Lecture 3 – Small-Step Operational Semantics

AAA551: Programming Language Theory

Jihyeok Park

2026 Spring

Recall

- Programming languages
- Mathematical preliminaries
  - Binary relations and functions
  - Inductively defined sets
- Big-step operational semantics
  - Example: VAE
  - Scala implementation
  - Derivation
- Simple imperative language – IMP
  - Big-step operational semantics of IMP
  - Equivalence of statements
  - Determinism

Contents

Contents

Operational Semantics

There are two main styles of operational semantics:

- **Big-Step Operational Semantics** defines the meaning of a program by specifying how it executes on a machine in **one big step**.

$$\frac{\cdots}{\vdash e \Rightarrow n}$$

(An expression e evaluates to n.)

- **Small-Step Operational Semantics** defines the meaning of a program by specifying how it executes on a machine **step-by-step**.

$$e \to e' \to e'' \to \ldots \to n$$

(An expression e is reduced to e', then to e'', and so on until n.)

Big-Step Operational Semantics – VAE

$$e ::= n \mid e \texttt{ + } e \mid e \texttt{ * } e \mid x \texttt{ := } e; \; e \mid x$$

$$\boxed{\sigma \vdash e \Rightarrow n}$$

$$\frac{}{\sigma \vdash n \Rightarrow n}$$

$$\frac{\sigma \vdash e_1 \Rightarrow n_1 \qquad \sigma \vdash e_2 \Rightarrow n_2}{\sigma \vdash e_1 \texttt{ + } e_2 \Rightarrow n_1 + n_2} \qquad \frac{\sigma \vdash e_1 \Rightarrow n_1 \qquad \sigma \vdash e_2 \Rightarrow n_2}{\sigma \vdash e_1 \texttt{ * } e_2 \Rightarrow n_1 \times n_2}$$

$$\frac{\sigma \vdash e_1 \Rightarrow n_1 \qquad \sigma[x \mapsto n_1] \vdash e_2 \Rightarrow n_2}{\sigma \vdash x \texttt{ := } e_1; \; e_2 \Rightarrow n_2} \qquad \frac{x \in \mathsf{dom}(\sigma)}{\sigma \vdash x \Rightarrow \sigma(x)}$$

Small-Step Operational Semantics – VAE

The small-step operational semantics of VAE is defined by the following **small-step evaluation relation** $\rightarrow$ on configurations $\langle \sigma, e \rangle$:

$$\boxed{\langle \sigma, e \rangle \rightarrow \langle \sigma, e \rangle}$$

where a **configuration** $\langle \sigma, e \rangle$ consists of

1. an environment $\sigma : \mathbb{X} \rightharpoonup \mathbb{Z}$ that maps variables to integers, and
2. an expression e.

Small-Step Operational Semantics – VAE

PLRG

$$\frac{\langle \sigma, e_1 \rangle \to \langle \sigma', e_1' \rangle}{\langle \sigma, e_1 \texttt{ + } e_2 \rangle \to \langle \sigma', e_1' \texttt{ + } e_2 \rangle}$$

$$\frac{\langle \sigma, e_1 \rangle \to \langle \sigma', e_1' \rangle}{\langle \sigma, e_1 \texttt{ * } e_2 \rangle \to \langle \sigma', e_1' \texttt{ * } e_2 \rangle}$$

$$\frac{\langle \sigma, e_2 \rangle \to \langle \sigma', e_2' \rangle}{\langle \sigma, n_1 \texttt{ + } e_2 \rangle \to \langle \sigma', n_1 \texttt{ + } e_2' \rangle}$$

$$\frac{\langle \sigma, e_2 \rangle \to \langle \sigma', e_2' \rangle}{\langle \sigma, n_1 \texttt{ * } e_2 \rangle \to \langle \sigma', n_1 \texttt{ * } e_2' \rangle}$$

$$\frac{}{\langle \sigma, n_1 \texttt{ + } n_2 \rangle \to \langle \sigma, n_1 + n_2 \rangle}$$

$$\frac{}{\langle \sigma, n_1 \texttt{ * } n_2 \rangle \to \langle \sigma, n_1 \times n_2 \rangle}$$

Small-Step Operational Semantics – VAE

$$\frac{\langle \sigma, e_1 \rangle \to \langle \sigma', e_1' \rangle}{\langle \sigma, x := e_1;\ e_2 \rangle \to \langle \sigma', x := e_1';\ e_2 \rangle}$$

$$\frac{}{\langle \sigma, x := n;\ e_2 \rangle \to \langle \sigma[x \mapsto n], e_2 \rangle}$$

$$\frac{x \in \mathsf{dom}(\sigma)}{\langle \sigma, x \rangle \to \langle \sigma, \sigma(x) \rangle}$$

Small-Step Operational Semantics – VAE

We can define the **multi-step evaluation** relation $\rightarrow^*$ as the reflexive and transitive closure of $\rightarrow$:

$$\frac{}{\langle \sigma, e \rangle \rightarrow^* \langle \sigma, e \rangle}$$

$$\frac{\langle \sigma, e \rangle \rightarrow^* \langle \sigma', e' \rangle \qquad \langle \sigma', e' \rangle \rightarrow \langle \sigma'', e'' \rangle}{\langle \sigma, e \rangle \rightarrow^* \langle \sigma'', e'' \rangle}$$

Derivation

$$
\begin{aligned}
\langle \varnothing, \mathtt{x := 1;\ \{y := 2;\ x + y\}} \rangle &\rightarrow \langle \sigma_0, \mathtt{y := 2;\ x + y} \rangle \\
&\rightarrow \langle \sigma_1, \mathtt{x + y} \rangle \\
&\rightarrow \langle \sigma_1, 1 + \mathtt{y} \rangle \\
&\rightarrow \langle \sigma_1, 1 + 2 \rangle \\
&\rightarrow \langle \sigma_1, 3 \rangle
\end{aligned}
$$

where

$$
\begin{aligned}
\sigma_0 &= [\mathtt{x} \mapsto 1] \\
\sigma_1 &= [\mathtt{x} \mapsto 1, \mathtt{y} \mapsto 2]
\end{aligned}
$$

Derivation

$$\begin{aligned}\langle \varnothing, \texttt{x := 1; } \{\texttt{y := 2; x + y}\}\rangle &\to \langle \sigma_0, \texttt{y := 2; x + y}\rangle \\ &\to \langle \sigma_1, \texttt{x + y}\rangle \\ &\to \langle \sigma_1, 1 \texttt{ + y}\rangle \\ &\to \langle \sigma_1, 1 \texttt{ + } 2\rangle \\ &\to \langle \sigma_1, 3\rangle\end{aligned}$$

where

$$\begin{aligned}\sigma_0 &= [\texttt{x} \mapsto 1] \\ \sigma_1 &= [\texttt{x} \mapsto 1, \texttt{y} \mapsto 2]\end{aligned}$$

We can also write the above derivation as a single multi-step evaluation:

$$\langle \varnothing, \texttt{x := 1; } \{\texttt{y := 2; x + y}\}\rangle \to^* \langle \sigma_1, 3\rangle$$

It means that the expression $\texttt{x := 1; } \{\texttt{y := 2; x + y}\}$ evaluates to 3.

Big-Step vs. Small-Step Operational Semantics

PLRG

Are big-step and small-step operational semantics equivalent?

Big-Step vs. Small-Step Operational Semantics

Are big-step and small-step operational semantics equivalent?

No! Consider the following expression:

$$\texttt{x := 1; } \{ \texttt{ x := 2; x } \} \texttt{ + x}$$

- It evaluates to 3 under the big-step operational semantics, but
- it evaluates to 4 under the small-step operational semantics.

Big-Step vs. Small-Step Operational Semantics

Are big-step and small-step operational semantics equivalent?

No! Consider the following expression:

$$\texttt{x := 1; \{ x := 2; x \} + x}$$

- It evaluates to 3 under the big-step operational semantics, but
- it evaluates to 4 under the small-step operational semantics.

Why?

$$\frac{\dfrac{\dots}{[\texttt{x} \mapsto 1] \vdash \texttt{x := 2; x} \Rightarrow 2} \qquad \dfrac{\dots}{[\texttt{x} \mapsto 1] \vdash \texttt{x} \Rightarrow 1}}{[\texttt{x} \mapsto 1] \vdash \{\texttt{x := 2; x}\} \texttt{ + x} \Rightarrow 3}$$

vs.

$$\langle [\texttt{x} \mapsto 1], \{\texttt{x := 2; x}\} \texttt{ + x} \rangle \rightarrow \langle [\texttt{x} \mapsto 2], \texttt{x + x} \rangle \rightarrow^* \langle [\texttt{x} \mapsto 2], 4 \rangle$$

Big-Step vs. Small-Step Operational Semantics

To make big-step and small-step operational semantics equivalent, we need to use **substitution** instead of **environments** in the small-step operational semantics:

$$\frac{}{\langle x := n;\ e_2 \rangle \to \langle e_2[x \mapsto n] \rangle}$$

where $e[x \mapsto y]$ is the expression obtained by substituting x with y in e.

Big-Step vs. Small-Step Operational Semantics

To make big-step and small-step operational semantics equivalent, we need to use **substitution** instead of **environments** in the small-step operational semantics:

$$\frac{}{\langle x \ := n;\ e_2 \rangle \to \langle e_2[x \mapsto n] \rangle}$$

where $e[x \mapsto y]$ is the expression obtained by substituting x with y in e.

With substitution, the above expression evaluates to 3 under the small-step operational semantics as well:

$$\langle \mathtt{x} \ := 1;\ \{\mathtt{x} \ := 2;\ \mathtt{x}\} + \mathtt{x} \rangle \to \langle \{\mathtt{x} \ := 2;\ \mathtt{x}\} + 1 \rangle \to \langle 2 + 1 \rangle \to 3$$

Big-Step vs. Small-Step Operational Semantics

To make big-step and small-step operational semantics equivalent, we need to use **substitution** instead of **environments** in the small-step operational semantics:

$$\frac{}{\langle x \texttt{ := } n;\ e_2 \rangle \to \langle e_2[x \mapsto n] \rangle}$$

where $e[x \mapsto y]$ is the expression obtained by substituting x with y in e.

With substitution, the above expression evaluates to 3 under the small-step operational semantics as well:

$$\langle \texttt{x := 1;}\ \{\texttt{x := 2;}\ \texttt{x}\} \texttt{ + x} \rangle \to \langle \{\texttt{x := 2;}\ \texttt{x}\} \texttt{ + 1} \rangle \to \langle \texttt{2 + 1} \rangle \to 3$$

This demonstrates that representing **lexical scope** through substitution is often more intuitive than using environments when defining small-step operational semantics.

We will discuss substitution in more detail in the next lecture.

Contents

Soundness

Let's prove the **soundness** of the small-step operational semantics of VAE.

- **Soundness**: Every configuration evaluates to an integer:

$$\forall \sigma \in \Sigma.\ \forall e \in \mathbb{E}.\ \exists \sigma' \in \Sigma.\ \exists n' \in \mathbb{Z}.\ \langle \sigma, e \rangle \to^* \langle \sigma', n' \rangle$$

Is it true?

Soundness

Let's prove the **soundness** of the small-step operational semantics of VAE.

- **Soundness**: Every configuration evaluates to an integer:

$$\forall \sigma \in \Sigma.\ \forall e \in \mathbb{E}.\ \exists \sigma' \in \Sigma.\ \exists n' \in \mathbb{Z}.\ \langle \sigma, e \rangle \to^* \langle \sigma', n' \rangle$$

Is it true? **No!** Consider the following configuration:

$$\langle \varnothing, \mathtt{x} \rangle \not\to$$

where $\langle \sigma, e \rangle \not\to$ is shorthand for:

$$\neg(\exists \sigma' \in \Sigma.\ \exists e' \in \mathbb{E}.\ \langle \sigma, e \rangle \to \langle \sigma', e' \rangle)$$

Well-Formed Configurations

We can fix the soundness issue by restricting the evaluation to **well-formed** configurations.

Definition (Well-Formed Expressions)

A configuration $\langle \sigma, e \rangle$ is **well-formed** if and only if $\mathsf{fvs}(e) \subseteq \mathsf{dom}(\sigma)$

where $\mathsf{fvs}(e)$ is the set of **free variables** in e defined as follows:

$$
\begin{array}{lcl}
\mathsf{fvs}(n) & \triangleq & \varnothing \\
\mathsf{fvs}(e_1 \texttt{ + } e_2) & \triangleq & \mathsf{fvs}(e_1) \cup \mathsf{fvs}(e_2) \\
\mathsf{fvs}(e_1 \texttt{ * } e_2) & \triangleq & \mathsf{fvs}(e_1) \cup \mathsf{fvs}(e_2) \\
\mathsf{fvs}(x \texttt{ := } e_1\texttt{; } e_2) & \triangleq & \mathsf{fvs}(e_1) \cup (\mathsf{fvs}(e_2) \setminus \{x\}) \\
\mathsf{fvs}(x) & \triangleq & \{x\}
\end{array}
$$

Soundness

With the well-formed configurations, we can restore the soundness of the small-step operational semantics of VAE:

Theorem (Soundness)

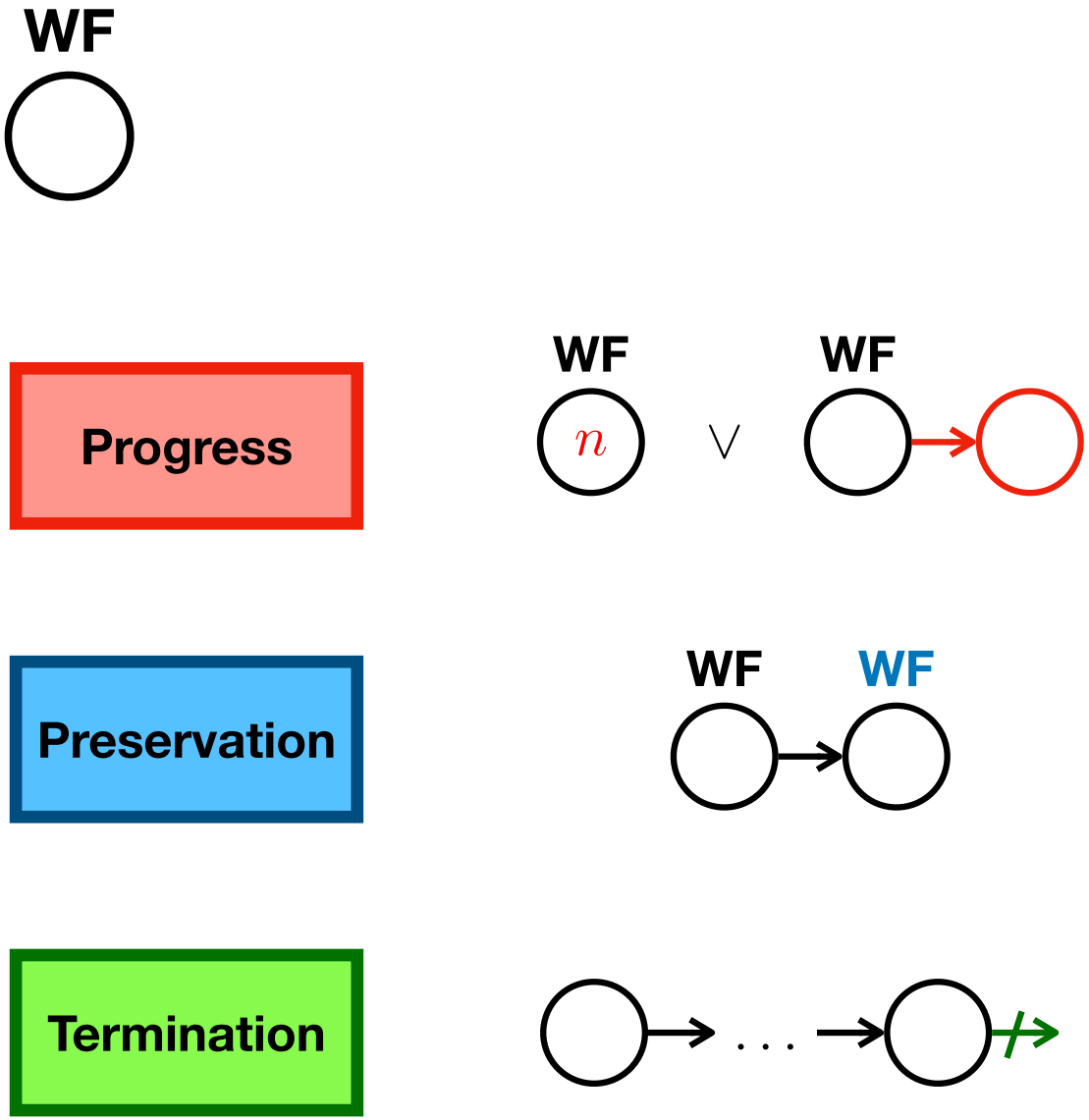
Every well-formed configuration evaluates to an integer:

$$\begin{aligned}&\forall \sigma \in \Sigma.\ \forall e \in \mathbb{E}.\\ &\quad \mathit{fvs}(e) \subseteq \mathit{dom}(\sigma) \implies\\ &\quad \exists \sigma' \in \Sigma.\ \exists n' \in \mathbb{Z}.\ \langle \sigma, e \rangle \to^* \langle \sigma', n' \rangle\end{aligned}$$

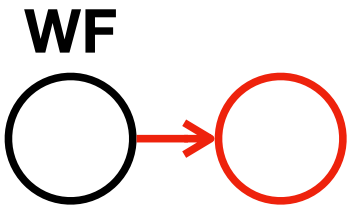
We will prove with the following three properties:

- **Progress**: Every well-formed configuration either is an integer or can take a step.
- **Preservation**: Every step preserves well-formedness of configurations.
- **Termination**: Every configuration terminates in a finite steps.

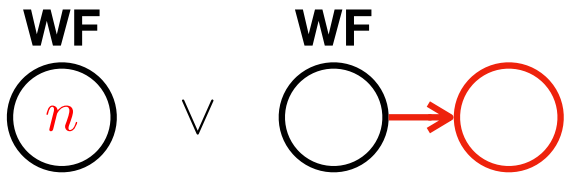
Soundness



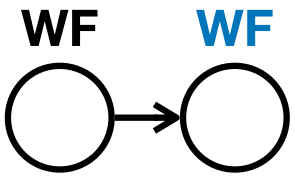
Soundness



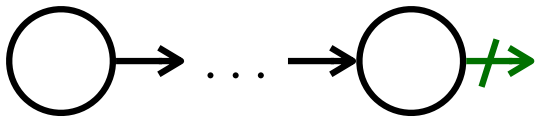
Progress



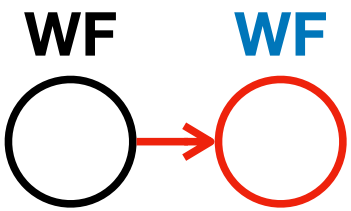
Preservation



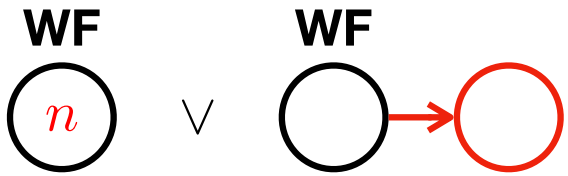
Termination



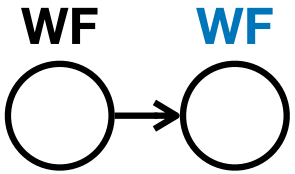
Soundness



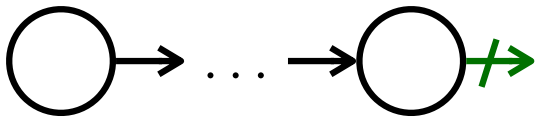
Progress



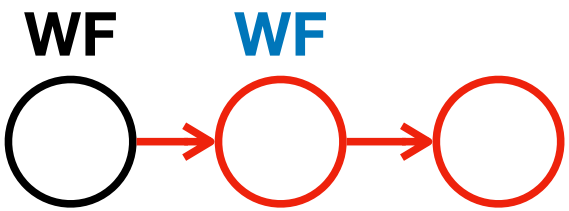
Preservation



Termination

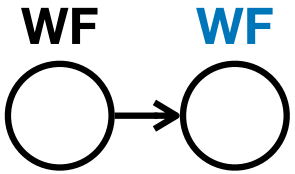


Soundness

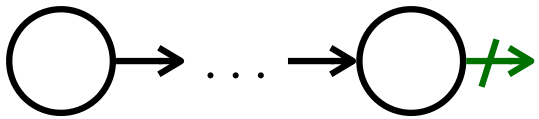


Progress

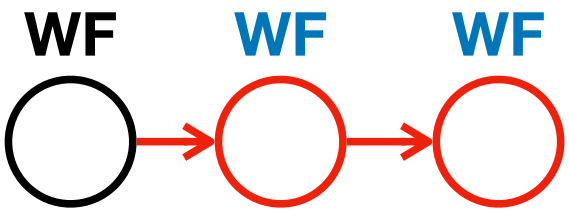
Preservation



Termination

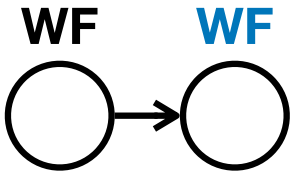


Soundness

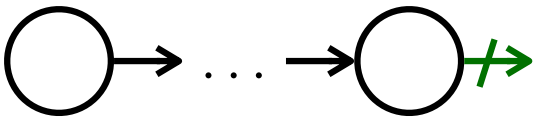


Progress

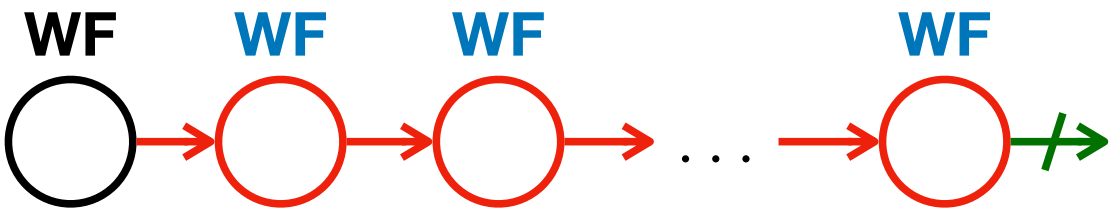
Preservation



Termination

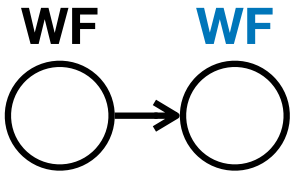


Soundness

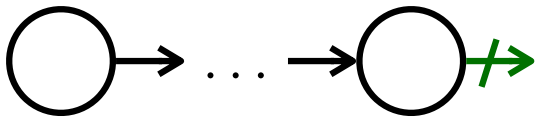


Progress

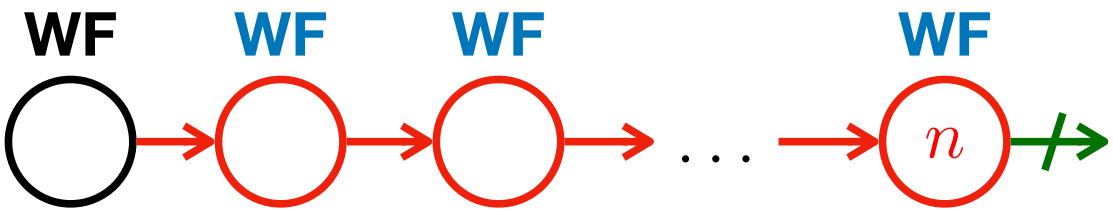
Preservation



Termination

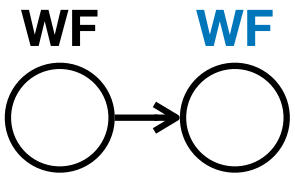


Soundness

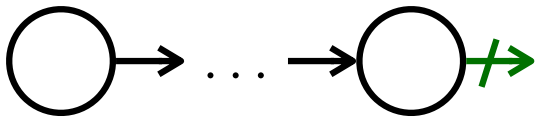


Progress

Preservation



Termination



Progress

Theorem (Progress)

Every well-formed configuration either is an integer or can take a step:

$$\begin{aligned}&\forall \sigma \in \Sigma.\ \forall e \in \mathbb{E}.\\ &\quad \mathit{fvs}(e) \subseteq \mathit{dom}(\sigma) \implies\\ &\quad e \in \mathbb{Z} \vee (\exists \sigma' \in \Sigma.\ \exists e' \in \mathbb{E}.\ \langle \sigma, e \rangle \to \langle \sigma', e' \rangle)\end{aligned}$$

Let's prove it by structural induction on expressions.

- $\boxed{e = n}$: e is an integer, so the progress property holds.

Progress

PLRG

Theorem (Progress)

Every well-formed configuration either is an integer or can take a step:

$$\begin{aligned}&\forall \sigma \in \Sigma.\ \forall e \in \mathbb{E}.\\ &\quad \mathit{fvs}(e) \subseteq \mathit{dom}(\sigma) \implies\\ &\quad e \in \mathbb{Z} \vee (\exists \sigma' \in \Sigma.\ \exists e' \in \mathbb{E}.\ \langle \sigma, e \rangle \to \langle \sigma', e' \rangle)\end{aligned}$$

Let's prove it by structural induction on expressions.

- $\boxed{e = n}$: e is an integer, so the progress property holds.
- $\boxed{e = x}$: Since $\mathsf{fvs}(e) = \{x\} \subseteq \mathsf{dom}(\sigma)$, we have $x \in \mathsf{dom}(\sigma)$. Thus, $\langle \sigma, x \rangle \to \langle \sigma, \sigma(x) \rangle$.

$$\frac{x \in \mathsf{dom}(\sigma)}{\langle \sigma, x \rangle \to \langle \sigma, \sigma(x) \rangle}$$

Progress

- $\boxed{e = e_1 + e_2}$: We know that $\langle \sigma, e_1 \rangle$ and $\langle \sigma, e_2 \rangle$ are well-formed:

$$\mathsf{fvs}(e_1) \subseteq \mathsf{fvs}(e_1 + e_2) \subseteq \mathsf{dom}(\sigma)$$
$$\mathsf{fvs}(e_2) \subseteq \mathsf{fvs}(e_1 + e_2) \subseteq \mathsf{dom}(\sigma)$$

because $\mathsf{fvs}(e_1 + e_2) \triangleq \mathsf{fvs}(e_1) \cup \mathsf{fvs}(e_2)$.

Progress

- $\boxed{e = e_1 \text{ + } e_2}$: We know that $\langle \sigma, e_1 \rangle$ and $\langle \sigma, e_2 \rangle$ are well-formed:

$$\mathsf{fvs}(e_1) \subseteq \mathsf{fvs}(e_1 \text{ + } e_2) \subseteq \mathsf{dom}(\sigma)$$
$$\mathsf{fvs}(e_2) \subseteq \mathsf{fvs}(e_1 \text{ + } e_2) \subseteq \mathsf{dom}(\sigma)$$

  because $\mathsf{fvs}(e_1 \text{ + } e_2) \triangleq \mathsf{fvs}(e_1) \cup \mathsf{fvs}(e_2)$.

  - $\boxed{e_1 \notin \mathbb{Z}}$: By the I.H., we have $\langle \sigma, e_1 \rangle \to \langle \sigma', e_1' \rangle$ for some $\sigma' \in \Sigma$ and $e_1' \in \mathbb{E}$. Thus, $\langle \sigma, e_1 \text{ + } e_2 \rangle \to \langle \sigma', e_1' \text{ + } e_2 \rangle$.

$$\frac{\langle \sigma, e_1 \rangle \to \langle \sigma', e_1' \rangle}{\langle \sigma, e_1 \text{ + } e_2 \rangle \to \langle \sigma', e_1' \text{ + } e_2 \rangle}$$

Progress

- $\boxed{e = e_1 \text{ + } e_2}$: We know that $\langle \sigma, e_1 \rangle$ and $\langle \sigma, e_2 \rangle$ are well-formed:

$$\mathsf{fvs}(e_1) \subseteq \mathsf{fvs}(e_1 \text{ + } e_2) \subseteq \mathsf{dom}(\sigma)$$
$$\mathsf{fvs}(e_2) \subseteq \mathsf{fvs}(e_1 \text{ + } e_2) \subseteq \mathsf{dom}(\sigma)$$

  because $\mathsf{fvs}(e_1 \text{ + } e_2) \triangleq \mathsf{fvs}(e_1) \cup \mathsf{fvs}(e_2)$.

  - $\boxed{e_1 \notin \mathbb{Z}}$: By the I.H., we have $\langle \sigma, e_1 \rangle \to \langle \sigma', e_1' \rangle$ for some $\sigma' \in \Sigma$ and $e_1' \in \mathbb{E}$. Thus, $\langle \sigma, e_1 \text{ + } e_2 \rangle \to \langle \sigma', e_1' \text{ + } e_2 \rangle$.

$$\frac{\langle \sigma, e_1 \rangle \to \langle \sigma', e_1' \rangle}{\langle \sigma, e_1 \text{ + } e_2 \rangle \to \langle \sigma', e_1' \text{ + } e_2 \rangle}$$

  - $\boxed{e_1 \in \mathbb{Z} \wedge e_2 \notin \mathbb{Z}}$: Similarly, we can use I.H. for $\langle \sigma, e_2 \rangle$.

$$\frac{\langle \sigma, e_2 \rangle \to \langle \sigma', e_2' \rangle}{\langle \sigma, n_1 \text{ + } e_2 \rangle \to \langle \sigma', n_1 \text{ + } e_2' \rangle}$$

Progress

- $\boxed{e = e_1 + e_2}$: We know that $\langle \sigma, e_1 \rangle$ and $\langle \sigma, e_2 \rangle$ are well-formed:

$$\mathsf{fvs}(e_1) \subseteq \mathsf{fvs}(e_1 + e_2) \subseteq \mathsf{dom}(\sigma)$$
$$\mathsf{fvs}(e_2) \subseteq \mathsf{fvs}(e_1 + e_2) \subseteq \mathsf{dom}(\sigma)$$

  because $\mathsf{fvs}(e_1 + e_2) \triangleq \mathsf{fvs}(e_1) \cup \mathsf{fvs}(e_2)$.

  - $\boxed{e_1 \notin \mathbb{Z}}$: By the I.H., we have $\langle \sigma, e_1 \rangle \to \langle \sigma', e_1' \rangle$ for some $\sigma' \in \Sigma$ and $e_1' \in \mathbb{E}$. Thus, $\langle \sigma, e_1 + e_2 \rangle \to \langle \sigma', e_1' + e_2 \rangle$.

$$\frac{\langle \sigma, e_1 \rangle \to \langle \sigma', e_1' \rangle}{\langle \sigma, e_1 + e_2 \rangle \to \langle \sigma', e_1' + e_2 \rangle}$$

  - $\boxed{e_1 \in \mathbb{Z} \wedge e_2 \notin \mathbb{Z}}$: Similarly, we can use I.H. for $\langle \sigma, e_2 \rangle$.

$$\frac{\langle \sigma, e_2 \rangle \to \langle \sigma', e_2' \rangle}{\langle \sigma, n_1 + e_2 \rangle \to \langle \sigma', n_1 + e_2' \rangle}$$

  - $\boxed{e_1 \in \mathbb{Z} \wedge e_2 \in \mathbb{Z}}$: Since there is no premise, it can always take a step.

$$\frac{}{\langle \sigma, n_1 + n_2 \rangle \to \langle \sigma, n_1 + n_2 \rangle}$$

Progress

- $\boxed{e = e_1 * e_2}$: Similar to the case of addition.

Progress

- $\boxed{e = e_1 \ * \ e_2}$: Similar to the case of addition.
- $\boxed{e = (x \ := e_1;\ e_2)}$: We know that $\langle \sigma, e_1 \rangle$ is well-formed:

$$\mathsf{fvs}(e_1) \subseteq \mathsf{fvs}(x \ := e_1;\ e_2) \subseteq \mathsf{dom}(\sigma)$$

because $\mathsf{fvs}(x \ := e_1;\ e_2) \triangleq \mathsf{fvs}(e_1) \cup (\mathsf{fvs}(e_2) \setminus \{x\})$.

Progress

- $\boxed{e = e_1 \ * \ e_2}$: Similar to the case of addition.
- $\boxed{e = (x \ := e_1;\ e_2)}$: We know that $\langle \sigma, e_1 \rangle$ is well-formed:

$$\mathsf{fvs}(e_1) \subseteq \mathsf{fvs}(x \ := e_1;\ e_2) \subseteq \mathsf{dom}(\sigma)$$

because $\mathsf{fvs}(x \ := e_1;\ e_2) \triangleq \mathsf{fvs}(e_1) \cup (\mathsf{fvs}(e_2) \setminus \{x\})$.

  - $\boxed{e_1 \notin \mathbb{Z}}$: By the I.H., we have $\langle \sigma, e_1 \rangle \to \langle \sigma', e_1' \rangle$ for some $\sigma' \in \Sigma$ and $e_1' \in \mathbb{E}$. Thus, $\langle \sigma, x \ := e_1;\ e_2 \rangle \to \langle \sigma', x \ := e_1';\ e_2 \rangle$.

$$\frac{\langle \sigma, e_1 \rangle \to \langle \sigma', e_1' \rangle}{\langle \sigma, x \ := e_1;\ e_2 \rangle \to \langle \sigma', x \ := e_1';\ e_2 \rangle}$$

Progress

- $\boxed{e = e_1 \ * \ e_2}$: Similar to the case of addition.
- $\boxed{e = (x \ := e_1;\ e_2)}$: We know that $\langle \sigma, e_1 \rangle$ is well-formed:

$$\mathsf{fvs}(e_1) \subseteq \mathsf{fvs}(x \ := e_1;\ e_2) \subseteq \mathsf{dom}(\sigma)$$

because $\mathsf{fvs}(x \ := e_1;\ e_2) \triangleq \mathsf{fvs}(e_1) \cup (\mathsf{fvs}(e_2) \setminus \{x\})$.

  - $\boxed{e_1 \notin \mathbb{Z}}$: By the I.H., we have $\langle \sigma, e_1 \rangle \to \langle \sigma', e_1' \rangle$ for some $\sigma' \in \Sigma$ and $e_1' \in \mathbb{E}$. Thus, $\langle \sigma, x \ := e_1;\ e_2 \rangle \to \langle \sigma', x \ := e_1';\ e_2 \rangle$.

$$\frac{\langle \sigma, e_1 \rangle \to \langle \sigma', e_1' \rangle}{\langle \sigma, x \ := e_1;\ e_2 \rangle \to \langle \sigma', x \ := e_1';\ e_2 \rangle}$$

  - $\boxed{e_1 \in \mathbb{Z}}$: Since there is no premise, it can always take a step.

$$\frac{}{\langle \sigma, x \ := n;\ e_2 \rangle \to \langle \sigma[x \mapsto n], e_2 \rangle}$$

Preservation

Theorem (Preservation)

Every step preserves well-formedness of configurations:

$$\forall \sigma, \sigma' \in \Sigma.\ \forall e, e' \in \mathbb{E}.$$
$$\mathit{fvs}(e) \subseteq \mathit{dom}(\sigma) \land \langle \sigma, e \rangle \to \langle \sigma', e' \rangle \implies \mathit{fvs}(e') \subseteq \mathit{dom}(\sigma')$$

Preservation

PLRG

Theorem (Preservation)

Every step preserves well-formedness of configurations:

$$\forall \sigma, \sigma' \in \Sigma.\ \forall e, e' \in \mathbb{E}.$$
$$\mathit{fvs}(e) \subseteq \mathit{dom}(\sigma) \land \langle \sigma, e \rangle \to \langle \sigma', e' \rangle \implies \mathit{fvs}(e') \subseteq \mathit{dom}(\sigma')$$

Let's prove it by induction on the derivation of inference rules.

- $\boxed{e = (x := n;\ e_2)}$: See the following inference rule:

$$\frac{}{\langle \sigma, x := n;\ e_2 \rangle \to \langle \sigma[x \mapsto n], e_2 \rangle}$$

We can show that $\mathsf{fvs}(e_2) \subseteq \mathsf{dom}(\sigma[x \mapsto n])$ as follows:

$$\begin{aligned} \mathsf{fvs}(e_2) &\subseteq \mathsf{fvs}(e_2) \setminus \{x\} \cup \{x\} \\ &= \mathsf{fvs}(x := n;\ e_2) \cup \{x\} \\ &\subseteq \mathsf{dom}(\sigma) \cup \{x\} \\ &= \mathsf{dom}(\sigma[x \mapsto n]) \end{aligned}$$

Preservation

- $\boxed{e = e_1 + e_2}$: We know that $\langle \sigma, e_1 \rangle$ and $\langle \sigma, e_2 \rangle$ are well-formed:

$$\mathsf{fvs}(e_1) \subseteq \mathsf{fvs}(e_1 + e_2) \subseteq \mathsf{dom}(\sigma)$$
$$\mathsf{fvs}(e_2) \subseteq \mathsf{fvs}(e_1 + e_2) \subseteq \mathsf{dom}(\sigma)$$

because $\mathsf{fvs}(e_1 + e_2) \triangleq \mathsf{fvs}(e_1) \cup \mathsf{fvs}(e_2)$.

$$\frac{\langle \sigma, e_1 \rangle \to \langle \sigma', e_1' \rangle}{\langle \sigma, e_1 + e_2 \rangle \to \langle \sigma', e_1' + e_2 \rangle}$$

Thus, I.H. tells us that $\mathsf{fvs}(e_1') \subseteq \mathsf{dom}(\sigma')$.

We know $\mathsf{dom}(\sigma) \subseteq \mathsf{dom}(\sigma')$ because all small-step evaluation rules only add new bindings to environments.

Thus, $\langle \sigma', e_1' + e_2 \rangle$ is well-formed because:

$$\mathsf{fvs}(e_1' + e_2) = \mathsf{fvs}(e_1') \cup \mathsf{fvs}(e_2) \subseteq \mathsf{dom}(\sigma') \cup \mathsf{dom}(\sigma) \subseteq \mathsf{dom}(\sigma')$$

- We can prove the rest of the cases similarly.

Termination

PLRG

Theorem (Termination)

Every configuration terminates in a finite steps:

$$\forall \sigma, \sigma' \in \Sigma.\ \forall e, e' \in \mathbb{E}.\ \exists k \in \mathbb{N}.\ \langle \sigma, e \rangle \rightarrow^{k} \langle \sigma', e' \rangle \not\rightarrow$$

Termination

PLRG

Theorem (Termination)

Every configuration terminates in a finite steps:

$$\forall \sigma, \sigma' \in \Sigma.\ \forall e, e' \in \mathbb{E}.\ \exists k \in \mathbb{N}.\ \langle \sigma, e \rangle \to^k \langle \sigma', e' \rangle \not\to$$

We can prove it by showing that each small-step evaluation strictly decreases the **size** of expressions:

$$\begin{array}{lcl}
\mathsf{size}(n) & \triangleq & 0 \\
\mathsf{size}(e_1 \texttt{ + } e_2) & \triangleq & 1 + \mathsf{size}(e_1) + \mathsf{size}(e_2) \\
\mathsf{size}(e_1 \texttt{ * } e_2) & \triangleq & 1 + \mathsf{size}(e_1) + \mathsf{size}(e_2) \\
\mathsf{size}(x \texttt{ := } e_1\texttt{; } e_2) & \triangleq & 1 + \mathsf{size}(e_1) + \mathsf{size}(e_2) \\
\mathsf{size}(x) & \triangleq & 1
\end{array}$$

Based on the induction on the derivation of inference rules, we can show that $\mathsf{size}(e) > \mathsf{size}(e')$ whenever $\langle \sigma, e \rangle \to \langle \sigma', e' \rangle$.

Soundness vs. Completeness

- **Soundness**: Every well-formed configuration evaluates to an integer.
- **Completeness**: Every configuration that evaluates to an integer is well-formed.

Soundness vs. Completeness

- **Soundness**: Every well-formed configuration evaluates to an integer.
- **Completeness**: Every configuration that evaluates to an integer is well-formed.

We have proved the soundness of the small-step operational semantics of VAE, but we have not proved its completeness.

Is it complete?

Soundness vs. Completeness

- **Soundness**: Every well-formed configuration evaluates to an integer.
- **Completeness**: Every configuration that evaluates to an integer is well-formed.

We have proved the soundness of the small-step operational semantics of VAE, but we have not proved its completeness.

Is it complete? **No!** Consider the following configuration evaluating to 2:

$$\langle \varnothing, \{\mathtt{x} := 1;\ \mathtt{x}\} + \mathtt{x} \rangle \to \langle [\mathtt{x} \mapsto 1], \mathtt{x} + \mathtt{x} \rangle \to^* \langle [\mathtt{x} \mapsto 1], 2 \rangle$$

But, this configuration is not well-formed:

$$\mathsf{fvs}((\mathtt{x} := 1;\ \mathtt{x}) + \mathtt{x}) = \{\mathtt{x}\} \not\subseteq \mathsf{dom}(\varnothing) = \varnothing$$

It happens because $\mathsf{fvs}(-)$ follows lexical scope, while the small-step operational semantics of VAE with environments does not.

Contents

Simple Imperative Language – IMP

PLRG

$$
\begin{array}{lll}
\text{Expressions} & e & ::= n \mid x \mid e \texttt{ + } e \mid e \texttt{ * } e \mid e \texttt{ < } e \mid \texttt{true} \mid \texttt{false} \\
\text{Statements} & s & ::= \texttt{skip} \\
 & & \mid x \texttt{ := } e \\
 & & \mid s \texttt{;} s \\
 & & \mid \texttt{if } e \texttt{ then } s \texttt{ else } s \\
 & & \mid \texttt{while } e \texttt{ do } s \\
\text{Values} & v & ::= n \mid \texttt{true} \mid \texttt{false}
\end{array}
$$

where $n \in \mathbb{Z}$, and $x \in \mathbb{X}$.

We will define two forms of small-step operational semantics for IMP:

$$\boxed{\langle \sigma, e \rangle \to \langle \sigma, v \rangle} \qquad \boxed{\langle \sigma, s \rangle \to \langle \sigma, s \rangle}$$

where $\sigma : \mathbb{X} \rightharpoonup \mathbb{V}$.

Expressions

$$\frac{x \in \mathsf{dom}(\sigma)}{\langle \sigma, x \rangle \to \langle \sigma, \sigma(x) \rangle}$$

$$\frac{\langle \sigma, e_1 \rangle \to \langle \sigma, e_1' \rangle}{\langle \sigma, e_1 \texttt{ + } e_2 \rangle \to \langle \sigma, e_1' \texttt{ + } e_2 \rangle} \qquad \frac{\langle \sigma, e_2 \rangle \to \langle \sigma, e_2' \rangle}{\langle \sigma, n_1 \texttt{ + } e_2 \rangle \to \langle \sigma, n_1 \texttt{ + } e_2' \rangle} \qquad \frac{}{\langle \sigma, n_1 \texttt{ + } n_2 \rangle \to \langle \sigma, n_1 + n_2 \rangle}$$

$$\frac{\langle \sigma, e_1 \rangle \to \langle \sigma, e_1' \rangle}{\langle \sigma, e_1 \texttt{ * } e_2 \rangle \to \langle \sigma, e_1' \texttt{ * } e_2 \rangle} \qquad \frac{\langle \sigma, e_2 \rangle \to \langle \sigma, e_2' \rangle}{\langle \sigma, n_1 \texttt{ * } e_2 \rangle \to \langle \sigma, n_1 \texttt{ * } e_2' \rangle} \qquad \frac{}{\langle \sigma, n_1 \texttt{ * } n_2 \rangle \to \langle \sigma, n_1 \times n_2 \rangle}$$

$$\frac{\langle \sigma, e_1 \rangle \to \langle \sigma, e_1' \rangle}{\langle \sigma, e_1 \texttt{ < } e_2 \rangle \to \langle \sigma, e_1' \texttt{ < } e_2 \rangle} \qquad \frac{\langle \sigma, e_2 \rangle \to \langle \sigma, e_2' \rangle}{\langle \sigma, n_1 \texttt{ < } e_2 \rangle \to \langle \sigma, n_1 \texttt{ < } e_2' \rangle} \qquad \frac{}{\langle \sigma, n_1 \texttt{ < } n_2 \rangle \to \langle \sigma, n_1 < n_2 \rangle}$$

Statements

$$\frac{\langle \sigma, e \rangle \to \langle \sigma, e' \rangle}{\langle \sigma, x \texttt{ := } e \rangle \to \langle \sigma, x \texttt{ := } e' \rangle} \qquad \frac{}{\langle \sigma, x \texttt{ := } v \rangle \to \langle \sigma[x \mapsto v], \texttt{skip} \rangle}$$

$$\frac{\langle \sigma, s_1 \rangle \to \langle \sigma, s_1' \rangle}{\langle \sigma, s_1 \texttt{ ; } s_2 \rangle \to \langle \sigma, s_1' \texttt{ ; } s_2 \rangle} \qquad \frac{}{\langle \sigma, \texttt{skip; } s_2 \rangle \to \langle \sigma, s_2 \rangle}$$

$$\frac{\langle \sigma, e \rangle \to \langle \sigma, e' \rangle}{\langle \sigma, \texttt{if } e \texttt{ then } s_1 \texttt{ else } s_2 \rangle \to \langle \sigma, \texttt{if } e' \texttt{ then } s_1 \texttt{ else } s_2 \rangle}$$

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

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

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

Next Lecture

- Evaluation Context and Lambda Calculus

Jihyeok Park
jihyeok_park@korea.ac.kr
https://plrg.korea.ac.kr