The SPL Language: Syntax

$x \in$ Var set of integer variables
$v \in$ Z set of integer constants
$\ell \in$ Lab set of labels

$a \in$ AExp set of arithmetic expressions
$b \in$ BExp set of boolean expressions
$s \in$ Stmt set of statements

x, a, b, s are called meta-variables

$a ::= v \mid x \mid a_1 + a_2 \mid a_1 - a_2 \mid a_1 * a_2$

$b ::= \text{true} \mid \text{false} \mid \neg b \mid b_1 \wedge b_2 \mid b_1 \vee b_2 \mid a_1 = a_2 \mid a_1 \leq a_2$

$s ::= x := a^\ell \mid \text{skip}^\ell \mid s_1;s_2 \mid \text{if } b^\ell \text{ then } s_1 \text{ else } s_2 \mid \text{while } b^\ell \text{ do } s$

- variables are not declared
- expressions have no side-effects, all side-effects in statements
- only basic statements: no functions, heap, exceptions,…
- semantics usually specified at abstract syntax level

Grammars

$$
\begin{aligned}
Z &\to 0 \mid 1 \mid -1 \mid 2 \mid -2 \mid \dots \\
Var &\to x \mid y \mid z \mid \dots \\
A &\to Z \mid Var \mid A + A \mid A - A \mid A * A \\
B &\to true \mid false \mid \neg B \mid B \wedge B \mid B \vee B \mid A = A \mid A \leq A \\
S &\to skip \mid Var := A \mid S; S \mid if\ \ B\ \ then\ S\ \ else\ S \mid while\ B\ \ do\ S
\end{aligned}
$$

- Z, Var, A, B, S are non-terminals
- To derive a program, we start with a non-terminal and use the rules until there is no non-terminal left

$$
\begin{aligned}
Z &\rightarrow 0 \mid 1 \mid -1 \mid 2 \mid -2 \mid \dots \\
Var &\rightarrow x \mid y \mid z \mid \dots \\
A &\rightarrow Z \mid Var \mid A + A \mid A - A \mid A * A \\
B &\rightarrow true \mid false \mid \neg B \mid B \wedge B \mid B \vee B \mid A = A \mid A \leq A \\
S &\rightarrow skip \mid Var := A \mid S; S \mid if\ \ B\ \ then\ S\ \ else\ \ B \mid while\ \ B\ \ do\ \ S
\end{aligned}
$$

$S \rightarrow$

$$
\begin{aligned}
Z &\rightarrow 0 \mid 1 \mid -1 \mid 2 \mid -2 \mid \dots \\
Var &\rightarrow x \mid y \mid z \mid \dots \\
A &\rightarrow Z \mid Var \mid A + A \mid A - A \mid A * A \\
B &\rightarrow true \mid false \mid \neg B \mid B \wedge B \mid B \vee B \mid A = A \mid A \leq A \\
S &\rightarrow skip \mid Var := A \mid S; S \mid if\ B\ then\ S\ else\ B \mid while\ B\ do\ S
\end{aligned}
$$

$S \rightarrow while\ B\ do\ S \rightarrow$

$$
\begin{aligned}
Z &\to 0 \mid 1 \mid -1 \mid 2 \mid -2 \mid \ldots \\
Var &\to x \mid y \mid z \mid \ldots \\
A &\to Z \mid Var \mid A + A \mid A - A \mid A * A \\
B &\to true \mid false \mid \neg B \mid B \wedge B \mid B \vee B \mid A = A \mid A \leq A \\
S &\to skip \mid Var := A \mid S; S \mid if\ \ B\ \ then\ S\ \ else\ \ B \mid while\ \ B\ \ do\ \ S
\end{aligned}
$$

$S \to while\ \ B\ \ do\ \ S \to while\ \ A \leq A\ \ do\ \ S \to$

$$
\begin{aligned}
Z &\rightarrow 0 \mid 1 \mid -1 \mid 2 \mid -2 \mid \ldots \\
Var &\rightarrow x \mid y \mid z \mid \ldots \\
A &\rightarrow Z \mid Var \mid A + A \mid A - A \mid A * A \\
B &\rightarrow true \mid false \mid \neg B \mid B \wedge B \mid B \vee B \mid A = A \mid A \leq A \\
S &\rightarrow skip \mid Var := A \mid S; S \mid if\ B\ then\ S\ else\ B \mid while\ B\ do\ S
\end{aligned}
$$

$$S \rightarrow while\ B\ do\ S \rightarrow while\ A \leq A\ do\ S \rightarrow while\ A + A \leq A\ do\ S$$

$$
\begin{aligned}
Z &\to 0 \mid 1 \mid -1 \mid 2 \mid -2 \mid \dots \\
Var &\to x \mid y \mid z \mid \dots \\
A &\to Z \mid Var \mid A + A \mid A - A \mid A * A \\
B &\to true \mid false \mid \neg B \mid B \wedge B \mid B \vee B \mid A = A \mid A \leq A \\
S &\to skip \mid Var := A \mid S; S \mid if\ B\ then\ S\ else\ B \mid while\ B\ do\ S
\end{aligned}
$$

$$
\begin{aligned}
S &\to while\ B\ do\ S \to while\ A \leq A\ do\ S \to while\ A + A \leq A\ do\ S \\
&\quad while\ Var + A \leq A\ do\ S \to
\end{aligned}
$$

$$
\begin{aligned}
Z &\to 0 \mid 1 \mid -1 \mid 2 \mid -2 \mid \ldots \\
Var &\to x \mid y \mid z \mid \ldots \\
A &\to Z \mid Var \mid A + A \mid A - A \mid A * A \\
B &\to true \mid false \mid \neg B \mid B \wedge B \mid B \vee B \mid A = A \mid A \leq A \\
S &\to skip \mid Var := A \mid S; S \mid if\ B\ then\ S\ else\ B \mid while\ B\ do\ S
\end{aligned}
$$

$$
\begin{aligned}
S \to\ & while\ B\ do\ S \to while\ A \leq A\ do\ S \to while\ A + A \leq A\ do\ S \\
& while\ Var + A \leq A\ do\ S \to while\ x + A \leq A\ do\ S \to
\end{aligned}
$$

$$
\begin{aligned}
Z &\rightarrow 0 \mid 1 \mid -1 \mid 2 \mid -2 \mid \dots \\
Var &\rightarrow x \mid y \mid z \mid \dots \\
A &\rightarrow Z \mid Var \mid A + A \mid A - A \mid A * A \\
B &\rightarrow true \mid false \mid \neg B \mid B \wedge B \mid B \vee B \mid A = A \mid A \leq A \\
S &\rightarrow skip \mid Var := A \mid S; S \mid if\ B\ then\ S\ else\ B \mid while\ B\ do\ S
\end{aligned}
$$

$$
\begin{aligned}
S \rightarrow\ & while\ B\ do\ S \rightarrow while\ A \leq A\ do\ S \rightarrow while\ A + A \leq A\ do\ S \\
& while\ Var + A \leq A\ do\ S \rightarrow while\ x + A \leq A\ do\ S \rightarrow \\
& while\ x + Z \leq A\ do\ S \rightarrow
\end{aligned}
$$

$$
\begin{aligned}
Z &\to 0 \mid 1 \mid -1 \mid 2 \mid -2 \mid \dots \\
Var &\to x \mid y \mid z \mid \dots \\
A &\to Z \mid Var \mid A + A \mid A - A \mid A * A \\
B &\to true \mid false \mid \neg B \mid B \wedge B \mid B \vee B \mid A = A \mid A \leq A \\
S &\to skip \mid Var := A \mid S; S \mid if\ \ B\ \ then\ S\ \ else\ B \mid while\ B\ do\ S
\end{aligned}
$$

$$
\begin{aligned}
S &\to while\ B\ do\ S \to while\ A \leq A\ do\ S \to while\ A + A \leq A\ do\ S \\
&\quad while\ Var + A \leq A\ do\ S \to while\ x + A \leq A\ do\ S \to \\
&\quad while\ x + Z \leq A\ do\ S \to while\ x + 3 \leq A\ do\ S \to
\end{aligned}
$$

$$
\begin{aligned}
Z &\to 0 \mid 1 \mid -1 \mid 2 \mid -2 \mid \dots \\
\mathit{Var} &\to x \mid y \mid z \mid \dots \\
A &\to Z \mid \mathit{Var} \mid A + A \mid A - A \mid A * A \\
B &\to \mathit{true} \mid \mathit{false} \mid \neg B \mid B \wedge B \mid B \vee B \mid A = A \mid A \leq A \\
S &\to \mathit{skip} \mid \mathit{Var} := A \mid S; S \mid \mathit{if}\ B\ \mathit{then}\ S\ \mathit{else}\ B \mid \mathit{while}\ B\ \mathit{do}\ S
\end{aligned}
$$

$$
\begin{aligned}
S &\to \mathit{while}\ B\ \mathit{do}\ S \to \mathit{while}\ A \leq A\ \mathit{do}\ S \to \mathit{while}\ A + A \leq A\ \mathit{do}\ S \\
&\quad \mathit{while}\ \mathit{Var} + A \leq A\ \mathit{do}\ S \to \mathit{while}\ x + A \leq A\ \mathit{do}\ S \to \\
&\quad \mathit{while}\ x + Z \leq A\ \mathit{do}\ S \to \mathit{while}\ x + 3 \leq A\ \mathit{do}\ S \to \\
&\quad \mathit{while}\ x + 3 \leq \mathit{Var}\ \mathit{do}\ S \to
\end{aligned}
$$

$$
\begin{aligned}
Z &\rightarrow 0 \mid 1 \mid -1 \mid 2 \mid -2 \mid \dots \\
Var &\rightarrow x \mid y \mid z \mid \dots \\
A &\rightarrow Z \mid Var \mid A + A \mid A - A \mid A * A \\
B &\rightarrow true \mid false \mid \neg B \mid B \wedge B \mid B \vee B \mid A = A \mid A \leq A \\
S &\rightarrow skip \mid Var := A \mid S; S \mid if\ \ B\ then\ S\ else\ B \mid while\ B\ do\ S
\end{aligned}
$$

$$
\begin{aligned}
S \rightarrow\ & while\ B\ do\ S \rightarrow while\ A \leq A\ do\ S \rightarrow while\ A + A \leq A\ do\ S \\
& while\ Var + A \leq A\ do\ S \rightarrow while\ x + A \leq A\ do\ S \rightarrow \\
& while\ x + Z \leq A\ do\ S \rightarrow while\ x + 3 \leq A\ do\ S \rightarrow \\
& while\ x + 3 \leq Var\ do\ S \rightarrow while\ x + 3 \leq y\ do\ S \rightarrow
\end{aligned}
$$

$$
\begin{aligned}
Z &\to 0 \mid 1 \mid -1 \mid 2 \mid -2 \mid \dots \\
Var &\to x \mid y \mid z \mid \dots \\
A &\to Z \mid Var \mid A + A \mid A - A \mid A * A \\
B &\to true \mid false \mid \neg B \mid B \wedge B \mid B \vee B \mid A = A \mid A \leq A \\
S &\to skip \mid Var := A \mid S; S \mid if\ B\ then\ S\ else\ B \mid while\ B\ do\ S
\end{aligned}
$$

$$
\begin{aligned}
S \to\ & while\ B\ do\ S \to while\ A \leq A\ do\ S \to while\ A + A \leq A\ do\ S \\
& while\ Var + A \leq A\ do\ S \to while\ x + A \leq A\ do\ S \to \\
& while\ x + Z \leq A\ do\ S \to while\ x + 3 \leq A\ do\ S \to \\
& while\ x + 3 \leq Var\ do\ S \to while\ x + 3 \leq y\ do\ S \to \\
& while\ x + 3 \leq y\ do\ Var := A \to
\end{aligned}
$$

$$
\begin{aligned}
Z &\to 0 \mid 1 \mid -1 \mid 2 \mid -2 \mid \dots \\
Var &\to x \mid y \mid z \mid \dots \\
A &\to Z \mid Var \mid A + A \mid A - A \mid A * A \\
B &\to true \mid false \mid \neg B \mid B \wedge B \mid B \vee B \mid A = A \mid A \leq A \\
S &\to skip \mid Var := A \mid S; S \mid if\ B\ then\ S\ else\ B \mid while\ B\ do\ S
\end{aligned}
$$

$$
\begin{aligned}
S \to\ & while\ B\ do\ S \to while\ A \leq A\ do\ S \to while\ A + A \leq A\ do\ S \\
& while\ Var + A \leq A\ do\ S \to while\ x + A \leq A\ do\ S \to \\
& while\ x + Z \leq A\ do\ S \to while\ x + 3 \leq A\ do\ S \to \\
& while\ x + 3 \leq Var\ do\ S \to while\ x + 3 \leq y\ do\ S \to \\
& while\ x + 3 \leq y\ do\ Var := A \to while\ x + 3 \leq y\ do\ x := A \to
\end{aligned}
$$

$$
\begin{aligned}
Z &\to 0 \mid 1 \mid -1 \mid 2 \mid -2 \mid \dots \\
Var &\to x \mid y \mid z \mid \dots \\
A &\to Z \mid Var \mid A + A \mid A - A \mid A * A \\
B &\to true \mid false \mid \neg B \mid B \wedge B \mid B \vee B \mid A = A \mid A \leq A \\
S &\to skip \mid Var := A \mid S; S \mid if\ \ B\ then\ S\ else\ B \mid while\ B\ do\ S
\end{aligned}
$$

$$
\begin{aligned}
S \to\ & while\ B\ do\ S \to while\ A \leq A\ do\ S \to while\ A + A \leq A\ do\ S \\
& while\ Var + A \leq A\ do\ S \to while\ x + A \leq A\ do\ S \to \\
& while\ x + Z \leq A\ do\ S \to while\ x + 3 \leq A\ do\ S \to \\
& while\ x + 3 \leq Var\ do\ S \to while\ x + 3 \leq y\ do\ S \to \\
& while\ x + 3 \leq y\ do\ Var := A \to while\ x + 3 \leq y\ do\ x := A \to \\
& while\ x + 3 \leq y\ do\ x := A + A \to
\end{aligned}
$$

$$
\begin{aligned}
Z &\rightarrow 0 \mid 1 \mid -1 \mid 2 \mid -2 \mid \ldots \\
Var &\rightarrow x \mid y \mid z \mid \ldots \\
A &\rightarrow Z \mid Var \mid A + A \mid A - A \mid A * A \\
B &\rightarrow true \mid false \mid \neg B \mid B \wedge B \mid B \vee B \mid A = A \mid A \leq A \\
S &\rightarrow skip \mid Var := A \mid S; S \mid if\ \ B\ then\ S\ else\ B \mid while\ B\ do\ S
\end{aligned}
$$

$$
\begin{aligned}
S \rightarrow\ & while\ B\ do\ S \rightarrow while\ A \leq A\ do\ S \rightarrow while\ A + A \leq A\ do\ S \\
& while\ Var + A \leq A\ do\ S \rightarrow while\ x + A \leq A\ do\ S \rightarrow \\
& while\ x + Z \leq A\ do\ S \rightarrow while\ x + 3 \leq A\ do\ S \rightarrow \\
& while\ x + 3 \leq Var\ do\ S \rightarrow while\ x + 3 \leq y\ do\ S \rightarrow \\
& while\ x + 3 \leq y\ do\ Var := A \rightarrow while\ x + 3 \leq y\ do\ x := A \rightarrow \\
& while\ x + 3 \leq y\ do\ x := A + A \rightarrow while\ x + 3 \leq y\ do\ x := Var + A \rightarrow
\end{aligned}
$$

$$Z \to 0 \mid 1 \mid -1 \mid 2 \mid -2 \mid \dots$$
$$Var \to x \mid y \mid z \mid \dots$$
$$A \to Z \mid Var \mid A + A \mid A - A \mid A * A$$
$$B \to true \mid false \mid \neg B \mid B \wedge B \mid B \vee B \mid A = A \mid A \leq A$$
$$S \to skip \mid Var := A \mid S; S \mid if\ B\ then\ S\ else\ B \mid while\ B\ do\ S$$

$$\begin{aligned}
S \to\ & while\ B\ do\ S \to while\ A \leq A\ do\ S \to while\ A + A \leq A\ do\ S \\
& while\ Var + A \leq A\ do\ S \to while\ x + A \leq A\ do\ S \to \\
& while\ x + Z \leq A\ do\ S \to while\ x + 3 \leq A\ do\ S \to \\
& while\ x + 3 \leq Var\ do\ S \to while\ x + 3 \leq y\ do\ S \to \\
& while\ x + 3 \leq y\ do\ Var := A \to while\ x + 3 \leq y\ do\ x := A \to \\
& while\ x + 3 \leq y\ do\ x := A + A \to while\ x + 3 \leq y\ do\ x := Var + A \to \\
& while\ x + 3 \leq y\ do\ x := x + A \to
\end{aligned}$$

$$
\begin{aligned}
Z &\to 0 \mid 1 \mid -1 \mid 2 \mid -2 \mid \dots \\
Var &\to x \mid y \mid z \mid \dots \\
A &\to Z \mid Var \mid A + A \mid A - A \mid A * A \\
B &\to true \mid false \mid \neg B \mid B \wedge B \mid B \vee B \mid A = A \mid A \leq A \\
S &\to skip \mid Var := A \mid S; S \mid if\ \ B\ then\ S\ else\ B \mid while\ B\ do\ S
\end{aligned}
$$

$$
\begin{aligned}
S \to\ & while\ B\ do\ S \to while\ A \leq A\ do\ S \to while\ A + A \leq A\ do\ S \\
& while\ Var + A \leq A\ do\ S \to while\ x + A \leq A\ do\ S \to \\
& while\ x + Z \leq A\ do\ S \to while\ x + 3 \leq A\ do\ S \to \\
& while\ x + 3 \leq Var\ do\ S \to while\ x + 3 \leq y\ do\ S \to \\
& while\ x + 3 \leq y\ do\ Var := A \to while\ x + 3 \leq y\ do\ x := A \to \\
& while\ x + 3 \leq y\ do\ x := A + A \to while\ x + 3 \leq y\ do\ x := Var + A \to \\
& while\ x + 3 \leq y\ do\ x := x + A \to while\ x + 3 \leq y\ do\ x := x + Z \to
\end{aligned}
$$

$$
\begin{aligned}
Z &\to 0 \mid 1 \mid -1 \mid 2 \mid -2 \mid \ldots \\
Var &\to x \mid y \mid z \mid \ldots \\
A &\to Z \mid Var \mid A + A \mid A - A \mid A * A \\
B &\to true \mid false \mid \neg B \mid B \wedge B \mid B \vee B \mid A = A \mid A \leq A \\
S &\to skip \mid Var := A \mid S; S \mid if\ B\ then\ S\ else\ B \mid while\ B\ do\ S
\end{aligned}
$$

$$
\begin{aligned}
S \to\ & while\ B\ do\ S \to while\ A \leq A\ do\ S \to while\ A + A \leq A\ do\ S \\
& while\ Var + A \leq A\ do\ S \to while\ x + A \leq A\ do\ S \to \\
& while\ x + Z \leq A\ do\ S \to while\ x + 3 \leq A\ do\ S \to \\
& while\ x + 3 \leq Var\ do\ S \to while\ x + 3 \leq y\ do\ S \to \\
& while\ x + 3 \leq y\ do\ Var := A \to while\ x + 3 \leq y\ do\ x := A \to \\
& while\ x + 3 \leq y\ do\ x := A + A \to while\ x + 3 \leq y\ do\ x := Var + A \to \\
& while\ x + 3 \leq y\ do\ x := x + A \to while\ x + 3 \leq y\ do\ x := x + Z \to \\
& while\ x + 3 \leq y\ do\ x := x + 2
\end{aligned}
$$

Operational Semantics

- Specifies how expressions and statements should be evaluated

- Evaluation depends on the shape of the expression/statement:
  - `1,2,3`, … do not evaluate any further
  - `x + y` is evaluated further

- Think of it as an interpreter

Operational Semantics

- Evaluation depends on values of variables
  - what does x + y evaluate to ?
  - depends on the values of x and y

- Values of variables at any moment in time are given by a function $\sigma \in$ Store = Var $\longrightarrow$ Z
  - Z is the set of integers
  - to simplify presentation we assume Store denotes total functions
  - if σ is such that `x ↦ 5` and `y ↦ 3`, then `x + y` is 8

Operational Semantics for SPL

- Configurations: $c \in \Sigma$ where $\Sigma = (\text{Stmt} \times \text{Store}) \cup \text{Store}$
  - $<S, \sigma>$ is a configuration
  - σ is also a configuration: a terminal configuration. All other configurations are non-terminal

- Transitions: $\rightarrow \subseteq \Sigma \times \Sigma$
  - steps between configurations

- Transition system: $(\Sigma, \rightarrow, I, F)$
  - $I \subseteq \Sigma$: initial configurations
  - $F \subseteq$ Store: final configurations

Operational Semantics for SPL

- We write $c \rightarrow c'$ when $(c, c') \in \rightarrow$
- $\rightarrow^*$ denotes the reflexive transitive closure of the relation $\rightarrow$. We say $c \rightarrow^* c'$ when:
  - $c = c_0$ and $c_n = c'$
  - there is a sequence $c_0 \rightarrow c_1 \rightarrow \ldots c_n$ for some $n \geq 0$

Notation: Rules of Inference

These are called evaluation rules

$$\frac{\text{Hypothesis}_1 \ \ldots \ \text{Hypothesis}_n}{\text{Conclusion}}$$

Example:

$$\frac{A \text{ is true} \qquad B \text{ is true}}{A \wedge B \text{ is true}}$$

Evaluation rules with no premises are called axioms

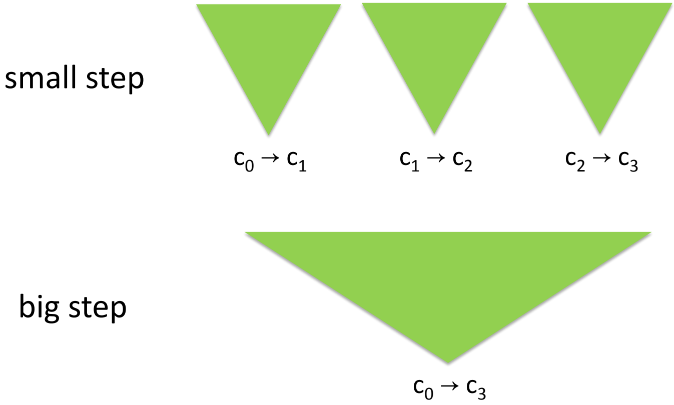
$$\frac{}{\text{Conclusion}}$$

Next: operational semantics of SPL

Operational Semantics of SPL

- There are two kinds: big-step and small-step
- Big-step
  - $c \longrightarrow c'$ describes the entire computation
- Small-step
  - $c \longrightarrow c'$ describes a single step of a larger computation

Small Step vs. Big Step



Operational Semantics of SPL

Next, we will give semantics of SPL. The statements will be evaluated in a small-step style, while the expressions will be evaluated in big-step style.

Auxiliary Relations

- To describe the semantics of AExp and BExp we use two auxiliary relations

$$\texttt{for AExp: } \Downarrow_a \subseteq (\texttt{AExp} \times \texttt{Store}) \times \texttt{Z}$$
$$\texttt{for BExp: } \Downarrow_b \subseteq (\texttt{BExp} \times \texttt{Store}) \times \{\texttt{true, false}\}$$

- Judgments such as

$$\langle a , \sigma \rangle \Downarrow_a v$$

are read as: "expression *a* evaluates to *v* in store σ"
Boolean expressions read similarly

Evaluation rules for AExp

$$\frac{\langle a_1 , \sigma \rangle \Downarrow_a v_1 \qquad \langle a_2 , \sigma \rangle \Downarrow_a v_2}{\langle a_1 + a_2 , \sigma \rangle \Downarrow_a v_1 + v_2}$$

$$\frac{}{\langle x , \sigma \rangle \Downarrow_a \sigma(x)}$$

Evaluation rules for BExp

$$\frac{\langle a_1, \sigma\rangle \Downarrow_a v_1 \qquad \langle a_2, \sigma\rangle \Downarrow_a v_2}{\langle a_1 \leq a_2, \sigma\rangle \Downarrow_b \text{bv}} \quad \text{bv is } v_1 \leq v_2$$

$$\frac{\langle a_1, \sigma\rangle \Downarrow_a v_1 \qquad \langle a_2, \sigma\rangle \Downarrow_a v_2}{\langle a_1 = a_2, \sigma\rangle \Downarrow_b \text{bv}} \quad \text{bv is } v_1 == v_2$$

$$\frac{}{\langle b_1 \wedge b_2, \sigma\rangle \Downarrow_b \ ???}$$

What about this ?

Evaluation rules for BExp

$$\frac{\langle a_1, \sigma\rangle \Downarrow_a v_1 \qquad \langle a_2, \sigma\rangle \Downarrow_a v_2}{\langle a_1 \le a_2, \sigma\rangle \Downarrow_b bv} \ bv \text{ is } v_1 \le v_2$$

$$\frac{\langle a_1, \sigma\rangle \Downarrow_a v_1 \qquad \langle a_2, \sigma\rangle \Downarrow_a v_2}{\langle a_1 = a_2, \sigma\rangle \Downarrow_b bv} \ bv \text{ is } v_1 == v_2$$

$$\frac{\langle b_1, \sigma\rangle \Downarrow_b \text{true} \qquad \langle b_2, \sigma\rangle \Downarrow_b \text{true}}{\langle b_1 \wedge b_2, \sigma\rangle \Downarrow_b \text{true}}$$

short-circuit evaluation

$$\frac{\langle b_1, \sigma\rangle \Downarrow_b \text{false}}{\langle b_1 \wedge b_2, \sigma\rangle \Downarrow_b \text{false}} \qquad \frac{\langle b_2, \sigma\rangle \Downarrow_b \text{false}}{\langle b_1 \wedge b_2, \sigma\rangle \Downarrow_b \text{false}}$$

How to read the rules

- Top-down: like inference rules
  - If we know hypothesis holds, conclusion holds
  - If $\langle x, \sigma \rangle \Downarrow_a 5$ and $\langle y, \sigma \rangle \Downarrow_a 6$ then $\langle x + y, \sigma \rangle \Downarrow_a 11$

- Bottom-up: read by inversion
  - Suppose we want to evaluate $\langle x + y, \sigma \rangle \Downarrow_a$
  - Lets look at rules with conclusion that has $\langle x + y, \sigma \rangle$
  - Here: only 1 rule has it as a conclusion (the addition rule)
  - Repeat a recursive tree-walk

Example: Derivation Tree

Evaluate this: $\langle$ (x + 3) $*$ (y + 4) , $\sigma\rangle$ where σ: x $\mapsto$ 1, y $\mapsto$ 2

Example: Derivation Tree

Evaluate this: $\langle\, (x + 3) * (y + 4)\, , \sigma\rangle$ where σ: $x \mapsto 1$, $y \mapsto 2$

$$\frac{}{\langle x , \sigma\rangle \Downarrow_a 1} \qquad \frac{}{\langle y , \sigma\rangle \Downarrow_a 2}$$

$$\frac{\langle x , \sigma\rangle \Downarrow_a 1 \qquad \langle 3 , \sigma\rangle \Downarrow_a 3}{\langle x + 3 , \sigma\rangle \Downarrow_a 4} \qquad \frac{\langle y , \sigma\rangle \Downarrow_a 2 \qquad \langle 4 , \sigma\rangle \Downarrow_a 4}{\langle y + 4 , \sigma\rangle \Downarrow_a 6}$$

$$\frac{\langle x + 3, \sigma\rangle \Downarrow_a 4 \qquad \langle y + 4 , \sigma\rangle \Downarrow_a 6}{\langle\, (x + 3) * (y + 4)\, , \sigma\rangle \Downarrow_a 24}$$

Evaluation of Statements

- Evaluating a statement produces a new store
  - $\langle s , \sigma\rangle \to \langle s' , \sigma'\rangle$
- Evaluation order is important
  - In $s_1 ; s_2$ s_1 is evaluated before s_2
  - In if true then s_1 else s_2 s_2 is not evaluated
- Some constructs have multiple rules
  - conditionals and while

Evaluation rules for Stmt I

$$\frac{\langle s_1 , \sigma\rangle \rightarrow \langle s_2 , \sigma_1\rangle}{\langle s_1 ; s_3 , \sigma\rangle \rightarrow \langle s_2 ; s_3 , \sigma_1\rangle}$$

$$\frac{\langle s_1 , \sigma\rangle \rightarrow \sigma_1}{\langle s_1 ; s_2 , \sigma\rangle \rightarrow \langle s_2 , \sigma_1\rangle}$$

$$\frac{}{\langle \text{skip}, \sigma\rangle \rightarrow \sigma}$$

$$\frac{\langle a , \sigma\rangle \Downarrow_a v}{\langle x{:=}\ a, \sigma\rangle \rightarrow \langle\ x{:=}\ v , \sigma\rangle}$$

$$\frac{}{\langle x{:=}\ v, \sigma\rangle \rightarrow \sigma[x \mapsto v]}$$

assignment not a single step

Evaluation rules for Stmt II

$$\frac{}{\langle \text{if true then } s_1 \text{ else } s_2, \sigma\rangle \rightarrow \langle s_1, \sigma\rangle} \qquad \frac{}{\langle \text{if false then } s_1 \text{ else } s_2, \sigma\rangle \rightarrow \langle s_2, \sigma\rangle}$$

$$\frac{\langle b_1, \sigma\rangle \Downarrow_b \text{bv}}{\langle \text{if } b_1 \text{ then } s_1 \text{ else } s_2, \sigma\rangle \rightarrow \langle \text{if bv then } s_1 \text{ else } s_2, \sigma\rangle}$$

$$\frac{}{\langle \text{while b do s}, \sigma\rangle \rightarrow \text{???}}$$

Evaluation rules for Stmt II

$$\frac{}{\langle \text{if true then } s_1 \text{ else } s_2, \sigma\rangle \to \langle s_1, \sigma\rangle} \qquad \frac{}{\langle \text{if false then } s_1 \text{ else } s_2, \sigma\rangle \to \langle s_2, \sigma\rangle}$$

$$\frac{\langle b_1, \sigma\rangle \Downarrow_b bv}{\langle \text{if } b_1 \text{ then } s_1 \text{ else } s_2, \sigma\rangle \to \langle \text{if } bv \text{ then } s_1 \text{ else } s_2, \sigma\rangle}$$

$$\frac{}{\langle \text{while } b \text{ do } s, \sigma\rangle \to \langle \text{if } b \text{ then } (s; \text{while } b \text{ do } s) \text{ else skip}, \sigma\rangle}$$

'while' expressed in terms of 'if'

Sequences

Note that for a program S_0 the steps are formed via the relation $\rightarrow$

That is, sequences are $\langle S_0 , \sigma_0 \rangle \rightarrow \langle S_1 , \sigma_1 \rangle \rightarrow \ldots$

The relations $\Downarrow_a$ or $\Downarrow_b$ are only used to justify the step with $\rightarrow$

In other words, $\Downarrow_a$ or $\Downarrow_b$ are only used to build the relation $\rightarrow$