Language for the Interpreter (simplified)

The language for the interpreter can be described by the following grammar:

```
<const> ::= int | name

<prog>  ::= <com> | <com>;<prog>

<com>   ::= push <const> | pop | add | sub | mul | div
```

What is the form of a program?

```
<com>;<com>;…;<com>
```

Is this the common way we write programs?

Arithmetical expressions: shape of expressions

- Let us consider this simple language for expressions

```
<expr> ::= <expr> <addop> <expr> | nat
<addop>::= add | sub
```

What are the challenges here?

What is the form of a program?

`nat(add|sub)nat(add|sub)nat...`

Do we need a stack here?

Operational semantics for basic arithmetical expressions

$$e \longrightarrow e'$$

Here the expression e is itself a configuration. We already have all the information we need to execute it.

Some rules

$$\frac{}{\texttt{v}_1 \ \texttt{add} \ \texttt{v}_2 \rightarrow \texttt{v}_1\texttt{+}\texttt{v}_2}$$

$$\frac{}{\texttt{v}_1 \ \texttt{sub} \ \texttt{v}_2 \rightarrow \texttt{v}_1\texttt{-}\texttt{v}_2}$$

What can we do when we have expressions instead of a values $\texttt{v}_1$ $\texttt{v}_2$?

Summing up

Are we done?

$$\frac{}{v_1 \texttt{ add } v_2 \to v_1+v_2}$$

$$\frac{}{v_1 \texttt{ sub } v_2 \to v_1-v_2}$$

$$\frac{e_1 \to e_1'}{e_1 \texttt{ add } e_2 \to e_1' \texttt{ add } e_2}$$

$$\frac{e_2 \to e_2'}{v_1 \texttt{ add } e_2 \to v_1 \texttt{ add } e_2'}$$

$$\frac{e_1 \to e_1'}{e_1 \texttt{ sub } e_2 \to e_1' \texttt{ sub } e_2}$$

$$\frac{e_2 \to e_2'}{v_1 \texttt{ sub } e_2 \to v_1 \texttt{ sub } e_2'}$$

Multiple steps of Operational semantics

We can define a multistep semantics as:

$$\boxed{e \rightarrow^{k} e'}$$

$$\frac{}{e \rightarrow^{0} e}$$

$$\frac{e \rightarrow e' \qquad e' \rightarrow^{k} e''}{e \rightarrow^{k+1} e''}$$

Summing up

$$\frac{}{v_1 \texttt{ add } v_2 \rightarrow v_1+v_2}\ (+V)$$

$$\frac{}{v_1 \texttt{ sub } v_2 \rightarrow v_1-v_2}\ (-V)$$

$$\frac{e_1 \rightarrow e_1'}{e_1 \texttt{ add } e_2 \rightarrow e_1' \texttt{ add } e_2}\ (+e1)$$

$$\frac{e_2 \rightarrow e_2'}{v_1 \texttt{ add } e_2 \rightarrow v_1 \texttt{ add } e_2'}\ (+e2)$$

$$\frac{e_1 \rightarrow e_1'}{e_1 \texttt{ sub } e_2 \rightarrow e_1' \texttt{ sub } e_2}\ (-e1)$$

$$\frac{e_2 \rightarrow e_2'}{v_1 \texttt{ sub } e_2 \rightarrow v_1 \texttt{ sub } e_2'}\ (-e2)$$

$$\frac{}{e \rightarrow^0 e}\ (s0)$$

$$\frac{e \rightarrow e' \qquad e' \rightarrow^k e''}{e \rightarrow^{k+1} e''}\ (s1)$$

An example:

$$\dfrac{\dfrac{\dfrac{}{2\ \mathtt{add}\ 3 \to 5}\,(\mathtt{+v})}{2\ \mathtt{add}\ 3\ \mathtt{add}\ 4 \to 5\ \mathtt{add}\ 4}\,(\mathtt{+e1}) \qquad \dfrac{\dfrac{}{5\ \mathtt{add}\ 4 \to 9}\,(\mathtt{+v}) \quad \dfrac{}{9 \to_0 9}\,(\mathtt{s0})}{5\ \mathtt{add}\ 4 \to_1 9}\,(\mathtt{s1})}{2\ \mathtt{add}\ 3\ \mathtt{add}\ 4 \to_2 9}\,(\mathtt{s1})$$

An example:

$$\texttt{2 add 3 add 4} \longrightarrow_2 \texttt{9}$$

An example:

$$\frac{2\ \mathtt{add}\ 3\ \mathtt{add}\ 4 \longrightarrow 5\ \mathtt{add}\ 4 \qquad\qquad 5\ \mathtt{add}\ 4 \longrightarrow_1 9}{2\ \mathtt{add}\ 3\ \mathtt{add}\ 4 \longrightarrow_2 9}\ (\mathtt{s1})$$

An example:

$$\dfrac{\dfrac{2\ \texttt{add}\ 3 \longrightarrow 5}{2\ \texttt{add}\ 3\ \texttt{add}\ 4 \longrightarrow 5\ \texttt{add}\ 4}\,\texttt{(+e1)} \qquad 5\ \texttt{add}\ 4 \longrightarrow_1 9}{2\ \texttt{add}\ 3\ \texttt{add}\ 4 \longrightarrow_2 9}\,\texttt{(s1)}$$

An example:

$$\dfrac{\dfrac{\dfrac{}{2 \ \mathtt{add} \ 3 \longrightarrow 5}\,(\mathtt{+v})}{2 \ \mathtt{add} \ 3 \ \mathtt{add} \ 4 \longrightarrow 5 \ \mathtt{add} \ 4}\,(\mathtt{+e1}) \qquad 5 \ \mathtt{add} \ 4 \longrightarrow_1 9}{2 \ \mathtt{add} \ 3 \ \mathtt{add} \ 4 \longrightarrow_2 9}\,(\mathtt{s1})$$

An example:

$$\dfrac{\dfrac{\dfrac{}{2\ \mathtt{add}\ 3 \longrightarrow 5}\,(\mathtt{+v})}{2\ \mathtt{add}\ 3\ \mathtt{add}\ 4 \longrightarrow 5\ \mathtt{add}\ 4}\,(\mathtt{+e1}) \qquad \dfrac{5\ \mathtt{add}\ 4 \longrightarrow 9 \qquad 9 \longrightarrow_0 9}{5\ \mathtt{add}\ 4 \longrightarrow_1 9}\,(\mathtt{s1})}{2\ \mathtt{add}\ 3\ \mathtt{add}\ 4 \longrightarrow_2 9}\,(\mathtt{s1})$$

An example:

$$\dfrac{\dfrac{\dfrac{}{2 \;\texttt{add}\; 3 \to 5}\,\texttt{(+v)}}{2 \;\texttt{add}\; 3 \;\texttt{add}\; 4 \to 5 \;\texttt{add}\; 4}\,\texttt{(+e1)} \qquad \dfrac{\dfrac{}{5 \;\texttt{add}\; 4 \to 9}\,\texttt{(+v)} \qquad 9 \to_0 9}{5 \;\texttt{add}\; 4 \to_1 9}\,\texttt{(s1)}}{2 \;\texttt{add}\; 3 \;\texttt{add}\; 4 \to_2 9}\,\texttt{(s1)}$$

An example:

$$
\dfrac{
\dfrac{\dfrac{}{2\ \mathtt{add}\ 3 \rightarrow 5}\,(\mathtt{+v})}{2\ \mathtt{add}\ 3\ \mathtt{add}\ 4 \rightarrow 5\ \mathtt{add}\ 4}\,(\mathtt{+e1})
\qquad
\dfrac{\dfrac{}{5\ \mathtt{add}\ 4 \rightarrow 9}\,(\mathtt{+v}) \quad \dfrac{}{9 \rightarrow_0 9}\,(\mathtt{s0})}{5\ \mathtt{add}\ 4 \rightarrow_1 9}\,(\mathtt{s1})
}{2\ \mathtt{add}\ 3\ \mathtt{add}\ 4 \rightarrow_2 9}\,(\mathtt{s1})
$$

Is this the only derivation?

Another example:

$$
\dfrac{
  \dfrac{
    \dfrac{}{3\ \texttt{add}\ 4 \longrightarrow 7}\,(\texttt{+v})
  }{
    2\ \texttt{add}\ 3\ \texttt{add}\ 4 \longrightarrow 2\ \texttt{add}\ 7
  }\,(\texttt{+e2})
  \qquad
  \dfrac{
    \dfrac{}{2\ \texttt{add}\ 7 \longrightarrow 9}\,(\texttt{+v})
    \quad
    \dfrac{}{9 \longrightarrow_0 9}\,(\texttt{s0})
  }{
    2\ \texttt{add}\ 7 \longrightarrow_1 9
  }\,(\texttt{s1})
}{
  2\ \texttt{add}\ 3\ \texttt{add}\ 4 \longrightarrow_2 9
}\,(\texttt{s1})
$$

Can we decrease the number of rules in our semantics?

Semantics 1

$$\frac{}{v_1 \ \texttt{add} \ v_2 \rightarrow v_1 + v_2} \ (+V)$$

$$\frac{}{v_1 \ \texttt{sub} \ v_2 \rightarrow v_1 - v_2} \ (-V)$$

$$\frac{e_2 \rightarrow e_2'}{v_1 \ \texttt{add} \ e_2 \rightarrow v_1 \ \texttt{add} \ e_2'} \ (+e)$$

$$\frac{e_2 \rightarrow e_2'}{v_1 \ \texttt{sub} \ e_2 \rightarrow v_1 \ \texttt{sub} \ e_2'} \ (-e)$$

$$\frac{}{e \rightarrow^0 e} \ (s0)$$

$$\frac{e \rightarrow e' \qquad e' \rightarrow^k e''}{e \rightarrow^{k+1} e''} \ (s1)$$

Semantics 2

$$\frac{}{v_1 \text{ add } v_2 \rightarrow v_1 + v_2}\ (+V)$$

$$\frac{}{v_1 \text{ sub } v_2 \rightarrow v_1 - v_2}\ (-V)$$

$$\frac{e_1 \rightarrow e_1'}{e_1 \text{ add } e_2 \rightarrow e_1' \text{ add } e_2}\ (+e)$$

$$\frac{e_1 \rightarrow e_1'}{e_1 \text{ sub } e_2 \rightarrow e_1' \text{ sub } e_2}\ (-e)$$

$$\frac{}{e \rightarrow^{0} e}\ (s0)$$

$$\frac{e \rightarrow e' \qquad e' \rightarrow^{k} e''}{e \rightarrow^{k+1} e''}\ (s1)$$

Grammar vs operational semantics

- We can use the shape of programs to choose the “right” semantics:

```
<expr> ::= nat <addop> <expr> | nat
<addop>::= add | sub
```

$$\frac{}{v_1 \text{ add } v_2 \rightarrow v_1+v_2}\ (+V)$$

$$\frac{}{v_1 \text{ sub } v_2 \rightarrow v_1-v_2}\ (-V)$$

$$\frac{e_2 \rightarrow e_2'}{v_1 \text{ add } e_2 \rightarrow v_1 \text{ add } e_2'}\ (+e)$$

$$\frac{e_2 \rightarrow e_2'}{v_1 \text{ sub } e_2 \rightarrow v_1 \text{ sub } e_2'}\ (-e)$$

$$\frac{}{e \rightarrow^{0} e}\ (s0)$$

$$\frac{e \rightarrow e' \qquad e' \rightarrow^{k} e''}{e \rightarrow^{k+1} e''}\ (s1)$$

Boolean expressions

- Let us consider this simple language for Boolean expressions

```
<bexpr> ::= <const> <bop> <bexpr> | <const>
<bop>::= and | or | eq
<const>::= bool | int
```

What are the challenges here?

Operational semantics for basic boolean expressions

$$e \longrightarrow ?$$

Here the expression `e` is itself a configuration. We already have all the information we need to execute it.

What can `?` be?

Operational semantics for basic boolean expressions

$$\texttt{e} \longrightarrow \texttt{?}$$

Here the expression `e` is itself a configuration. We already have all the information we need to execute it.

What can `?` be?

$$\texttt{e} \longrightarrow \texttt{e'} \qquad\qquad \texttt{e} \longrightarrow \texttt{err}$$

Rules

c here is a configuration, either an expression e or err

$$\frac{}{c \rightarrow^{0} c}$$

$$\frac{c \rightarrow c' \qquad c' \rightarrow^{k} c''}{c \rightarrow^{k+1} c''}$$

$$\frac{v_1 \ v_2 \text{ different type}}{v_1 \text{ eq } v_2 \rightarrow \text{err}}$$

$$\frac{v_1 \ v_2 \text{ same type}}{v_1 \text{ eq } v_2 \rightarrow v_1 = v_2}$$

$$\frac{v_1 \ v_2 \text{ bool}}{v_1 \text{ and } v_2 \rightarrow v_1 \wedge v_2}$$

$$\frac{v_1 \ v_2 \text{ bool}}{v_1 \text{ or } v_2 \rightarrow v_1 \vee v_2}$$

$$\frac{e_1 \rightarrow e_1' \qquad e_1' \neq \text{err}}{e_1 \text{ bop } e_2 \rightarrow e_1' \text{ bop } e_2}$$

$$\frac{e_2 \rightarrow e_2' \qquad e_2' \neq \text{err}}{v_1 \text{ bop } e_2 \rightarrow v_1 \text{ bop } e_2'}$$

$$\frac{v_1 \ v_2 \text{ not bool}}{v_1 \text{ and } v_2 \rightarrow \text{err}}$$

$$\frac{v_1 \ v_2 \text{ not bool}}{v_1 \text{ or } v_2 \rightarrow \text{err}}$$

$$\frac{e_1 \rightarrow \text{err}}{e_1 \text{ bop } e_2 \rightarrow \text{err}}$$

$$\frac{e_2 \rightarrow \text{err}}{v_1 \text{ bop } e_2 \rightarrow \text{err}}$$

What can we do to have a more efficient semantics for boolean expressions?

What can we do to have a more efficient semantics for boolean expressions?

What if we know that one of the elements of an or is true or one of the elements of an and is false?

More efficient rules

$$\frac{\texttt{e}_2 \longrightarrow \texttt{e}_2'}{\texttt{v}_1 \texttt{ bop } \texttt{e}_2 \longrightarrow \texttt{v}_1 \texttt{ bop } \texttt{e}_2'}$$

We could change this rule:

$$\frac{}{\texttt{true or } \texttt{e}_2 \longrightarrow \texttt{true}}$$

$$\frac{\texttt{e}_2 \longrightarrow \texttt{e}_2'}{\texttt{false or } \texttt{e}_2 \longrightarrow \texttt{false or } \texttt{e}_2'}$$

Are the two semantics equivalent?

What if we want to check the second branch first?