

4541.664A Homework 4

Semantics

Kihong Heo, Woosuk Lee, Wonchan Lee

1 C----

1.1 Big-step Operational Semantics

$$\begin{array}{lll} M & \in & \text{Memory} = \text{Var} \xrightarrow{\text{fin}} \text{Val} \\ v & \in & \text{Val} = \mathbb{Z} \end{array}$$

$$\boxed{M \vdash E \Rightarrow v}$$

$$\overline{M \vdash n \Rightarrow n}$$

$$\frac{M \vdash E_1 \Rightarrow v_1 \quad M \vdash E_2 \Rightarrow v_2}{M \vdash E_1 + E_2 \Rightarrow v_1 + v_2}$$

$$\frac{M \vdash E \Rightarrow v}{M \vdash \neg E \Rightarrow \neg v}$$

$$\boxed{M \vdash C \Rightarrow M'}$$

$$\overline{M \vdash \text{skip} \Rightarrow M}$$

$$\frac{M \vdash E \Rightarrow v}{M \vdash x := E \Rightarrow M\{x \mapsto v\}}$$

$$\frac{M \vdash C_1 \Rightarrow M' \quad M' \vdash C_2 \Rightarrow M''}{M \vdash C_1 ; C_2 \Rightarrow M''}$$

$$\frac{M \vdash E \Rightarrow 0 \quad M \vdash C_2 \Rightarrow M'}{M \vdash \text{if } E \ C_1 \ C_2 \Rightarrow M'}$$

$$\frac{M \vdash E \Rightarrow v \quad M \vdash C_1 \Rightarrow M'}{M \vdash \text{if } E \ C_1 \ C_2 \Rightarrow M'} \ v \neq 0$$

$$\frac{M \vdash E \Rightarrow 0}{M \vdash \text{while } E \ C \Rightarrow M}$$

$$\frac{M \vdash E \Rightarrow v \quad M \vdash C \Rightarrow M' \quad M' \vdash \text{while } E \ C \Rightarrow M''}{M \vdash \text{while } E \ C \Rightarrow M''} \ v \neq 0$$

1.2 Abstract Machine Semantics

Machine

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

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

Translation

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

Transition

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

2 C---

2.1 Big-step Operational Semantics

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

$\boxed{\sigma, M \vdash C \Rightarrow M'}$

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

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

$$\frac{\sigma, M \vdash C_1 \Rightarrow M' \quad \sigma, M' \vdash C_2 \Rightarrow M''}{\sigma, M \vdash C_1 ; C_2 \Rightarrow M''}$$

$$\frac{\sigma, M \vdash E \Rightarrow 0 \quad \sigma, M \vdash C_2 \Rightarrow M'}{\sigma, M \vdash \text{if } E \ C_1 \ C_2 \Rightarrow M'}$$

$$\frac{\sigma, M \vdash E \Rightarrow v \quad \sigma, M \vdash C_1 \Rightarrow M'}{\sigma, M \vdash \text{if } E \ C_1 \ C_2 \Rightarrow M'} \ v \neq 0$$

$$\frac{\sigma, M \vdash E \Rightarrow 0}{\sigma, M \vdash \text{while } E \ C \Rightarrow M}$$

$$\frac{\sigma, M \vdash E \Rightarrow v \quad \sigma, M \vdash C \Rightarrow M' \quad \sigma, M' \vdash \text{while } E \ C \Rightarrow M''}{\sigma, M \vdash \text{while } E \ C \Rightarrow M''} \ v \neq 0$$

$$\frac{\sigma, M \vdash E \Rightarrow v \quad \sigma\{x \mapsto l\}, M\{l \mapsto v\} \vdash C \Rightarrow M'}{\sigma, M \vdash \text{local } x := E \ \text{in } C \Rightarrow M'} \ l \notin \text{Dom } M$$

$\boxed{\sigma, M \vdash E \Rightarrow v}$

$$\overline{\sigma, M \vdash n \Rightarrow n}$$

$$\overline{\sigma, M \vdash x \Rightarrow M(\sigma(x))}$$

$$\frac{\sigma, M \vdash E_1 \Rightarrow v_1 \quad \sigma, M \vdash E_2 \Rightarrow v_2}{\sigma, M \vdash E_1 + E_2 \Rightarrow v_1 + v_2}$$

$$\frac{\sigma, M \vdash E \Rightarrow v}{\sigma, M \vdash -E \Rightarrow -v}$$

2.2 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

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

Transition

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

3 C--

3.1 Big-step Operational Semantics

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

$$\boxed{\sigma, M \vdash C \Rightarrow M'}$$

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

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

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

$$\frac{\sigma, M \vdash C_1 \Rightarrow M' \quad \sigma, M' \vdash C_2 \Rightarrow M''}{\sigma, M \vdash C_1 ; C_2 \Rightarrow M''}$$

$$\frac{\sigma, M \vdash E \Rightarrow 0 \quad \sigma, M \vdash C_2 \Rightarrow M'}{\sigma, M \vdash \text{if } E C_1 C_2 \Rightarrow M'}$$

$$\frac{\sigma, M \vdash E \Rightarrow v \quad \sigma, M \vdash C_1 \Rightarrow M'}{\sigma, M \vdash \text{if } E C_1 C_2 \Rightarrow M'} \quad v \neq 0$$

$$\frac{\sigma, M \vdash E \Rightarrow 0}{\sigma, M \vdash \text{while } E C \Rightarrow M}$$

$$\frac{\sigma, M \vdash E \Rightarrow v \quad \sigma, M \vdash C \Rightarrow M' \quad \sigma, M' \vdash \text{while } E C \Rightarrow M''}{\sigma, M \vdash \text{while } E C \Rightarrow M''} \quad v \neq 0$$

$$\frac{\sigma, M \vdash E \Rightarrow v \quad \sigma\{x \mapsto l\}, M\{l \mapsto v\} \vdash C \Rightarrow M'}{\sigma, M \vdash \text{local } x := E \text{ in } C \Rightarrow M'} \quad l \notin \text{Dom } M$$

$$\begin{array}{c}
\boxed{\sigma, M \vdash E \Rightarrow v} \\
\hline
\sigma, M \vdash n \Rightarrow n \\
\hline
\sigma, M \vdash x \Rightarrow M(\sigma(x)) \\
\hline
\frac{\sigma, M \vdash x \Rightarrow l}{\sigma, M \vdash *x \Rightarrow M(l)} \\
\hline
\sigma, M \vdash & \& x \Rightarrow \sigma(x) \\
\hline
\frac{\sigma, M \vdash E_1 \Rightarrow v_1 \quad \sigma, M \vdash E_2 \Rightarrow v_2}{\sigma, M \vdash E_1 + E_2 \Rightarrow v_1 + v_2} \\
\hline
\frac{\sigma, M \vdash E \Rightarrow v}{\sigma, M \vdash -E \Rightarrow -v}
\end{array}$$

3.2 Abstract Machine Semantics

Machine

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

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

Translation

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

Transition

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

4 C----

4.1 Big-step Operational Semantics

$$\begin{array}{llll}
M & \in & \text{Memory} & = \text{Var} \xrightarrow{\text{fin}} \text{Val} \\
v & \in & \text{Val} & = \{\cdot\} + \mathbb{Z} \\
\uparrow & \in & \text{Exn} & \\
r & \in & \text{Result} & = \text{Val} + \text{Exn}
\end{array}$$

$$\frac{\boxed{M \vdash C \Rightarrow r, M'}}{\overline{M \vdash \text{skip} \Rightarrow \cdot, M}}$$

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

$$\begin{array}{c}
\frac{M \vdash C_1 \Rightarrow \uparrow, M'}{M \vdash C_1 ; C_2 \Rightarrow \uparrow, M'} \\
\frac{M \vdash C_1 \Rightarrow v, M' \quad M' \vdash C_2 \Rightarrow r, M''}{M \vdash C_1 ; C_2 \Rightarrow r, M''} \\
\frac{M \vdash E \Rightarrow 0 \quad M \vdash C_2 \Rightarrow r, M'}{M \vdash \text{if } E C_1 C_2 \Rightarrow r, M'} \\
\frac{M \vdash E \Rightarrow v \quad M \vdash C_1 \Rightarrow r, M'}{M \vdash \text{if } E C_1 C_2 \Rightarrow r, M'} \quad v \neq 0 \\
\frac{M \vdash E \Rightarrow 0}{M \vdash \text{while } E C \Rightarrow \cdot, M} \\
\frac{M \vdash E \Rightarrow v \quad M \vdash C \Rightarrow v', M' \quad M' \vdash \text{while } E C \Rightarrow r, M''}{M \vdash \text{while } E C \Rightarrow r, M''} \quad v \neq 0 \\
\frac{M \vdash E \Rightarrow v \quad M \vdash C \Rightarrow \uparrow, M'}{M \vdash \text{while } E C \Rightarrow \uparrow, M'} \quad v \neq 0 \\
\frac{M \vdash C_1 \Rightarrow \uparrow, M' \quad M \vdash C_2 \Rightarrow r, M''}{M \vdash \text{try } C_1 \text{ handle } C_2 \Rightarrow r, M''} \\
\frac{M \vdash C_1 \Rightarrow v, M'}{M \vdash \text{try } C_1 \text{ handle } C_2 \Rightarrow v, M'} \\
\frac{}{M \vdash \text{raise} \Rightarrow \uparrow, M'} \\
\boxed{\sigma, M \vdash E \Rightarrow v} \\
\frac{}{M \vdash n \Rightarrow n} \\
\frac{}{M \vdash x \Rightarrow M(x)} \\
\frac{M \vdash E_1 \Rightarrow v_1 \quad M \vdash E_2 \Rightarrow v_2}{M \vdash E_1 + E_2 \Rightarrow v_1 + v_2} \\
\frac{M \vdash E \Rightarrow v}{M \vdash -E \Rightarrow -v}
\end{array}$$

4.2 Abstract Machine Semantics

Machine

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

S	$\rightarrow \epsilon$	(empty stack)
	$ n.S$	$(n \in \mathbb{Z})$
	$ l.S$	$(l \in Addr)$
C	$\rightarrow \epsilon$	(empty command)
	$ \text{push}(n).C$	
	$ \text{add}.C$	
	$ \text{neg}.C$	
	$ \text{jmpz}(C, C).C$	
	$ \text{store}(x).C$	
	$ \text{load}(x).C$	
	$ \text{loop}(C, C).C$	
	$ \text{sethndlr}(C, C).C$	
	$ \text{gethndlr}.C$	
	$ \text{freehndlr}.C$	
H	$\rightarrow \epsilon$	(empty handle stack)
	$ \langle S, C \rangle .H$	

Translation

$\llbracket x := E \rrbracket$	\rightarrow	$\llbracket E \rrbracket.\text{store}(x)$
$\llbracket \text{if } E \text{ then } C_1 \text{ else } C_2 \rrbracket$	\rightarrow	$\llbracket E \rrbracket.\text{jmpz}(\llbracket C_1 \rrbracket, \llbracket C_2 \rrbracket)$
$\llbracket \text{while } E C \rrbracket$	\rightarrow	$\llbracket E \rrbracket.\text{loop}(\llbracket E \rrbracket, \llbracket C \rrbracket)$
$\llbracket C_1 ; C_2 \rrbracket$	\rightarrow	$\llbracket C_1 \rrbracket.[\llbracket C_2 \rrbracket]$
$\llbracket \text{try } C_1 \text{ handle } C_2 \rrbracket$	\rightarrow	$\text{sethndlr}(\llbracket C_1 \rrbracket, \llbracket C_2 \rrbracket).\text{freehndlr}$
$\llbracket \text{raise} \rrbracket$	\rightarrow	gethndlr
$\llbracket n \rrbracket$	\rightarrow	$\text{push}(n)$
$\llbracket x \rrbracket$	\rightarrow	$\text{load}(x)$
$\llbracket E_1 + E_2 \rrbracket$	\rightarrow	$\llbracket E_1 \rrbracket.[\llbracket E_2 \rrbracket].\text{add}$
$\llbracket -E \rrbracket$	\rightarrow	$\llbracket E \rrbracket.\text{neg}$

Transition

$\langle S, M, \text{push}(n).C, H \rangle$	\rightarrow	$\langle n.S, M, C, H \rangle$
$\langle v_2.v_1.S, M, \text{add}.C, H \rangle$	\rightarrow	$\langle v.S, M, C, H \rangle$ $(v = v_1 + v_2)$
$\langle v.S, M, \text{neg}.C, H \rangle$	\rightarrow	$\langle -v.S, M, C, H \rangle$
$\langle 0.S, M, \text{jmpz}(C_1, C_2).C, H \rangle$	\rightarrow	$\langle S, M, C_1.C, H \rangle$
$\langle v.S, M, \text{jmpz}(C_1, C_2).C, H \rangle$	\rightarrow	$\langle S, M, C_2.C, H \rangle$ $(v \neq 0)$
$\langle v.S, M, \text{store}(x).C, H \rangle$	\rightarrow	$\langle S, M\{x \mapsto v\}, C, H \rangle$
$\langle l.S, M, \text{load}(x).C, H \rangle$	\rightarrow	$\langle M(x).S, M, C, H \rangle$
$\langle 0.S, M, \text{loop}(C_1, C_2).C, H \rangle$	\rightarrow	$\langle S, M, C, H \rangle$
$\langle v.S, M, \text{loop}(C_1, C_2).C, H \rangle$	\rightarrow	$\langle S, M, C_2.C_1.\text{loop}(C_1, C_2).C, H \rangle$ $(v \neq 0)$
$\langle S, M, \text{sethndlr}(C_1, C_2).C, H \rangle$	\rightarrow	$\langle S, M, C_1.C, \langle S, C_2.C \rangle :: H \rangle$
$\langle S, M, \text{gethndlr}.C, \langle S', C' \rangle :: H \rangle$	\rightarrow	$\langle S', M, C', H \rangle$
$\langle S, M, \text{freehndlr}.C, \langle S', C' \rangle :: H \rangle$	\rightarrow	$\langle S, M, C, H \rangle$