Homework 1

1 Division by Zero

Big Step Semantics

$\text{divzero}((a_1 \Downarrow i_1) \,/\, a_2, \sigma)$ is a configuration that capture the error information when dividing by zero. It says that the state before the division is σ, the numerator is a_1 that evaluates to i_1, and the denominator is a_2 which evaluates to 0 that causes the problem.

Arithmetic

$$\langle i, \sigma \rangle \Downarrow \langle i \rangle \qquad (\textsc{BigStep-Int})$$

$$\langle x, \sigma \rangle \Downarrow \langle \sigma(x) \rangle \quad \text{if } \sigma(x) \neq \bot \qquad (\textsc{BigStep-Lookup})$$

$$\frac{\langle a_1, \sigma \rangle \Downarrow \langle i_1 \rangle \quad \langle a_2, \sigma \rangle \Downarrow \langle i_2 \rangle}{\langle a_1 + a_2, \sigma \rangle \Downarrow \langle i_1 +_{\text{Int}} i_2 \rangle} \qquad (\textsc{BigStep-Add})$$

$$\frac{\langle a_1, \sigma \rangle \Downarrow \langle \text{divzero}(e) \rangle}{\langle a_1 + a_2, \sigma \rangle \Downarrow \langle \text{divzero}(e) \rangle} \qquad \boxed{\textsc{BigStep-AddHalt1}}$$

$$\frac{\langle a_1, \sigma \rangle \Downarrow \langle i_1 \rangle \quad \langle a_2, \sigma \rangle \Downarrow \langle \text{divzero}(e) \rangle}{\langle a_1 + a_2, \sigma \rangle \Downarrow \langle \text{divzero}(e) \rangle} \qquad \boxed{\textsc{BigStep-AddHalt2}}$$

$$\frac{\langle a_1, \sigma \rangle \Downarrow \langle i_1 \rangle \quad \langle a_2, \sigma \rangle \Downarrow \langle i_2 \rangle}{\langle a_1 \,/\, a_2, \sigma \rangle \Downarrow \langle i_1 \,/_{\text{Int}}\, i_2 \rangle} \quad \text{if } i_2 \neq 0 \qquad (\textsc{BigStep-Div})$$

$$\frac{\langle a_1, \sigma \rangle \Downarrow \langle \text{divzero}(e) \rangle}{\langle a_1 \,/\, a_2, \sigma \rangle \Downarrow \langle \text{divzero}(e) \rangle} \quad \text{if } i_2 \neq 0 \qquad \boxed{\textsc{BigStep-DivHalt1}}$$

$$\frac{\langle a_1, \sigma \rangle \Downarrow \langle i_1 \rangle \quad \langle a_2, \sigma \rangle \Downarrow \langle \text{divzero}(e) \rangle}{\langle a_1 \,/\, a_2, \sigma \rangle \Downarrow \langle \text{divzero}(e) \rangle} \quad \text{if } i_2 \neq 0 \qquad \boxed{\textsc{BigStep-DivHalt2}}$$

$$\frac{\langle a_1, \sigma \rangle \Downarrow \langle i_1 \rangle \quad \langle a_2, \sigma \rangle \Downarrow \langle 0 \rangle}{\langle a_1 \,/\, a_2, \sigma \rangle \Downarrow \langle \text{divzero}((a_1 \Downarrow i_1) \,/\, a_2, \sigma) \rangle} \qquad \boxed{\textsc{BigStep-Divzero}}$$

Boolean

$$\langle t, \sigma \rangle \Downarrow \langle t \rangle \qquad \text{(BigStep-Bool)}$$

$$\frac{\langle a_1, \sigma \rangle \Downarrow \langle i_1 \rangle \quad \langle a_2, \sigma \rangle \Downarrow \langle i_2 \rangle}{\langle a_1 \texttt{ <= } a_2, \sigma \rangle \Downarrow \langle i_1 \leq_{\text{Int}} i_2 \rangle} \qquad \text{(BigStep-Leq)}$$

$$\frac{\langle a_1, \sigma \rangle \Downarrow \langle \text{divzero}(e) \rangle}{\langle a_1 \texttt{ <= } a_2, \sigma \rangle \Downarrow \langle \text{divzero}(e) \rangle} \qquad \boxed{\text{BigStep-LeqHalt1}}$$

$$\frac{\langle a_1, \sigma \rangle \Downarrow \langle i_1 \rangle \quad \langle a_2, \sigma \rangle \Downarrow \langle \text{divzero}(e) \rangle}{\langle a_1 \texttt{ <= } a_2, \sigma \rangle \Downarrow \langle \text{divzero}(e) \rangle} \qquad \boxed{\text{BigStep-LeqHalt2}}$$

$$\frac{\langle b, \sigma \rangle \Downarrow \langle \texttt{true} \rangle}{\langle \texttt{!}\, b, \sigma \rangle \Downarrow \langle \texttt{false} \rangle} \qquad \text{(BigStep-Not-True)}$$

$$\frac{\langle b, \sigma \rangle \Downarrow \langle \text{divzero}(e) \rangle}{\langle \texttt{!}\, b, \sigma \rangle \Downarrow \langle \text{divzero}(e) \rangle} \qquad \boxed{\text{BigStep-Not-TrueHalt}}$$

$$\frac{\langle b, \sigma \rangle \Downarrow \langle \texttt{true} \rangle}{\langle \texttt{!}\, b, \sigma \rangle \Downarrow \langle \texttt{false} \rangle} \qquad \text{(BigStep-Not-False)}$$

$$\frac{\langle b, \sigma \rangle \Downarrow \langle \text{divzero}(e) \rangle}{\langle \texttt{!}\, b, \sigma \rangle \Downarrow \langle \text{divzero}(e) \rangle} \qquad \boxed{\text{BigStep-Not-FalseHalt}}$$

$$\frac{\langle b_1, \sigma \rangle \Downarrow \langle \texttt{false} \rangle}{\langle b_1 \,\texttt{\&\&}\, b_2, \sigma \rangle \Downarrow \langle \texttt{false} \rangle} \qquad \text{BigStep-And-False)}$$

$$\frac{\langle b_1, \sigma \rangle \Downarrow \langle \text{divzero}(e) \rangle}{\langle b_1 \,\texttt{\&\&}\, b_2, \sigma \rangle \Downarrow \langle \text{divzero}(e) \rangle} \qquad \boxed{\text{BigStep-And-FalseHalt}}$$

$$\frac{\langle b_1, \sigma \rangle \Downarrow \langle \texttt{true} \rangle \quad \langle b_2, \sigma \rangle \Downarrow \langle t \rangle}{\langle b_1 \,\texttt{\&\&}\, b_2, \sigma \rangle \Downarrow \langle t \rangle} \qquad \text{(BigStep-And-True)}$$

$$\frac{\langle b_1, \sigma \rangle \Downarrow \langle \texttt{true} \rangle \quad \langle b_2, \sigma \rangle \Downarrow \langle \text{divzero}(e) \rangle}{\langle b_1 \,\texttt{\&\&}\, b_2, \sigma \rangle \Downarrow \langle \text{divzero}(e) \rangle} \qquad \boxed{\text{BigStep-And-TrueHalt}}$$

Control Flow

$$\langle \{\}, \sigma\rangle \Downarrow \langle \sigma \rangle \qquad \text{(BigStep-Empty-Block)}$$

$$\frac{\langle s, \sigma\rangle \Downarrow \langle \sigma' \rangle}{\langle \texttt{\{}\ s\ \texttt{\}}, \sigma\rangle \Downarrow \langle \sigma' \rangle} \qquad \text{(BigStep-Block)}$$

$$\frac{\langle s, \sigma\rangle \Downarrow \langle \text{divzero}(e) \rangle}{\langle \texttt{\{}\ s\ \texttt{\}}, \sigma\rangle \Downarrow \langle \text{divzero}(e) \rangle} \qquad \boxed{\text{BigStep-BlockHalt}}$$

$$\frac{\langle a, \sigma\rangle \Downarrow \langle i \rangle}{\langle x \texttt{=} a\texttt{;}, \sigma\rangle \Downarrow \langle \sigma[i/x] \rangle} \qquad \text{(BigStep-Asgn)}$$

$$\frac{\langle a, \sigma\rangle \Downarrow \langle \text{divzero}(e) \rangle}{\langle x \texttt{=} a\texttt{;}, \sigma\rangle \Downarrow \langle \text{divzero}(e) \rangle} \qquad \boxed{\text{BigStep-AsgnHalt}}$$

$$\frac{\langle s_1, \sigma\rangle \Downarrow \langle \sigma_1 \rangle \quad \langle s_2, \sigma_1\rangle \Downarrow \langle \sigma_2 \rangle}{\langle s_1\ s_2, \sigma\rangle \Downarrow \langle \sigma_2 \rangle} \qquad \text{(BigStep-Seq)}$$

$$\frac{\langle s_1, \sigma\rangle \Downarrow \langle \text{divzero}(e) \rangle}{\langle s_1\ s_2, \sigma\rangle \Downarrow \langle \text{divzero}(e) \rangle} \qquad \boxed{\text{BigStep-SeqHalt1}}$$

$$\frac{\langle s_1, \sigma\rangle \Downarrow \langle \sigma_1 \rangle \quad \langle s_2, \sigma_1\rangle \Downarrow \langle \text{divzero}(e) \rangle}{\langle s_1\ s_2, \sigma\rangle \Downarrow \langle \text{divzero}(e) \rangle} \qquad \boxed{\text{BigStep-SeqHalt2}}$$

$$\frac{\langle b, \sigma\rangle \Downarrow \langle \texttt{true} \rangle \quad \langle s_1, \sigma\rangle \Downarrow \langle \sigma_1 \rangle}{\langle \texttt{if}\ (b)\ s_1\ \texttt{else}\ s_2, \sigma\rangle \Downarrow \langle \sigma_1 \rangle} \qquad \text{(BigStep-If-True)}$$

$$\frac{\langle b, \sigma\rangle \Downarrow \langle \text{divzero}(e) \rangle}{\langle \texttt{if}\ (b)\ s_1\ \texttt{else}\ s_2, \sigma\rangle \Downarrow \langle \text{divzero}(e) \rangle} \qquad \boxed{\text{BigStep-If-TrueHalt1}}$$

$$\frac{\langle b, \sigma\rangle \Downarrow \langle \texttt{true} \rangle \quad \langle s_1, \sigma\rangle \Downarrow \langle \text{divzero}(e) \rangle}{\langle \texttt{if}\ (b)\ s_1\ \texttt{else}\ s_2, \sigma\rangle \Downarrow \langle \text{divzero}(e) \rangle} \qquad \boxed{\text{BigStep-If-TrueHalt2}}$$

$$\frac{\langle b, \sigma\rangle \Downarrow \langle \texttt{false} \rangle \quad \langle s_2, \sigma\rangle \Downarrow \langle \sigma_2 \rangle}{\langle \texttt{if}\ (b)\ s_1\ \texttt{else}\ s_2, \sigma\rangle \Downarrow \langle \sigma_2 \rangle} \qquad \text{(BigStep-If-False)}$$

$$\frac{\langle b, \sigma\rangle \Downarrow \langle \texttt{false} \rangle \quad \langle s_2, \sigma\rangle \Downarrow \langle \text{divzero}(e) \rangle}{\langle \texttt{if}\ (b)\ s_1\ \texttt{else}\ s_2, \sigma\rangle \Downarrow \langle \text{divzero}(e) \rangle} \qquad \boxed{\text{BigStep-If-FalseHalt2}}$$

$$\frac{\langle b, \sigma\rangle \Downarrow \langle \texttt{false} \rangle}{\langle \texttt{while}\ (b)\ s, \sigma\rangle \Downarrow \langle \sigma \rangle} \qquad \text{(BigStep-While-False)}$$

$$\frac{\langle b, \sigma\rangle \Downarrow \langle \text{divzero}(e) \rangle}{\langle \texttt{while}\ (b)\ s, \sigma\rangle \Downarrow \langle \text{divzero}(e) \rangle} \qquad \boxed{\text{BigStep-While-FalseHalt}}$$

$$\frac{\langle b, \sigma\rangle \Downarrow \langle \texttt{true} \rangle \quad \langle s\ \texttt{while}\ (b)\ s, \sigma\rangle \Downarrow \langle \sigma' \rangle}{\langle \texttt{while}\ (b)\ s, \sigma\rangle \Downarrow \langle \sigma' \rangle} \qquad \text{(BigStep-While-True)}$$

$$\frac{\langle b, \sigma\rangle \Downarrow \langle \text{divzero}(e) \rangle}{\langle \texttt{while}\ (b)\ s, \sigma\rangle \Downarrow \langle \text{divzero}(e) \rangle} \qquad \boxed{\text{BigStep-While-TrueHalt1}}$$

$$\frac{\langle b, \sigma\rangle \Downarrow \langle \texttt{true} \rangle \quad \langle s\ \texttt{while}\ (b)\ s, \sigma\rangle \Downarrow \langle \text{divzero}(e) \rangle}{\langle \texttt{while}\ (b)\ s, \sigma\rangle \Downarrow \langle \text{divzero}(e) \rangle} \qquad \boxed{\text{BigStep-While-TrueHalt2}}$$

Small Step Semantics

$\mathrm{divzero}(a_1 \,/\, a_2, \sigma)$ is a configuration that capture the error information when dividing by zero. It says that the state before the division is σ, the numerator is a_1, and the denominator is a_2 which evaluates to 0 that causes the problem.

$$\langle x, \sigma \rangle \to \langle \sigma(x), \sigma \rangle \quad \text{if } \sigma(x) \neq \bot \qquad (\textsc{SmallStep-Lookup})$$

$$\frac{\langle a_1, \sigma \rangle \to \langle a_1', \sigma \rangle}{\langle a_1 + a_2, \sigma \rangle \to \langle a_1' + a_2, \sigma \rangle} \qquad (\textsc{SmallStep-Add-Arg1})$$

$$\frac{\langle a_1, \sigma \rangle \to \langle \mathrm{divzero}(e) \rangle}{\langle a_1 + a_2, \sigma \rangle \to \langle \mathrm{divzero}(e) \rangle} \qquad (\textsc{SmallStep-Add-Arg1-Halt})$$

$$\frac{\langle a_2, \sigma \rangle \to \langle a_2', \sigma \rangle}{\langle a_1 + a_2, \sigma \rangle \to \langle a_1 + a_2', \sigma \rangle} \qquad (\textsc{SmallStep-Add-Arg2})$$

$$\frac{\langle a_2, \sigma \rangle \to \langle \mathrm{divzero}(e) \rangle}{\langle a_1 + a_2, \sigma \rangle \to \langle \mathrm{divzero}(e) \rangle} \qquad (\textsc{SmallStep-Add-Arg2-Halt})$$

$$\langle i_1 + i_2, \sigma \rangle \to \langle i_1 +_{\mathrm{Int}} i_2, \sigma \rangle \qquad (\textsc{SmallStep-Add})$$

$$\frac{\langle a_1, \sigma \rangle \to \langle a_1', \sigma \rangle}{\langle a_1 \,/\, a_2, \sigma \rangle \to \langle a_1' \,/\, a_2, \sigma \rangle} \qquad (\textsc{SmallStep-Div-Arg1})$$

$$\frac{\langle a_1, \sigma \rangle \to \langle \mathrm{divzero}(e) \rangle}{\langle a_1 \,/\, a_2, \sigma \rangle \to \langle \mathrm{divzero}(e) \rangle} \qquad (\textsc{SmallStep-Div-Arg1-Halt})$$

$$\frac{\langle a_2, \sigma \rangle \to \langle a_2', \sigma \rangle}{\langle a_1 \,/\, a_2, \sigma \rangle \to \langle a_1 \,/\, a_2', \sigma \rangle} \qquad (\textsc{SmallStep-Div-Arg2})$$

$$\frac{\langle a_2, \sigma \rangle \to \langle \mathrm{divzero}(e) \rangle}{\langle a_1 \,/\, a_2, \sigma \rangle \to \langle \mathrm{divzero}(e) \rangle} \qquad (\textsc{SmallStep-Div-Arg2-Halt})$$

$$\langle i_1 \,/\, i_2, \sigma \rangle \to \langle i_1 \,/_{\mathrm{Int}}\, i_2, \sigma \rangle \quad \text{if } i_2 \neq 0 \qquad (\textsc{SmallStep-Div})$$

$$\frac{\langle a_2, \sigma \rangle \to \langle 0, \sigma \rangle}{\langle a_1 \,/\, a_2, \sigma \rangle \to \langle \mathrm{divzero}(a_1 \,/\, a_2, \sigma) \rangle} \qquad (\textsc{SmallStep-Divzero})$$

$$\frac{\langle a_1, \sigma \rangle \to \langle a_1', \sigma \rangle}{\langle a_1 \texttt{<=} a_2, \sigma \rangle \to \langle a_1' \texttt{<=} a_2, \sigma \rangle} \qquad (\textsc{SmallStep-Leq-Arg1})$$

$$\frac{\langle a_1, \sigma \rangle \to \langle \mathrm{divzero}(e) \rangle}{\langle a_1 \texttt{<=} a_2, \sigma \rangle \to \langle \mathrm{divzero}(e) \rangle} \qquad (\textsc{SmallStep-Leq-Arg1-Halt})$$

$$\frac{\langle a_2, \sigma \rangle \to \langle a_2', \sigma \rangle}{\langle a_1 \texttt{<=} a_2, \sigma \rangle \to \langle a_1 \texttt{<=} a_2, \sigma \rangle} \qquad (\textsc{SmallStep-Leq-Arg2})$$

$$\frac{\langle a_2, \sigma \rangle \to \langle \mathrm{divzero}(e) \rangle}{\langle a_1 \texttt{<=} a_2, \sigma \rangle \to \langle \mathrm{divzero}(e) \rangle} \qquad (\textsc{SmallStep-Leq-Arg2-Halt})$$

$$\langle i_1 \texttt{<=} i_2, \sigma \rangle \to \langle i_1 \leq_{\mathrm{Int}} i_2, \sigma \rangle \qquad (\textsc{SmallStep-Leq})$$

$$\frac{\langle b,\sigma\rangle \to \langle b',\sigma\rangle}{\langle \texttt{!}\, b,\sigma\rangle \to \langle \texttt{!}\, b',\sigma\rangle} \qquad (\textsc{SmallStep-Not-Arg})$$

$$\frac{\langle b,\sigma\rangle \to \langle \text{divzero}(e)\rangle}{\langle \texttt{!}\, b,\sigma\rangle \to \langle \text{divzero}(e)\rangle} \qquad (\textsc{SmallStep-Not-ArgHalt})$$

$$\langle \texttt{!}\,\texttt{true},\sigma\rangle \to \langle \texttt{false},\sigma\rangle \qquad (\textsc{SmallStep-Not-True})$$

$$\langle \texttt{!}\,\texttt{false},\sigma\rangle \to \langle \texttt{true},\sigma\rangle \qquad (\textsc{SmallStep-Not-False})$$

$$\frac{\langle b_1,\sigma\rangle \to \langle b_1',\sigma\rangle}{\langle b_1 \,\texttt{\&\&}\, b_2,\sigma\rangle \to \langle b_1' \,\texttt{\&\&}\, b_2,\sigma\rangle} \qquad (\textsc{SmallStep-And-Arg1})$$

$$\frac{\langle b_1,\sigma\rangle \to \langle \text{divzero}(e)\rangle}{\langle b_1 \,\texttt{\&\&}\, b_2,\sigma\rangle \to \langle \text{divzero}(e)\rangle} \qquad (\textsc{SmallStep-And-Arg1-Halt})$$

$$\langle \texttt{false} \,\texttt{\&\&}\, b_2,\sigma\rangle \to \langle \texttt{false},\sigma\rangle \qquad (\textsc{SmallStep-And-False})$$

$$\langle \texttt{true} \,\texttt{\&\&}\, b_2,\sigma\rangle \to \langle b_2,\sigma\rangle \qquad (\textsc{SmallStep-And-True})$$

$$\langle \texttt{\{}\, s\, \texttt{\}},\sigma\rangle \to \langle s,\sigma\rangle \qquad (\textsc{SmallStep-Block})$$

$$\frac{\langle a,\sigma\rangle \to \langle a',\sigma\rangle}{\langle x\,\texttt{=}\,a\,\texttt{;},\sigma\rangle \to \langle x\,\texttt{=}\,a'\,\texttt{;},\sigma\rangle} \qquad (\textsc{SmallStep-Asgn-Arg2})$$

$$\frac{\langle a,\sigma\rangle \to \langle \text{divzero}(e)\rangle}{\langle x\,\texttt{=}\,a\,\texttt{;},\sigma\rangle \to \langle \text{divzero}(e)\rangle} \qquad (\textsc{SmallStep-Asgn-Arg2-Halt})$$

$$\langle x\,\texttt{=}\,i\,\texttt{;},\sigma\rangle \to \langle \texttt{\{\}},\sigma[i/x]\rangle \quad \text{if } \sigma(x) \neq \bot \qquad (\textsc{SmallStep-Asgn})$$

$$\frac{\langle s_1,\sigma\rangle \to \langle s_1',\sigma\rangle}{\langle s_1\ s_2,\sigma\rangle \to \langle s_1'\ s_2,\sigma\rangle} \qquad (\textsc{SmallStep-Seq-Arg1})$$

$$\frac{\langle s_1,\sigma\rangle \to \langle \text{divzero}(e)\rangle}{\langle s_1\ s_2,\sigma\rangle \to \langle \text{divzero}(e)\rangle} \qquad (\textsc{SmallStep-Seq-Arg1-Halt})$$

$$\langle \texttt{\{\}}\ \ s_2\rangle \to \langle s_2,\sigma\rangle \qquad (\textsc{SmallStep-Seq-Empty-Block})$$

$$\frac{\langle b,\sigma\rangle \to \langle b',\sigma\rangle}{\langle \texttt{if}\ (b)\ s_1\ \texttt{else}\ \ s_2,\sigma\rangle \to \langle \texttt{if}\ (b')\ s_1\ \texttt{else}\ \ s_2,\sigma\rangle} \qquad (\textsc{SmallStep-If-Arg1})$$

$$\frac{\langle b,\sigma\rangle \to \langle \text{divzero}(e)\rangle}{\langle \texttt{if}\ (b)\ s_1\ \texttt{else}\ \ s_2,\sigma\rangle \to \langle \text{divzero}(e)\rangle} \qquad (\textsc{SmallStep-If-Arg1-Halt})$$

$$\langle \texttt{if (true)}\ s_1\ \texttt{else}\ s_2,\sigma\rangle \to \langle s_1,\sigma\rangle \qquad (\textsc{SmallStep-If-True})$$

$$\langle \texttt{if (false)}\ s_1\ \texttt{else}\ s_2,\sigma\rangle \to \langle s_2,\sigma\rangle \qquad (\textsc{SmallStep-If-False})$$

$$\langle \texttt{while (}b\texttt{)}\ s,\sigma\rangle \to \langle \texttt{if (}b\texttt{) \{}\ s\ \texttt{while (}b\texttt{)}\ s\ \texttt{\} else \{\}},\sigma\rangle \qquad (\textsc{SmallStep-While})$$

$$\langle \texttt{int}\ xl\texttt{;}\ s\rangle \to \langle s, xl \mapsto 0\rangle \qquad (\textsc{SmallStep-Var})$$

2 Increment Construct

It's assumed that the **operator priority has already been taken care of by lexer and parser**, so semantics will not bother considering the priority of operators.

BNF

```
Int   ::= the domain of integers with integer operations
Bool  ::= the domain of booleans
Id    ::= standard identifiers
AExp  ::= Int
        | Id
        | ++ Id                            <-- Added Semantics
        | AExp + AExp
        | AExp / AExp
BExp  ::= Bool
        | AExp <= AExp
        | ! BExp
        | BExp && BExp
Block ::= { }
        | { Stmt }
Stmt  ::= Block
        | Id = AExp ;
        | Stmt Stmt
        | if ( BExp ) Block else Block;
        | while ( BExp ) Block;
Pgm   ::= int List{Id} ; Stmt
```

Big Step Semantics

The biggest problem is that previously we treat arithmetic and boolean operation *pure*, so the operation only yield a integer or boolean result without state. To add the increment semantics, we must also include the updated states in the boolean and arithmetic operations.

Also, since the increment semantic is *impure*, if we want our semantic **consistent**, we have to enforce a **strict sequential order** for all binary operations that are previously non-deterministic. Otherwise the code like `(++x) + x` might yield different result if follow different derivation tree. We could choose the order **from left to right**. (Technically, you cannot really enforce an order in big step semantics, the trick here is to make the right operand evaluation uses the updated state from the evaluation results of left operand)

$$\langle x, \sigma\rangle \Downarrow \langle \sigma(x), \sigma\rangle \quad \text{if } \sigma(x) \neq \bot \qquad (\textsc{BigStep-Lookup})$$

$$\frac{\langle a_1, \sigma\rangle \Downarrow \langle i_1, \sigma_1\rangle \quad \langle a_2, \sigma_1\rangle \Downarrow \langle i_2, \sigma_2\rangle}{\langle a_1 \texttt{+} a_2, \sigma\rangle \Downarrow \langle i_1 +_{\text{Int}} i_2, \sigma_2\rangle} \qquad (\textsc{BigStep-Add})$$

$$\frac{\langle a_1, \sigma\rangle \Downarrow \langle i_1, \sigma_1\rangle \quad \langle a_2, \sigma_1\rangle \Downarrow \langle i_2, \sigma_2\rangle}{\langle a_1 \texttt{/} a_2, \sigma\rangle \Downarrow \langle i_1 /_{\text{Int}} i_2, \sigma_2\rangle} \quad \text{if } i_2 \neq 0 \qquad (\textsc{BigStep-Div})$$

$$\frac{\langle x, \sigma\rangle \Downarrow \langle i, \sigma_1\rangle}{\langle \texttt{++}\, x, \sigma\rangle \Downarrow \langle i +_{\text{Int}} 1, \sigma_1[i +_{\text{Int}} 1/x]\rangle} \quad \text{if } \sigma_1(x) \neq \bot \qquad (\textsc{BigStep-Increment})$$

$$\frac{\langle a_1, \sigma\rangle \Downarrow \langle i_1, \sigma_1\rangle \quad \langle a_2, \sigma_1\rangle \Downarrow \langle i_2, \sigma_2\rangle}{\langle a_1 \texttt{<=}\, a_2, \sigma\rangle \Downarrow \langle i_1 \leq_{\text{Int}} i_2, \sigma_2\rangle} \qquad (\textsc{BigStep-Leq})$$

$$\frac{\langle b, \sigma\rangle \Downarrow \langle \texttt{true}, \sigma_1\rangle}{\langle \texttt{!}\, b, \sigma\rangle \Downarrow \langle \texttt{false}, \sigma_1\rangle} \qquad (\textsc{BigStep-Not-True})$$

$$\frac{\langle b, \sigma\rangle \Downarrow \langle \texttt{false}, \sigma_1\rangle}{\langle \texttt{!}\, b, \sigma\rangle \Downarrow \langle \texttt{true}, \sigma_1\rangle} \qquad (\textsc{BigStep-Not-False})$$

$$\frac{\langle b_1, \sigma\rangle \Downarrow \langle \texttt{false}, \sigma_1\rangle}{\langle b_1 \,\texttt{\&\&}\, b_2, \sigma\rangle \Downarrow \langle \texttt{false}, \sigma_1\rangle} \qquad (\textsc{BigStep-And-False})$$

$$\frac{\langle b_1, \sigma\rangle \Downarrow \langle \texttt{true}, \sigma_1\rangle \quad \langle b_2, \sigma_1\rangle \Downarrow \langle t, \sigma_2\rangle}{\langle b_1 \,\texttt{\&\&}\, b_2, \sigma\rangle \Downarrow \langle t, \sigma_2\rangle} \qquad (\textsc{BigStep-And-True})$$

$$\langle \texttt{\{\}}, \sigma\rangle \Downarrow \langle \sigma\rangle \qquad (\textsc{BigStep-Empty-Block})$$

$$\frac{\langle s, \sigma\rangle \Downarrow \langle \sigma_1\rangle}{\langle \texttt{\{}\, s\, \texttt{\}}, \sigma\rangle \Downarrow \langle \sigma_1\rangle} \qquad (\textsc{BigStep-Block})$$

$$\frac{\langle a, \sigma\rangle \Downarrow \langle i, \sigma_1\rangle}{\langle x \texttt{=} a\texttt{;}, \sigma\rangle \Downarrow \langle \sigma_1[i/x]\rangle} \quad \text{if } \sigma_1(x) \neq \bot \qquad (\textsc{BigStep-Asgn})$$

$$\frac{\langle s_1, \sigma\rangle \Downarrow \langle \sigma_1\rangle \quad \langle s_2, \sigma_1\rangle \Downarrow \langle \sigma_2\rangle}{\langle s_1\ s_2, \sigma\rangle \Downarrow \langle \sigma_2\rangle} \qquad (\textsc{BigStep-Seq})$$

$$\frac{\langle b, \sigma\rangle \Downarrow \langle \texttt{true}, \sigma_1\rangle \quad \langle s_1, \sigma_1\rangle \Downarrow \langle \sigma_2\rangle}{\langle \texttt{if (}b\texttt{)}\ s_1\ \texttt{else}\ s_2, \sigma\rangle \Downarrow \langle \sigma_2\rangle} \qquad (\textsc{BigStep-If-True})$$

$$\frac{\langle b, \sigma\rangle \Downarrow \langle \texttt{false}, \sigma_1\rangle \quad \langle s_2, \sigma_1\rangle \Downarrow \langle \sigma_2\rangle}{\langle \texttt{if (}b\texttt{)}\ s_1\ \texttt{else}\ s_2, \sigma\rangle \Downarrow \langle \sigma_2\rangle} \qquad (\textsc{BigStep-If-False})$$

$$\frac{\langle b, \sigma\rangle \Downarrow \langle \texttt{false}, \sigma_1\rangle}{\langle \texttt{while (}b\texttt{)}\ s, \sigma\rangle \Downarrow \langle \sigma_1\rangle} \qquad (\textsc{BigStep-While-False})$$

$$\frac{\langle b, \sigma\rangle \Downarrow \langle \texttt{true}, \sigma_1\rangle \quad \langle s\ \texttt{while (}b\texttt{)}\ s, \sigma_1\rangle \Downarrow \langle \sigma_2\rangle}{\langle \texttt{while (}b\texttt{)}\ s, \sigma\rangle \Downarrow \langle \sigma_2\rangle} \qquad (\textsc{BigStep-While-True})$$

Simply put, we just enforce the sequential order, include the state change, add one new rule. Technically, the premises in the increment rule could simply be $\langle x, \sigma\rangle \Downarrow \langle i, \sigma\rangle$, as the rule that could apply here is variable lookup, and that doesn't change the state.

Type Systems

Type system only concerns about the type of variables. In this case, increment an integer still results an integer, and there aren't new information to store. Therefore, we only need one new rule:

$$\frac{xl \vdash a : \text{int}}{xl \vdash \texttt{++}\, a : \text{int}} \quad (\textsc{BigStepTypeSystem-Increment})$$

Small Step Semantics

Like Big Step Semantics, we have to enforce strict sequential order and update the states.

$$\langle x, \sigma \rangle \to \langle \sigma(x), \sigma \rangle \quad \text{if } \sigma(x) \neq \bot \qquad (\textsc{SmallStep-Lookup})$$

$$\frac{\langle a_1, \sigma \rangle \to \langle a_1', \sigma_1 \rangle}{\langle a_1 \texttt{+} a_2, \sigma \rangle \to \langle a_1' \texttt{+} a_2, \sigma_1 \rangle} \qquad (\textsc{SmallStep-Add-Arg1})$$

$$\frac{\langle a_2, \sigma \rangle \to \langle a_2', \sigma_1 \rangle}{\langle i_1 \texttt{+} a_2, \sigma \rangle \to \langle i_1 \texttt{+} a_2', \sigma_1 \rangle} \qquad (\textsc{SmallStep-Add-Arg2})$$

$$\langle i_1 \texttt{+} i_2, \sigma \rangle \to \langle i_1 +_{\text{Int}} i_2, \sigma \rangle \qquad (\textsc{SmallStep-Add})$$

$$\frac{\langle a_1, \sigma \rangle \to \langle a_1', \sigma_1 \rangle}{\langle a_1 \,\texttt{/}\, a_2, \sigma \rangle \to \langle a_1' \,\texttt{/}\, a_2, \sigma_1 \rangle} \qquad (\textsc{SmallStep-Div-Arg1})$$

$$\frac{\langle a_2, \sigma \rangle \to \langle a_2', \sigma_1 \rangle}{\langle i_1 \,\texttt{/}\, a_2, \sigma \rangle \to \langle i_1 \,\texttt{/}\, a_2', \sigma_1 \rangle} \qquad (\textsc{SmallStep-Div-Arg2})$$

$$\langle i_1 \,\texttt{/}\, i_2, \sigma \rangle \to \langle i_1 /_{\text{Int}} i_2, \sigma \rangle \quad \text{if } i_2 \neq 0 \qquad (\textsc{SmallStep-Div})$$

$$\frac{\langle x, \sigma \rangle \to \langle i, \sigma_1 \rangle}{\langle \texttt{++}\, x, \sigma \rangle \to \langle i +_{\text{Int}} 1, \sigma_1[i +_{\text{Int}} 1/x] \rangle} \quad \text{if } \sigma_1(x) \neq \bot \qquad (\textsc{SmallStep-Increment})$$

$$\frac{\langle a_1, \sigma \rangle \to \langle a_1', \sigma_1 \rangle}{\langle a_1 \texttt{<=} a_2, \sigma \rangle \to \langle a_1' \texttt{<=} a_2, \sigma_1 \rangle} \qquad (\textsc{SmallStep-Leq-Arg1})$$

$$\frac{\langle a_2, \sigma \rangle \to \langle a_2', \sigma_1 \rangle}{\langle i_1 \texttt{<=} a_2, \sigma \rangle \to \langle i_1 \texttt{<=} a_2, \sigma_1 \rangle} \qquad (\textsc{SmallStep-Leq-Arg2})$$

$$\langle i_1 \texttt{<=} i_2, \sigma \rangle \to \langle i_1 \leq_{\text{Int}} i_2, \sigma \rangle \qquad (\textsc{SmallStep-Leq})$$

$$\frac{\langle b, \sigma \rangle \to \langle b', \sigma_1 \rangle}{\langle \texttt{!}\, b, \sigma \rangle \to \langle \texttt{!}\, b', \sigma_1 \rangle} \qquad (\textsc{SmallStep-Not-Arg})$$

$$\langle \texttt{!}\, \texttt{true}, \sigma \rangle \to \langle \texttt{false}, \sigma \rangle \qquad (\textsc{SmallStep-Not-True})$$

$$\langle \texttt{!}\, \texttt{false}, \sigma \rangle \to \langle \texttt{true}, \sigma \rangle \qquad (\textsc{SmallStep-Not-False})$$

$$\frac{\langle b_1, \sigma \rangle \to \langle b_1', \sigma_1 \rangle}{\langle b_1 \,\texttt{\&\&}\, b_2, \sigma \rangle \to \langle b_1' \,\texttt{\&\&}\, b_2, \sigma_1 \rangle} \qquad (\textsc{SmallStep-And-Arg1})$$

$$\langle \texttt{false} \,\texttt{\&\&}\, b_2, \sigma \rangle \to \langle \texttt{false}, \sigma \rangle \qquad (\textsc{SmallStep-And-False})$$

$$\langle \texttt{true} \,\texttt{\&\&}\, b_2, \sigma \rangle \to \langle b_2, \sigma \rangle \qquad (\textsc{SmallStep-And-True})$$

$$\langle \texttt{\{}\, s \,\texttt{\}}, \sigma \rangle \to \langle s, \sigma \rangle \qquad \text{(SmallStep-Block)}$$

$$\frac{\langle a, \sigma \rangle \to \langle a', \sigma_1 \rangle}{\langle x = a\,\texttt{;}, \sigma \rangle \to \langle x = a'\,\texttt{;}, \sigma_1 \rangle} \qquad \text{(SmallStep-Asgn-Arg2)}$$

$$\langle x = i\,\texttt{;}, \sigma \rangle \to \langle \texttt{\{\}}, \sigma[i/x] \rangle \quad \text{if } \sigma(x) \neq \bot \qquad \text{(SmallStep-Asgn)}$$

$$\frac{\langle s_1, \sigma \rangle \to \langle s_1', \sigma_1 \rangle}{\langle s_1\ s_2, \sigma \rangle \to \langle s_1'\ s_2, \sigma_1 \rangle} \qquad \text{(SmallStep-Seq-Arg1)}$$

$$\langle \texttt{\{\}}\ s_2 \rangle \to \langle s_2, \sigma \rangle \qquad \text{(SmallStep-Seq-Empty-Block)}$$

$$\frac{\langle b, \sigma \rangle \to \langle b', \sigma_1 \rangle}{\langle \texttt{if}\ (b)\ s_1\ \texttt{else}\ s_2, \sigma \rangle \to \langle \texttt{if}\ (b')\ s_1\ \texttt{else}\ s_2, \sigma_1 \rangle} \qquad \text{(SmallStep-If-Arg1)}$$

$$\langle \texttt{if (true)}\ s_1\ \texttt{else}\ s_2, \sigma \rangle \to \langle s_1, \sigma \rangle \qquad \text{(SmallStep-If-True)}$$

$$\langle \texttt{if (false)}\ s_1\ \texttt{else}\ s_2, \sigma \rangle \to \langle s_2, \sigma \rangle \qquad \text{(SmallStep-If-False)}$$

$$\langle \texttt{while}\ (b)\ s, \sigma \rangle \to \langle \texttt{if}\ (b)\ \texttt{\{}\, s\ \texttt{while}\ (b)\ s \,\texttt{\}}\ \texttt{else}\ \texttt{\{\}}, \sigma \rangle \qquad \text{(SmallStep-While)}$$

$$\langle \texttt{int}\ xl\texttt{;}\ s \rangle \to \langle s, xl \mapsto 0 \rangle \qquad \text{(SmallStep-Var)}$$

Similarly, the premises in the increment rule could simply be $\langle x, \sigma \rangle \to \langle i, \sigma \rangle$, as the rule that could apply here is variable lookup, and that doesn't change the state.

3 IO Construct

BNF

```
Int   ::= the domain of integers with integer operations
Bool  ::= the domain of booleans
Id    ::= standard identifiers
AExp  ::= Int
        | Id
        | read()                         <-- Added Semantics
        | AExp + AExp
        | AExp / AExp
BExp  ::= Bool
        | AExp <= AExp
        | ! BExp
        | BExp && BExp
Block ::= { }
        | { Stmt }
Stmt  ::= Block
        | Id = AExp ;
        | Stmt Stmt
        | print ( AExp )                 <-- Added Semantics
        | if ( BExp ) Block else Block;
```

```
       | while ( BExp ) Block;
Pgm   ::= int List{Id} ; Stmt
```

Big Step Semantics

We see that the `read()` and `print(AExp)` is *impure* here, but it is only *impure* in a sense that it changes the outside system's state, but not the σ inside the language. So, we don't need to include state in the arithmetic and boolean operations. We do, however, need to enforce **strict sequential order** as they are *impure*. To enforce the sequential order of outside system's state, **introduce a system state** γ. There are two operations on system state. $\gamma(\leftarrow)$ which takes an integer string in and change the system state (equivalent to taking input from `stdin`), and $\gamma(\rightarrow)$ which puts an integer and change the system state (equivalent to putting output to `stdout`). For sake of simplicity, let's assume the system operation always work.

$$\langle x, \sigma, \gamma \rangle \Downarrow \langle \sigma(x), \gamma \rangle \quad \text{if } \sigma(x) \neq \bot \qquad (\textsc{BigStep-Lookup})$$

$$\frac{\langle a_1, \sigma, \gamma \rangle \Downarrow \langle i_1, \gamma_1 \rangle \quad \langle a_2, \sigma, \gamma_1 \rangle \Downarrow \langle i_2, \gamma_2 \rangle}{\langle a_1 + a_2, \sigma, \gamma \rangle \Downarrow \langle i_1 +_{\text{Int}} i_2, \sigma, \gamma_2 \rangle} \qquad (\textsc{BigStep-Add})$$

$$\frac{\langle a_1, \sigma, \gamma \rangle \Downarrow \langle i_1, \gamma_1 \rangle \quad \langle a_2, \sigma, \gamma_1 \rangle \Downarrow \langle i_2, \gamma_2 \rangle}{\langle a_1 \mathbin{/} a_2, \sigma, \gamma \rangle \Downarrow \langle i_1 /_{\text{Int}} i_2, \gamma_2 \rangle} \quad \text{if } i_2 \neq 0 \qquad (\textsc{BigStep-Div})$$

$$\langle \texttt{read()}, \sigma, \gamma \rangle \Downarrow \langle i, \gamma(\leftarrow) \rangle \qquad (\textsc{BigStep-Read})$$

$$\frac{\langle a, \sigma, \gamma \rangle \Downarrow \langle i, \gamma_1 \rangle}{\langle \texttt{print}\ (a), \sigma, \gamma \rangle \Downarrow \langle \sigma, \gamma_1(\rightarrow) \rangle} \qquad (\textsc{BigStep-Print})$$

$$\frac{\langle a_1, \sigma, \gamma \rangle \Downarrow \langle i_1, \gamma_1 \rangle \quad \langle a_2, \sigma, \gamma_1 \rangle \Downarrow \langle i_2, \gamma_2 \rangle}{\langle a_1 \texttt{ <= } a_2, \sigma, \gamma \rangle \Downarrow \langle i_1 \leq_{\text{Int}} i_2, \gamma_2 \rangle} \qquad (\textsc{BigStep-Leq})$$

$$\frac{\langle b, \sigma, \gamma \rangle \Downarrow \langle \texttt{true}, \gamma_1 \rangle}{\langle \texttt{!}\ b, \sigma, \gamma \rangle \Downarrow \langle \texttt{false}, \gamma_1 \rangle} \qquad (\textsc{BigStep-Not-True})$$

$$\frac{\langle b, \sigma, \gamma \rangle \Downarrow \langle \texttt{false}, \gamma_1 \rangle}{\langle \texttt{!}\ b, \sigma, \gamma \rangle \Downarrow \langle \texttt{true}, \gamma_1 \rangle} \qquad (\textsc{BigStep-Not-False})$$

$$\frac{\langle b_1, \sigma, \gamma \rangle \Downarrow \langle \texttt{false}, \gamma_1 \rangle}{\langle b_1 \texttt{ \&\& } b_2, \sigma \rangle \Downarrow \langle \texttt{false}, \gamma_1 \rangle} \qquad (\textsc{BigStep-And-False})$$

$$\frac{\langle b_1, \sigma, \gamma \rangle \Downarrow \langle \texttt{true}, \gamma_1 \rangle \quad \langle b_2, \sigma, \gamma_1 \rangle \Downarrow \langle t, \gamma_2 \rangle}{\langle b_1 \texttt{ \&\& } b_2, \sigma, \gamma \rangle \Downarrow \langle t, \gamma_2 \rangle} \qquad (\textsc{BigStep-And-True})$$

$$\langle \texttt{\{\}}, \sigma, \gamma \rangle \Downarrow \langle \sigma, \gamma \rangle \qquad \text{(BigStep-Empty-Block)}$$

$$\frac{\langle s, \sigma, \gamma \rangle \Downarrow \langle \sigma_1, \gamma_1 \rangle}{\langle \texttt{\{}\, s\, \texttt{\}}, \sigma, \gamma \rangle \Downarrow \langle \sigma_1, \gamma_1 \rangle} \qquad \text{(BigStep-Block)}$$

$$\frac{\langle a, \sigma, \gamma \rangle \Downarrow \langle i, \sigma, \gamma_1 \rangle}{\langle x \,\texttt{=}\, a \texttt{;}, \sigma, \gamma \rangle \Downarrow \langle \sigma[i/x], \gamma_1 \rangle} \quad \text{if } \sigma(x) \neq \bot \qquad \text{(BigStep-Asgn)}$$

$$\frac{\langle s_1, \sigma, \gamma \rangle \Downarrow \langle \sigma_1, \gamma_1 \rangle \quad \langle s_2, \sigma_1, \gamma_1 \rangle \Downarrow \langle \sigma_2, \gamma_2 \rangle}{\langle s_1\ s_2, \sigma, \gamma \rangle \Downarrow \langle \sigma_2, \gamma_2 \rangle} \qquad \text{(BigStep-Seq)}$$

$$\frac{\langle b, \sigma, \gamma \rangle \Downarrow \langle \texttt{true}, \gamma_1 \rangle \quad \langle s_1, \sigma, \gamma_1 \rangle \Downarrow \langle \sigma_1, \gamma_2 \rangle}{\langle \texttt{if (}b\texttt{)}\ s_1\ \texttt{else}\ s_2, \sigma, \gamma \rangle \Downarrow \langle \sigma_1, \gamma_2 \rangle} \qquad \text{(BigStep-If-True)}$$

$$\frac{\langle b, \sigma, \gamma \rangle \Downarrow \langle \texttt{false}, \gamma_1 \rangle \quad \langle s_2, \sigma, \gamma_1 \rangle \Downarrow \langle \sigma_2, \gamma_2 \rangle}{\langle \texttt{if (}b\texttt{)}\ s_1\ \texttt{else}\ s_2, \sigma, \gamma \rangle \Downarrow \langle \sigma_2, \gamma_2 \rangle} \qquad \text{(BigStep-If-False)}$$

$$\frac{\langle b, \sigma, \gamma \rangle \Downarrow \langle \texttt{false}, \gamma_1 \rangle}{\langle \texttt{while (}b\texttt{)}\ s, \sigma, \gamma \rangle \Downarrow \langle \sigma, \gamma_1 \rangle} \qquad \text{(BigStep-While-False)}$$

$$\frac{\langle b, \sigma, \gamma \rangle \Downarrow \langle \texttt{true}, \gamma_1 \rangle \quad \langle s\ \texttt{while (}b\texttt{)}\ s, \sigma, \gamma_1 \rangle \Downarrow \langle \sigma_1, \gamma_2 \rangle}{\langle \texttt{while (}b\texttt{)}\ s, \sigma, \gamma \rangle \Downarrow \langle \sigma_1, \gamma_2 \rangle} \qquad \text{(BigStep-While-True)}$$

Type Systems

$$xl \vdash \texttt{read()} : \text{int} \qquad \text{(BigStepTypeSystem-Read)}$$

$$\frac{xl \vdash a : \text{int}}{xl \vdash \texttt{print (}a\texttt{)} : \text{stmt}} \qquad \text{(BigStepTypeSystem-Print)}$$

Small Step Semantics

Like Big Step Semantics, we have to enforce strict sequential order, but we could prevent using γ, as we could enforce the order directly.

$$\langle x, \sigma \rangle \to \langle \sigma(x), \sigma \rangle \quad \text{if } \sigma(x) \neq \bot \qquad \text{(SmallStep-Lookup)}$$

$$\frac{\langle a_1, \sigma \rangle \to \langle a_1', \sigma \rangle}{\langle a_1 \texttt{+} a_2, \sigma \rangle \to \langle a_1' \texttt{+} a_2, \sigma \rangle} \qquad \text{(SmallStep-Add-Arg1)}$$

$$\frac{\langle a_2, \sigma \rangle \to \langle a_2', \sigma \rangle}{\langle i_1 \texttt{+} a_2, \sigma \rangle \to \langle i_1 \texttt{+} a_2', \sigma \rangle} \qquad \text{(SmallStep-Add-Arg2)}$$

$$\langle i_1 \texttt{+} i_2, \sigma \rangle \to \langle i_1 +_{\text{Int}} i_2, \sigma \rangle \qquad \text{(SmallStep-Add)}$$

$$\langle \texttt{read()}, \sigma \rangle \to \langle i, \sigma \rangle \qquad \text{(SmallStep-Read)}$$

$$\frac{\langle a, \sigma \rangle \to \langle i, \sigma \rangle}{\langle \texttt{print (}a\texttt{)}, \sigma \rangle \to \langle \texttt{\{\}}, \sigma \rangle} \qquad \text{(SmallStep-Print)}$$

$$\frac{\langle a_1, \sigma\rangle \to \langle a_1', \sigma\rangle}{\langle a_1 \,\texttt{/}\, a_2, \sigma\rangle \to \langle a_1' \,\texttt{/}\, a_2, \sigma\rangle} \quad (\textsc{SmallStep-Div-Arg1})$$

$$\frac{\langle a_2, \sigma\rangle \to \langle a_2', \sigma\rangle}{\langle i_1 \,\texttt{/}\, a_2, \sigma\rangle \to \langle i_1 \,\texttt{/}\, a_2', \sigma\rangle} \quad (\textsc{SmallStep-Div-Arg2})$$

$$\langle i_1 \,\texttt{/}\, i_2, \sigma\rangle \to \langle i_1 /_{\mathrm{Int}} i_2, \sigma\rangle \quad \text{if } i_2 \neq 0 \quad (\textsc{SmallStep-Div})$$

$$\frac{\langle a_1, \sigma\rangle \to \langle a_1', \sigma\rangle}{\langle a_1 \,\texttt{<=}\, a_2, \sigma\rangle \to \langle a_1' \,\texttt{<=}\, a_2, \sigma\rangle} \quad (\textsc{SmallStep-Leq-Arg1})$$

$$\frac{\langle a_2, \sigma\rangle \to \langle a_2', \sigma\rangle}{\langle i_1 \,\texttt{<=}\, a_2, \sigma\rangle \to \langle i_1 \,\texttt{<=}\, a_2, \sigma\rangle} \quad (\textsc{SmallStep-Leq-Arg2})$$

$$\langle i_1 \,\texttt{<=}\, i_2, \sigma\rangle \to \langle i_1 \leq_{\mathrm{Int}} i_2, \sigma\rangle \quad \text{if } i_2 \neq 0 \quad (\textsc{SmallStep-Leq})$$

$$\frac{\langle b, \sigma\rangle \to \langle b', \sigma\rangle}{\langle \texttt{!}\, b, \sigma\rangle \to \langle \texttt{!}\, b', \sigma\rangle} \quad (\textsc{SmallStep-Not-Arg})$$

$$\langle \texttt{!}\,\texttt{true}, \sigma\rangle \to \langle \texttt{false}, \sigma\rangle \quad (\textsc{SmallStep-Not-True})$$

$$\langle \texttt{!}\,\texttt{false}, \sigma\rangle \to \langle \texttt{true}, \sigma\rangle \quad (\textsc{SmallStep-Not-False})$$

$$\frac{\langle b_1, \sigma\rangle \to \langle b_1', \sigma\rangle}{\langle b_1 \,\texttt{\&\&}\, b_2, \sigma\rangle \to \langle b_1' \,\texttt{\&\&}\, b_2, \sigma\rangle} \quad (\textsc{SmallStep-And-Arg1})$$

$$\langle \texttt{false} \,\texttt{\&\&}\, b_2, \sigma\rangle \to \langle \texttt{false}, \sigma\rangle \quad (\textsc{SmallStep-And-False})$$

$$\langle \texttt{true} \,\texttt{\&\&}\, b_2, \sigma\rangle \to \langle b_2, \sigma\rangle \quad (\textsc{SmallStep-And-True})$$

$$\langle \texttt{\{}\, s\, \texttt{\}}, \sigma\rangle \to \langle s, \sigma\rangle \quad (\textsc{SmallStep-Block})$$

$$\frac{\langle a, \sigma\rangle \to \langle a', \sigma\rangle}{\langle x \,\texttt{=}\, a\,\texttt{;}, \sigma\rangle \to \langle x \,\texttt{=}\, a'\,\texttt{;}, \sigma\rangle} \quad (\textsc{SmallStep-Asgn-Arg2})$$

$$\langle x \,\texttt{=}\, i\,\texttt{;}, \sigma\rangle \to \langle \texttt{\{\}}, \sigma[i/x]\rangle \quad \text{if } \sigma(x) \neq \bot \quad (\textsc{SmallStep-Asgn})$$

$$\frac{\langle s_1, \sigma\rangle \to \langle s_1', \sigma\rangle}{\langle s_1\ s_2, \sigma\rangle \to \langle s_1'\ s_2, \sigma\rangle} \quad (\textsc{SmallStep-Seq-Arg1})$$

$$\langle \texttt{\{\}}\ \ s_2\rangle \to \langle s_2, \sigma\rangle \quad (\textsc{SmallStep-Seq-Empty-Block})$$

$$\frac{\langle b, \sigma\rangle \to \langle b', \sigma\rangle}{\langle \texttt{if}\,(b)\, s_1\, \texttt{else}\ s_2, \sigma\rangle \to \langle \texttt{if}\,(b')\, s_1\, \texttt{else}\ s_2, \sigma\rangle} \quad (\textsc{SmallStep-If-Arg1})$$

$$\langle \texttt{if (true)}\, s_1\, \texttt{else}\, s_2, \sigma\rangle \to \langle s_1, \sigma\rangle \quad (\textsc{SmallStep-If-True})$$

$$\langle \texttt{if (false)}\, s_1\, \texttt{else}\, s_2, \sigma\rangle \to \langle s_2, \sigma\rangle \quad (\textsc{SmallStep-If-False})$$

$$\langle \texttt{while}\,(b)\, s, \sigma\rangle \to \langle \texttt{if}\,(b)\,\texttt{\{}\, s\, \texttt{while}\,(b)\, s\,\texttt{\}}\, \texttt{else}\,\texttt{\{\}}, \sigma\rangle \quad (\textsc{SmallStep-While})$$

$$\langle \texttt{int}\, xl\texttt{;}\ s\rangle \to \langle s, xl \mapsto 0\rangle \quad (\textsc{SmallStep-Var})$$

4 All Together

It's very disgusting. (First time my interactive latex editor has temporarily frozened!) The general idea is pretty straightforward. For big step semantics, add the states and add the halt case. For small semantics, add the evaluation order and add the halt case. They are pretty hard to work with, as adding one new feature usually requires changes all the rules more or less, and sometimes need to duplicate rules two times or three times to propagate the errors / information. Small step semantics seems easier to work with, as it's more granular by design.

Big Step Semantics

$$\langle x, \sigma, \gamma\rangle \Downarrow \langle \sigma(x), \sigma, \gamma\rangle \quad \text{if } \sigma(x) \neq \bot \qquad (\textsc{BigStep-Lookup})$$

$$\frac{\langle a_1, \sigma, \gamma\rangle \Downarrow \langle i_1, \sigma_1, \gamma_1\rangle \quad \langle a_2, \sigma_1, \gamma_1\rangle \Downarrow \langle i_2, \sigma_2, \gamma_2\rangle}{\langle a_1 + a_2, \sigma\rangle \Downarrow \langle i_1 +_{\text{Int}} i_2, \sigma_2, \gamma_2\rangle} \qquad (\textsc{BigStep-Add})$$

$$\frac{\langle a_1, \sigma, \gamma\rangle \Downarrow \langle \text{divzero}(e)\rangle}{\langle a_1 + a_2, \sigma\rangle \Downarrow \langle \text{divzero}(e)\rangle} \qquad \boxed{\textsc{BigStep-AddHalt1}}$$

$$\frac{\langle a_1, \sigma, \gamma\rangle \Downarrow \langle i_1, \sigma_1, \gamma_1\rangle \quad \langle a_2, \sigma_1, \gamma_1\rangle \Downarrow \langle \text{divzero}(e)\rangle}{\langle a_1 + a_2, \sigma\rangle \Downarrow \langle \text{divzero}(e)\rangle} \qquad \boxed{\textsc{BigStep-AddHalt2}}$$

$$\frac{\langle a_1, \sigma, \gamma\rangle \Downarrow \langle i_1, \sigma_1, \gamma_1\rangle \quad \langle a_2, \sigma_1, \gamma_1\rangle \Downarrow \langle i_2, \sigma_2, \gamma_2\rangle}{\langle a_1 \,/\, a_2, \sigma, \gamma\rangle \Downarrow \langle i_1 /_{\text{Int}} i_2, \sigma_2, \gamma_2\rangle} \quad \text{if } i_2 \neq 0 \qquad (\textsc{BigStep-Div})$$

$$\frac{\langle a_1, \sigma, \gamma\rangle \Downarrow \langle \text{divzero}(e)\rangle}{\langle a_1 \,/\, a_2, \sigma, \gamma\rangle \Downarrow \langle \text{divzero}(e)\rangle} \quad \text{if } i_2 \neq 0 \qquad (\textsc{BigStep-DivHalt1})$$

$$\frac{\langle a_1, \sigma, \gamma\rangle \Downarrow \langle i_1, \sigma_1, \gamma_1\rangle \quad \langle a_2, \sigma_1, \gamma_1\rangle \Downarrow \langle \text{divzero}(e)\rangle}{\langle a_1 \,/\, a_2, \sigma, \gamma\rangle \Downarrow \langle \text{divzero}(e)\rangle} \quad \text{if } i_2 \neq 0 \qquad (\textsc{BigStep-DivHalt2})$$

$$\frac{\langle a_1, \sigma, \gamma\rangle \Downarrow \langle i_1, \sigma_1, \gamma_1\rangle \quad \langle a_2, \sigma_1, \gamma_1\rangle \Downarrow \langle 0, \sigma_2, \gamma_2\rangle}{\langle a_1 \,/\, a_2, \sigma\rangle \Downarrow \langle \text{divzero}((a_1 \Downarrow i_1) \,/\, a_2, \sigma_2)\rangle} \qquad \boxed{\textsc{BigStep-Divzero}}$$

$$\frac{\langle x, \sigma, \gamma\rangle \Downarrow \langle i, \sigma_1, \gamma_1\rangle}{\langle {+}{+}\, x, \sigma, \gamma\rangle \Downarrow \langle i +_{\text{Int}} 1, \sigma_1[i +_{\text{Int}} 1/x], \gamma_1\rangle} \quad \text{if } \sigma_1(x) \neq \bot \qquad (\textsc{BigStep-Increment})$$

$$\langle \texttt{read()}, \sigma, \gamma\rangle \Downarrow \langle i, \sigma, \gamma(\leftarrow)\rangle \qquad (\textsc{BigStep-Read})$$

$$\frac{\langle a, \sigma, \gamma\rangle \Downarrow \langle i, \sigma_1, \gamma_1\rangle}{\langle \texttt{print (}a\texttt{)}, \sigma, \gamma\rangle \Downarrow \langle \sigma_1, \gamma_1(\rightarrow)\rangle} \qquad (\textsc{BigStep-Print})$$

$$\frac{\langle a, \sigma, \gamma\rangle \Downarrow \langle \text{divzero}(e)\rangle}{\langle \texttt{print (}a\texttt{)}, \sigma, \gamma\rangle \Downarrow \langle \text{divzero}(e)\rangle} \qquad (\textsc{BigStep-PrintHalt})$$

$$\frac{\langle a_1, \sigma, \gamma\rangle \Downarrow \langle i_1, \sigma_1, \gamma_1\rangle \quad \langle a_2, \sigma_1, \gamma_1\rangle \Downarrow \langle i_2, \sigma_2, \gamma_2\rangle}{\langle a_1 \texttt{ <= } a_2, \sigma, \gamma\rangle \Downarrow \langle i_1 \leq_{\text{Int}} i_2, \sigma_2, \gamma_2\rangle} \quad \text{(BigStep-Leq)}$$

$$\frac{\langle a_1, \sigma, \gamma\rangle \Downarrow \langle \text{divzero}(e)\rangle}{\langle a_1 \texttt{ <= } a_2, \sigma, \gamma\rangle \Downarrow \langle \text{divzero}(e)\rangle} \quad \text{(BigStep-LeqHalt1)}$$

$$\frac{\langle a_1, \sigma, \gamma\rangle \Downarrow \langle i_1, \sigma_1, \gamma_1\rangle \quad \langle a_2, \sigma_1, \gamma_1\rangle \Downarrow \langle \text{divzero}(e)\rangle}{\langle a_1 \texttt{ <= } a_2, \sigma, \gamma\rangle \Downarrow \langle \text{divzero}(e)\rangle} \quad \text{(BigStep-LeqHalt2)}$$

$$\frac{\langle b, \sigma, \gamma\rangle \Downarrow \langle \texttt{true}, \sigma_1, \gamma_1\rangle}{\langle \texttt{!}\, b, \sigma, \gamma\rangle \Downarrow \langle \texttt{false}, \sigma_1, \gamma_1\rangle} \quad \text{(BigStep-Not-True)}$$

$$\frac{\langle b, \sigma, \gamma\rangle \Downarrow \langle \text{divzero}(e)\rangle}{\langle \texttt{!}\, b, \sigma, \gamma\rangle \Downarrow \langle \text{divzero}(e)\rangle} \quad \text{(BigStep-Not-TrueHalt)}$$

$$\frac{\langle b, \sigma, \gamma\rangle \Downarrow \langle \texttt{false}, \sigma_1, \gamma_1\rangle}{\langle \texttt{!}\, b, \sigma, \gamma\rangle \Downarrow \langle \texttt{true}, \sigma_1, \gamma_1\rangle} \quad \text{(BigStep-Not-False)}$$

$$\frac{\langle b_1, \sigma, \gamma\rangle \Downarrow \langle \text{divzero}(e)\rangle}{\langle b_1 \texttt{ \&\& } b_2, \sigma\rangle \Downarrow \langle \text{divzero}(e)\rangle} \quad \text{(BigStep-And-FalseHalt)}$$

$$\frac{\langle b_1, \sigma, \gamma\rangle \Downarrow \langle \texttt{true}, \sigma_1, \gamma_1\rangle \quad \langle b_2, \sigma_1, \gamma_1\rangle \Downarrow \langle t, \sigma_2, \gamma_2\rangle}{\langle b_1 \texttt{ \&\& } b_2, \sigma, \gamma\rangle \Downarrow \langle t, \sigma_2, \gamma_2\rangle} \quad \text{(BigStep-And-True)}$$

$$\frac{\langle b_1, \sigma, \gamma\rangle \Downarrow \langle \text{divzero}(e)\rangle}{\langle b_1 \texttt{ \&\& } b_2, \sigma, \gamma\rangle \Downarrow \langle \text{divzero}(e)\rangle} \quad \text{(BigStep-And-TrueHalt1)}$$

$$\frac{\langle b_1, \sigma, \gamma\rangle \Downarrow \langle \texttt{true}, \sigma_1, \gamma_1\rangle \quad \langle b_2, \sigma_1, \gamma_1\rangle \Downarrow \langle \text{divzero}(e)\rangle}{\langle b_1 \texttt{ \&\& } b_2, \sigma, \gamma\rangle \Downarrow \langle \text{divzero}(e)\rangle} \quad \text{(BigStep-And-TrueHalt2)}$$

$$\langle \texttt{\{\}}, \sigma, \gamma\rangle \Downarrow \langle \sigma, \gamma\rangle \quad \text{(BigStep-Empty-Block)}$$

$$\frac{\langle s, \sigma, \gamma\rangle \Downarrow \langle \sigma_1, \gamma_1\rangle}{\langle \texttt{\{}\, s\, \texttt{\}}, \sigma, \gamma\rangle \Downarrow \langle \sigma_1, \gamma_1\rangle} \quad \text{(BigStep-Block)}$$

$$\frac{\langle s, \sigma, \gamma\rangle \Downarrow \langle \text{divzero}(e)\rangle}{\langle \texttt{\{}\, s\, \texttt{\}}, \sigma, \gamma\rangle \Downarrow \langle \text{divzero}(e)\rangle} \quad \text{(BigStep-BlockHalt)}$$

$$\frac{\langle a, \sigma, \gamma\rangle \Downarrow \langle i, \sigma_1, \gamma_1\rangle}{\langle x \texttt{=} a\texttt{;}, \sigma, \gamma\rangle \Downarrow \langle \sigma_1[i/x], \gamma_1\rangle} \quad \text{if } \sigma_1(x) \neq \bot \quad \text{(BigStep-Asgn)}$$

$$\frac{\langle a, \sigma, \gamma\rangle \Downarrow \langle \text{divzero}(e)\rangle}{\langle x \texttt{=} a\texttt{;}, \sigma, \gamma\rangle \Downarrow \langle \text{divzero}(e)\rangle} \quad \text{if } \sigma_1(x) \neq \bot \quad \text{(BigStep-AsgnHalt)}$$

$$\frac{\langle s_1, \sigma, \gamma\rangle \Downarrow \langle \sigma_1, \gamma_1\rangle \quad \langle s_2, \sigma_1, \gamma_1\rangle \Downarrow \langle \sigma_2, \gamma_2\rangle}{\langle s_1\ s_2, \sigma, \gamma\rangle \Downarrow \langle \sigma_2, \gamma_2\rangle} \quad \text{(BigStep-Seq)}$$

$$\frac{\langle s_1, \sigma, \gamma\rangle \Downarrow \langle \text{divzero}(e)\rangle}{\langle s_1\ s_2, \sigma, \gamma\rangle \Downarrow \langle \text{divzero}(e)\rangle} \quad \text{(BigStep-SeqHalt1)}$$

$$\frac{\langle s_1, \sigma, \gamma\rangle \Downarrow \langle \sigma_1, \gamma_1\rangle \quad \langle s_2, \sigma_1, \gamma_1\rangle \Downarrow \langle \text{divzero}(e)\rangle}{\langle s_1\ s_2, \sigma, \gamma\rangle \Downarrow \langle \text{divzero}(e)\rangle} \quad \text{(BigStep-SeqHalt2)}$$

$$\frac{\langle b, \sigma, \gamma \rangle \Downarrow \langle \texttt{true}, \sigma_1, \gamma_1 \rangle \quad \langle s_1, \sigma_1, \gamma_1 \rangle \Downarrow \langle \sigma_2, \gamma_2 \rangle}{\langle \texttt{if (}b\texttt{)}\ s_1\ \texttt{else}\ s_2, \sigma, \gamma \rangle \Downarrow \langle \sigma_2, \gamma_2 \rangle} \quad (\textsc{BigStep-If-True})$$

$$\frac{\langle b, \sigma, \gamma \rangle \Downarrow \langle \text{divzero}(e) \rangle}{\langle \texttt{if (}b\texttt{)}\ s_1\ \texttt{else}\ s_2, \sigma, \gamma \rangle \Downarrow \langle \text{divzero}(e) \rangle} \quad (\textsc{BigStep-If-TrueHalt1})$$

$$\frac{\langle b, \sigma, \gamma \rangle \Downarrow \langle \texttt{true}, \sigma_1, \gamma_1 \rangle \quad \langle s_1, \sigma_1, \gamma_1 \rangle \Downarrow \langle \text{divzero}(e) \rangle}{\langle \texttt{if (}b\texttt{)}\ s_1\ \texttt{else}\ s_2, \sigma, \gamma \rangle \Downarrow \langle \text{divzero}(e) \rangle} \quad (\textsc{BigStep-If-TrueHalt2})$$

$$\frac{\langle b, \sigma, \gamma \rangle \Downarrow \langle \texttt{false}, \sigma_1, \gamma_1 \rangle \quad \langle s_2, \sigma_1, \gamma_1 \rangle \Downarrow \langle \sigma_2, \gamma_2 \rangle}{\langle \texttt{if (}b\texttt{)}\ s_1\ \texttt{else}\ s_2, \sigma, \gamma \rangle \Downarrow \langle \sigma_2, \gamma_2 \rangle} \quad (\textsc{BigStep-If-False})$$

$$\frac{\langle b, \sigma, \gamma \rangle \Downarrow \langle \texttt{false}, \sigma_1, \gamma_1 \rangle \quad \langle s_1, \sigma_1, \gamma_1 \rangle \Downarrow \langle \text{divzero}(e) \rangle}{\langle \texttt{if (}b\texttt{)}\ s_1\ \texttt{else}\ s_2, \sigma, \gamma \rangle \Downarrow \langle \text{divzero}(e) \rangle} \quad (\textsc{BigStep-If-FalseHalt2})$$

$$\frac{\langle b, \sigma, \gamma \rangle \Downarrow \langle \texttt{false}, \sigma_1, \gamma_1 \rangle}{\langle \texttt{while (}b\texttt{)}\ s, \sigma, \gamma \rangle \Downarrow \langle \sigma_1, \gamma_1 \rangle} \quad (\textsc{BigStep-While-False})$$

$$\frac{\langle b, \sigma, \gamma \rangle \Downarrow \langle \text{divzero}(e) \rangle}{\langle \texttt{while (}b\texttt{)}\ s, \sigma, \gamma \rangle \Downarrow \langle \text{divzero}(e) \rangle} \quad (\textsc{BigStep-While-FalseHalt})$$

$$\frac{\langle b, \sigma, \gamma \rangle \Downarrow \langle \texttt{true}, \sigma_1, \gamma_1 \rangle \quad \langle s\ \texttt{while (}b\texttt{)}\ s, \sigma_1, \gamma_1 \rangle \Downarrow \langle \sigma_2, \gamma_2 \rangle}{\langle \texttt{while (}b\texttt{)}\ s, \sigma, \gamma \rangle \Downarrow \langle \sigma_2, \gamma_2 \rangle} \quad (\textsc{BigStep-While-True})$$

$$\frac{\langle b, \sigma, \gamma \rangle \Downarrow \langle \text{divzero}(e) \rangle}{\langle \texttt{while (}b\texttt{)}\ s, \sigma, \gamma \rangle \Downarrow \langle \text{divzero}(e) \rangle} \quad (\textsc{BigStep-While-TrueHalt1})$$

$$\frac{\langle b, \sigma, \gamma \rangle \Downarrow \langle \texttt{true}, \sigma_1, \gamma_1 \rangle \quad \langle s\ \texttt{while (}b\texttt{)}\ s, \sigma_1, \gamma_1 \rangle \Downarrow \langle \text{divzero}(e) \rangle}{\langle \texttt{while (}b\texttt{)}\ s, \sigma, \gamma \rangle \Downarrow \langle \text{divzero}(e) \rangle} \quad (\textsc{BigStep-While-TrueHalt2})$$

Increment won't have halt case, as its premise is essentially a variable lookup, which cannot have errors.

Small Step Semantics

$$\langle x, \sigma \rangle \to \langle \sigma(x), \sigma \rangle \quad \text{if } \sigma(x) \neq \bot \quad (\textsc{SmallStep-Lookup})$$

$$\frac{\langle a_1, \sigma \rangle \to \langle a_1', \sigma_1 \rangle}{\langle a_1 + a_2, \sigma \rangle \to \langle a_1' + a_2, \sigma_1 \rangle} \quad (\textsc{SmallStep-Add-Arg1})$$

$$\frac{\langle a_1, \sigma \rangle \to \langle \text{divzero}(e) \rangle}{\langle a_1 + a_2, \sigma \rangle \to \langle \text{divzero}(e) \rangle} \quad (\textsc{SmallStep-Add-Arg1-Halt})$$

$$\frac{\langle a_2, \sigma \rangle \to \langle a_2', \sigma_1 \rangle}{\langle i_1 + a_2, \sigma \rangle \to \langle i_1 + a_2', \sigma_1 \rangle} \quad (\textsc{SmallStep-Add-Arg2})$$

$$\frac{\langle a_2, \sigma \rangle \to \langle \text{divzero}(e) \rangle}{\langle i_1 + a_2, \sigma \rangle \to \langle \text{divzero}(e) \rangle} \quad (\textsc{SmallStep-Add-Arg2-Halt})$$

$$\langle i_1 + i_2, \sigma \rangle \to \langle i_1 +_{\text{Int}} i_2, \sigma \rangle \quad (\textsc{SmallStep-Add})$$

$$\frac{\langle a_1, \sigma\rangle \to \langle a_1', \sigma_1\rangle}{\langle a_1 \texttt{ / } a_2, \sigma\rangle \to \langle a_1' \texttt{ / } a_2, \sigma_1\rangle} \qquad (\textsc{SmallStep-Div-Arg1})$$

$$\frac{\langle a_1, \sigma\rangle \to \langle \mathrm{divzero}(e)\rangle}{\langle a_1 \texttt{ / } a_2, \sigma\rangle \to \langle \mathrm{divzero}(e)\rangle} \qquad (\textsc{SmallStep-Div-Arg1-Halt})$$

$$\frac{\langle a_2, \sigma\rangle \to \langle a_2', \sigma_1\rangle}{\langle i_1 \texttt{ / } a_2, \sigma\rangle \to \langle i_1 \texttt{ / } a_2', \sigma_1\rangle} \qquad (\textsc{SmallStep-Div-Arg2})$$

$$\frac{\langle a_2, \sigma\rangle \to \langle \mathrm{divzero}(e)\rangle}{\langle i_1 \texttt{ / } a_2, \sigma\rangle \to \langle \mathrm{divzero}(e)\rangle} \qquad (\textsc{SmallStep-Div-Arg2-Halt})$$

$$\langle i_1 \texttt{ / } i_2, \sigma\rangle \to \langle i_1 /_{\mathrm{Int}}\, i_2, \sigma\rangle \quad \text{if } i_2 \neq 0 \qquad (\textsc{SmallStep-Div})$$

$$\frac{\langle a_2, \sigma\rangle \to \langle 0, \sigma_1\rangle}{\langle i_1 \texttt{ / } a_2, \sigma\rangle \to \langle \mathrm{divzero}(i_1 \texttt{ / } a_2, \sigma_1)\rangle} \qquad (\textsc{SmallStep-Divzero})$$

$$\frac{\langle a_1, \sigma\rangle \to \langle a_1', \sigma_1\rangle}{\langle a_1 \texttt{<=} a_2, \sigma\rangle \to \langle a_1' \texttt{<=} a_2, \sigma_1\rangle} \qquad (\textsc{SmallStep-Leq-Arg1})$$

$$\frac{\langle a_1, \sigma\rangle \to \langle \mathrm{divzero}(e)\rangle}{\langle a_1 \texttt{<=} a_2, \sigma\rangle \to \langle \mathrm{divzero}(e)\rangle} \qquad (\textsc{SmallStep-Leq-Arg1-Halt})$$

$$\frac{\langle a_2, \sigma\rangle \to \langle a_2', \sigma_1\rangle}{\langle i_1 \texttt{<=} a_2, \sigma\rangle \to \langle i_1 \texttt{<=} a_2, \sigma_1\rangle} \qquad (\textsc{SmallStep-Leq-Arg2})$$

$$\frac{\langle a_2, \sigma\rangle \to \langle \mathrm{divzero}(e)\rangle}{\langle a_1 \texttt{<=} a_2, \sigma\rangle \to \langle \mathrm{divzero}(e)\rangle} \qquad (\textsc{SmallStep-Leq-Arg2-Halt})$$

$$\langle i_1 \texttt{<=} i_2, \sigma\rangle \to \langle i_1 \leq_{\mathrm{Int}} i_2, \sigma\rangle \qquad (\textsc{SmallStep-Leq})$$

$$\frac{\langle b, \sigma\rangle \to \langle b', \sigma_1\rangle}{\langle \texttt{!}\, b, \sigma\rangle \to \langle \texttt{!}\, b', \sigma_1\rangle} \qquad (\textsc{SmallStep-Not-Arg})$$

$$\frac{\langle b, \sigma\rangle \to \langle \mathrm{divzero}(e)\rangle}{\langle \texttt{!}\, b, \sigma\rangle \to \langle \mathrm{divzero}(e)\rangle} \qquad (\textsc{SmallStep-Not-ArgHalt})$$

$$\langle \texttt{! true}, \sigma\rangle \to \langle \texttt{false}, \sigma\rangle \qquad (\textsc{SmallStep-Not-True})$$

$$\langle \texttt{! false}, \sigma\rangle \to \langle \texttt{true}, \sigma\rangle \qquad (\textsc{SmallStep-Not-False})$$

$$\frac{\langle b_1, \sigma\rangle \to \langle b_1', \sigma_1\rangle}{\langle b_1 \texttt{ \&\& } b_2, \sigma\rangle \to \langle b_1' \texttt{ \&\& } b_2, \sigma_1\rangle} \qquad (\textsc{SmallStep-And-Arg1})$$

$$\frac{\langle b_1, \sigma\rangle \to \langle \mathrm{divzero}(e)\rangle}{\langle b_1 \texttt{ \&\& } b_2, \sigma\rangle \to \langle \mathrm{divzero}(e)\rangle} \qquad (\textsc{SmallStep-And-Arg1-Halt})$$

$$\langle \texttt{false \&\& } b_2, \sigma\rangle \to \langle \texttt{false}, \sigma\rangle \qquad (\textsc{SmallStep-And-False})$$

$$\langle \texttt{true \&\& } b_2, \sigma\rangle \to \langle b_2, \sigma\rangle \qquad (\textsc{SmallStep-And-True})$$

$$\langle \texttt{\{}\, s\, \texttt{\}}, \sigma\rangle \to \langle s, \sigma\rangle \qquad (\textsc{SmallStep-Block})$$

$$\frac{\langle a, \sigma\rangle \to \langle a', \sigma_1\rangle}{\langle x \,\texttt{=}\, a\,\texttt{;}, \sigma\rangle \to \langle x \,\texttt{=}\, a'\,\texttt{;}, \sigma_1\rangle} \qquad (\textsc{SmallStep-Asgn-Arg2})$$

$$\frac{\langle a, \sigma\rangle \to \langle \mathrm{divzero}(e)\rangle}{\langle x \,\texttt{=}\, a\,\texttt{;}, \sigma\rangle \to \langle \mathrm{divzero}(e)\rangle} \qquad (\textsc{SmallStep-Asgn-Arg2-Halt})$$

$$\langle x \,\texttt{=}\, i\,\texttt{;}, \sigma\rangle \to \langle \texttt{\{\}}, \sigma[i/x]\rangle \quad \text{if } \sigma(x) \neq \bot \qquad (\textsc{SmallStep-Asgn})$$

$$\frac{\langle s_1, \sigma\rangle \to \langle s_1', \sigma_1\rangle}{\langle s_1\ s_2, \sigma\rangle \to \langle s_1'\ s_2, \sigma_1\rangle} \qquad (\textsc{SmallStep-Seq-Arg1})$$

$$\frac{\langle s_1, \sigma\rangle \to \langle \mathrm{divzero}(e)\rangle}{\langle s_1\ s_2, \sigma\rangle \to \langle \mathrm{divzero}(e)\rangle} \qquad (\textsc{SmallStep-Seq-Arg1-Halt})$$

$$\langle \texttt{\{\}}\ \ s_2\rangle \to \langle s_2, \sigma\rangle \qquad (\textsc{SmallStep-Seq-Empty-Block})$$

$$\frac{\langle b, \sigma\rangle \to \langle b', \sigma_1\rangle}{\langle \texttt{if}\ (b)\ s_1\ \texttt{else}\ s_2, \sigma\rangle \to \langle \texttt{if}\ (b')\ s_1\ \texttt{else}\ s_2, \sigma_1\rangle} \qquad (\textsc{SmallStep-If-Arg1})$$

$$\frac{\langle b, \sigma\rangle \to \langle \mathrm{divzero}(e)\rangle}{\langle \texttt{if}\ (b)\ s_1\ \texttt{else}\ s_2, \sigma\rangle \to \langle \mathrm{divzero}(e)\rangle} \qquad (\textsc{SmallStep-If-Arg1-Halt})$$

$$\langle \texttt{if (true)}\ s_1\ \texttt{else}\ s_2, \sigma\rangle \to \langle s_1, \sigma\rangle \qquad (\textsc{SmallStep-If-True})$$

$$\langle \texttt{if (false)}\ s_1\ \texttt{else}\ s_2, \sigma\rangle \to \langle s_2, \sigma\rangle \qquad (\textsc{SmallStep-If-False})$$

$$\langle \texttt{while (}b\texttt{)}\ s, \sigma\rangle \to \langle \texttt{if (}b\texttt{) \{}\ s\ \texttt{while (}b\texttt{)}\ s\ \texttt{\} else \{\}}, \sigma\rangle \qquad (\textsc{SmallStep-While})$$

$$\langle \texttt{int}\ xl\texttt{;}\ s\rangle \to \langle s, xl \mapsto 0\rangle \qquad (\textsc{SmallStep-Var})$$

$$\frac{\langle x, \sigma\rangle \to \langle i, \sigma_1\rangle}{\langle \texttt{++}\,x, \sigma\rangle \to \langle i +_{\mathrm{Int}} 1, \sigma_1[i +_{\mathrm{Int}} 1/x]\rangle} \quad \text{if } \sigma_1(x) \neq \bot \qquad (\textsc{SmallStep-Increment})$$

$$\langle \texttt{read()}, \sigma\rangle \to \langle i, \sigma\rangle \qquad (\textsc{SmallStep-Read})$$

$$\frac{\langle a, \sigma\rangle \to \langle i, \sigma_1\rangle}{\langle \texttt{print (}a\texttt{)}, \sigma\rangle \to \langle \texttt{\{\}}, \sigma_1\rangle} \qquad (\textsc{SmallStep-Print})$$

$$\frac{\langle a, \sigma\rangle \to \langle \mathrm{divzero}(e)\rangle}{\langle \texttt{print (}a\texttt{)}, \sigma\rangle \to \langle \texttt{\{\}}, \mathrm{divzero}(e)\rangle} \qquad (\textsc{SmallStep-PrintHalt})$$