow Val$$

lookup : find the foremost binding of x in σ

$$\sigma(x) = lookup \sigma x$$

$$\boxed{(M, \sigma, C) \rightarrow (M', \sigma', C')}$$

$$\overline{(M, \sigma, \text{skip}) \rightarrow (M, \sigma, \text{done})}$$

$$\frac{(M, \sigma, E) \rightarrow (M, \sigma, E')}{(M, \sigma, x := E) \rightarrow (M, \sigma, x := E')}$$

$$\overline{(M, \sigma, x := v) \rightarrow (M\{\sigma(x) \mapsto v\}, \sigma, \text{done})}$$

$$\frac{(M, \sigma, E) \rightarrow (M, \sigma, E')}{(M, \sigma, *x := E) \rightarrow (M, \sigma, *x := E')}$$

$$\overline{(M, \sigma, *x := v) \rightarrow (M\{M(\sigma(x)) \mapsto v\}, \sigma, \text{done})}$$

$$\frac{(M, \sigma, C_1) \rightarrow (M', \sigma', C'_1)}{(M, \sigma, C_1 ; C_2) \rightarrow (M', \sigma', C'_1 ; C_2)}$$

$$\overline{(M, \sigma, \text{done} ; C_2) \rightarrow (M, \sigma, C_2)}$$

$$\frac{(M, \sigma, E) \rightarrow (M, \sigma, E')}{(M, \sigma, \text{if } E \ C_1 \ C_2) \rightarrow (M, \sigma, \text{if } E' \ C_1 \ C_2)}$$

$$\overline{(M, \sigma, \text{if } v \ C_1 \ C_2) \rightarrow (M, \sigma, C_1)} \ v \neq 0$$

$$\overline{(M, \sigma, \text{if } 0 \ C_1 \ C_2) \rightarrow (M, \sigma, C_2)}$$

$$\overline{(M, \sigma, \text{while } E \ C) \rightarrow (M, \sigma, \text{if } E \ (C ; \text{while } E \ C) \ (\text{done}))}$$

$$\frac{(M, \sigma, E) \rightarrow (M, \sigma, E')}{(M, \sigma, \text{local } x := E \ \text{in } C) \rightarrow (M, \sigma, \text{local } x := E' \ \text{in } C)}$$

$$\overline{(M, \sigma, \text{local } x := v \ \text{in } C) \rightarrow (M\{l \mapsto v\}, (x, l) :: \sigma, C ; \text{ldone})} \ l \notin \text{Dom } M$$

$$\overline{(M, (x, l) :: \sigma, \text{ldone}) \rightarrow (M, \sigma, \text{done})}$$

$$\boxed{(M, \sigma, E) \rightarrow (M', \sigma', E')}$$

$$\overline{(M, \sigma, n) \rightarrow (M, \sigma, v)} \ v = n$$

$$\overline{(M, \sigma, x) \rightarrow (M, \sigma, M(\sigma(x)))}$$

$$\overline{(M, \sigma, *x) \rightarrow (M, \sigma, M(M(\sigma(x))))}$$

$$\overline{(M, \sigma, \&x) \rightarrow (M, \sigma, \sigma(x))}$$

$$\frac{(M, \sigma, E_1) \rightarrow (M, \sigma, E'_1)}{(M, \sigma, E_1 + E_2) \rightarrow (M, \sigma, E'_1 + E_2)}$$

$$\begin{array}{c}
\dfrac{(M, \sigma, E_2) \rightarrow (M, \sigma, E'_2)}{(M, \sigma, v + E_2) \rightarrow (M, \sigma, v + E'_2)} \\
\dfrac{}{(M, \sigma, v_1 + v_2) \rightarrow (M, \sigma, v)} \quad v = v_1 + v_2 \\
\dfrac{(M, \sigma, E) \rightarrow (M, \sigma, E')}{(M, \sigma, -E) \rightarrow (M, \sigma, -E')} \\
\dfrac{}{(M, \sigma, -v) \rightarrow (M, \sigma, v')} \quad v' = -v
\end{array}$$

Abstract Machine Semantics

Machine

$\langle S, M, E, C \rangle \in Stack \times Memory \times Environment \times Cmd$

$$\begin{array}{lcl}
S & \rightarrow & \epsilon \quad (\text{empty stack}) \\
& | & n.S \quad (n \in \mathbb{Z}) \\
& | & l.S \quad (l \in Addr) \\
E & \rightarrow & \epsilon \quad (\text{empty environment}) \\
& | & (x, l).E \\
C & \rightarrow & \epsilon \quad (\text{empty command}) \\
& | & \text{push}(n).C \\
& | & \text{push}(x).C \\
& | & \text{add}.C \\
& | & \text{neg}.C \\
& | & \text{jmpz}(C, C).C \\
& | & \text{store}.C \\
& | & \text{load}.C \\
& | & \text{loop}(C, C).C \\
& | & \text{bind}(x).C \\
& | & \text{unbind}.C
\end{array}$$

Translation

$$\begin{array}{ll}
\llbracket x := E \rrbracket & \rightarrow \llbracket E \rrbracket.\text{push}(x).\text{store} \\
\llbracket *x := E \rrbracket & \rightarrow \llbracket E \rrbracket.\text{push}(x).\text{load.store} \\
\llbracket \text{if } E \text{ then } C_1 \text{ else } C_2 \rrbracket & \rightarrow \llbracket E \rrbracket.\text{jumpz}(\llbracket C_1 \rrbracket, \llbracket C_2 \rrbracket) \\
\llbracket \text{while } E \text{ } C \rrbracket & \rightarrow \llbracket E \rrbracket.\text{loop}(\llbracket E \rrbracket, \llbracket C \rrbracket) \\
\llbracket C_1 ; C_2 \rrbracket & \rightarrow \llbracket C_1 \rrbracket. \llbracket C_2 \rrbracket \\
\llbracket \text{local } x := E \text{ in } C \rrbracket & \rightarrow \text{bind}(x).\llbracket E \rrbracket.\text{push}(x).\text{store}.C.\text{unbind} \\
\\
\llbracket n \rrbracket & \rightarrow \text{push}(n) \\
\llbracket x \rrbracket & \rightarrow \text{push}(x).\text{load} \\
\llbracket *x \rrbracket & \rightarrow \text{push}(x).\text{load.load} \\
\llbracket & \& x \rrbracket & \rightarrow \text{push}(x) \\
\llbracket E_1 + E_2 \rrbracket & \rightarrow \llbracket E_1 \rrbracket. \llbracket E_2 \rrbracket.\text{add} \\
\llbracket -E \rrbracket & \rightarrow \llbracket E \rrbracket.\text{neg}
\end{array}$$

Transition

$\langle S, M, E, \text{push}(n).C \rangle$	\rightarrow	$\langle n.S, M, E, C \rangle$
$\langle S, M, E, \text{push}(x).C \rangle$	\rightarrow	$\langle l.S, M, E, C \rangle$ (if (x, l) is the first such entry in E)
$\langle v_2.v_1.S, M, \text{add}.C \rangle$	\rightarrow	$\langle v.S, M, E, C \rangle$ ($v = v_1 + v_2$)
$\langle v.S, M, E, \text{neg}.C \rangle$	\rightarrow	$\langle -v.S, M, E, C \rangle$
$\langle 0.S, M, E, \text{jmpz}(C_1, C_2).C \rangle$	\rightarrow	$\langle S, M, E, C_1.C \rangle$
$\langle v.S, M, E, \text{jmpz}(C_1, C_2).C \rangle$	\rightarrow	$\langle S, M, E, C_2.C \rangle$ ($v \neq 0$)
$\langle l.v.S, M, E, \text{store}.C \rangle$	\rightarrow	$\langle S, M\{l \mapsto v\}, E, C \rangle$
$\langle l.S, M, E, \text{load}.C \rangle$	\rightarrow	$\langle M(l).S, M, E, C \rangle$
$\langle 0.S, M, E, \text{loop}(C_1, C_2).C \rangle$	\rightarrow	$\langle S, M, E, C \rangle$
$\langle v.S, M, E, \text{loop}(C_1, C_2).C \rangle$	\rightarrow	$\langle S, M, E, C_2.C_1.\text{loop}(C_1, C_2).C \rangle$ ($v \neq 0$)
$\langle S, M, E, \text{bind}(x).C \rangle$	\rightarrow	$\langle S, M, (x, l).E, C \rangle$ ($l \notin \text{Dom } M$)
$\langle S, M, (x, l).E, \text{unbind}.C \rangle$	\rightarrow	$\langle S, M, E, C \rangle$