Big-step Semantics

Renato Neves

Universidade do Minho

HASLab
HIGH-ASSURANCE
SOFTWARE LABORATORY

Table of Contents

Semantics for every season

Operational semantics	How a program operates
Denotational semantics	What a program is
Axiomatic semantics	Which logical properties a program satisfies

Motivation

The semantics $\longrightarrow^\star$ provides a notion of program equivalence

$$\mathrm{p} \equiv \mathrm{q} \text{ iff } \Big(\langle \mathrm{p}, \sigma \rangle \longrightarrow^\star v \text{ iff } \langle \mathrm{q}, \sigma \rangle \longrightarrow^\star v\Big)$$

Motivation

The semantics $\longrightarrow^\star$ provides a notion of program equivalence

$$\mathrm{p} \equiv \mathrm{q} \text{ iff } \Big(\langle \mathrm{p}, \sigma \rangle \longrightarrow^\star v \text{ iff } \langle \mathrm{q}, \sigma \rangle \longrightarrow^\star v\Big)$$

This leads us to a previous slide . . .

The search for meaning

Examples

- $\mathtt{p};(\mathtt{q};\mathtt{r}) \stackrel{?}{=} (\mathtt{p};\mathtt{q});\mathtt{r}$
- $\mathtt{p} \parallel \mathtt{q} \stackrel{?}{=} \mathtt{q} \parallel \mathtt{p}$
- $\left(\mathtt{p} +_{\frac{1}{2}} \mathtt{q}\right);\mathtt{r} \stackrel{?}{=} \mathtt{p};\mathtt{r} +_{\frac{1}{2}} \mathtt{q};\mathtt{r}$
- $\mathtt{entangle}(\mathtt{x},\mathtt{y}) \stackrel{?}{=}$ spooky action

Motivation

However (dis)proving equivalences via $\longrightarrow^\star$ is quite cumbersome

Due to equivalence being concerned only with output ... not intermediate steps ...

... and outputs obtained via $\longrightarrow^\star$ relying on these

Motivation

However (dis)proving equivalences via $\longrightarrow^\star$ is quite cumbersome

Due to equivalence being concerned only with output . . . not intermediate steps . . .

. . . and outputs obtained via $\longrightarrow^\star$ relying on these

Can we build a more direct, big-step semantics ?

Table of Contents

A simple while-language

Arithmetic expressions

$\mathtt{e} ::= \mathtt{n} \mid \mathtt{e} \cdot \mathtt{e} \mid \mathtt{x} \mid \mathtt{e} + \mathtt{e}$

Programs

$\mathtt{p} ::= \mathtt{x} := \mathtt{e} \mid \mathtt{p} \,\mathtt{;}\, \mathtt{p} \mid \mathtt{if}\ \mathtt{b}\ \mathtt{then}\ \mathtt{p}\ \mathtt{else}\ \mathtt{p} \mid \mathtt{while}\ \mathtt{b}\ \mathtt{do}\ \{\ \mathtt{p}\ \}$

A simple while-language

Arithmetic expressions

$\mathtt{e} ::= \mathtt{n} \mid \mathtt{e} \cdot \mathtt{e} \mid \mathtt{x} \mid \mathtt{e} + \mathtt{e}$

Programs

$\mathtt{p} ::= \mathtt{x} := \mathtt{e} \mid \mathtt{p} \mathbin{;} \mathtt{p} \mid \mathtt{if}\ \mathtt{b}\ \mathtt{then}\ \mathtt{p}\ \mathtt{else}\ \mathtt{p} \mid \mathtt{while}\ \mathtt{b}\ \mathtt{do}\ \{\ \mathtt{p}\ \}$

Homework: provide semantics to the arithmetic expressions

A while-language and its semantics

$$\frac{\langle \mathtt{e}, \sigma \rangle \Downarrow \mathtt{v}}{\langle \mathtt{x} := \mathtt{e}, \sigma \rangle \Downarrow \sigma[\mathtt{v}/\mathtt{x}]} \text{ (asg)} \qquad \frac{\langle \mathtt{p}, \sigma \rangle \Downarrow \sigma' \quad \langle \mathtt{q}, \sigma' \rangle \Downarrow \sigma''}{\langle \mathtt{p} \,;\, \mathtt{q}, \sigma \rangle \Downarrow \sigma''} \text{ (seq)}$$

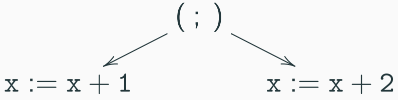
$$\frac{\langle \mathtt{b}, \sigma \rangle \Downarrow \mathtt{tt} \quad \langle \mathtt{p}, \sigma \rangle \Downarrow \sigma'}{\langle \mathtt{if}\ \mathtt{b}\ \mathtt{then}\ \mathtt{p}\ \mathtt{else}\ \mathtt{q}, \sigma \rangle \Downarrow \sigma'} \text{ (if}_1\text{)} \qquad \frac{\langle \mathtt{b}, \sigma \rangle \Downarrow \mathtt{ff} \quad \langle \mathtt{q}, \sigma \rangle \Downarrow \sigma'}{\langle \mathtt{if}\ \mathtt{b}\ \mathtt{then}\ \mathtt{p}\ \mathtt{else}\ \mathtt{q}, \sigma \rangle \Downarrow \sigma'} \text{ (if}_2\text{)}$$

$$\frac{\langle \mathtt{b}, \sigma \rangle \Downarrow \mathtt{tt} \quad \langle \mathtt{p}, \sigma \rangle \Downarrow \sigma' \quad \langle \mathtt{while}\ \mathtt{b}\ \mathtt{do}\ \{\ \mathtt{p}\ \}, \sigma' \rangle \Downarrow \sigma''}{\langle \mathtt{while}\ \mathtt{b}\ \mathtt{do}\ \{\ \mathtt{p}\ \}, \sigma \rangle \Downarrow \sigma''} \text{ (wh}_1\text{)}$$

$$\frac{\langle \mathtt{b}, \sigma \rangle \Downarrow \mathtt{ff}}{\langle \mathtt{while}\ \mathtt{b}\ \mathtt{do}\ \{\ \mathtt{p}\ \}, \sigma \rangle \Downarrow \sigma} \text{ (wh}_2\text{)}$$

The semantics at work

Program $\mathtt{x} := \mathtt{x} + 1 \,;\, \mathtt{x} := \mathtt{x} + 2$ corresponds to the syntax tree



Memory $\sigma = x \mapsto 3$ yields the derivation tree

$$\dfrac{\dfrac{\langle \mathtt{x} + 1, \mathtt{x} \mapsto 3 \rangle \Downarrow 4}{\langle \mathtt{x} := \mathtt{x} + 1, \mathtt{x} \mapsto 3 \rangle \Downarrow \mathtt{x} \mapsto 4} \qquad \dfrac{\langle \mathtt{x} + 2, \mathtt{x} \mapsto 4 \rangle \Downarrow 6}{\langle \mathtt{x} := \mathtt{x} + 2, \mathtt{x} \mapsto 4 \rangle \Downarrow \mathtt{x} \mapsto 6}}{\langle \mathtt{x} := \mathtt{x} + 1 \,;\, \mathtt{x} := \mathtt{x} + 2, \mathtt{x} \mapsto 3 \rangle \Downarrow \mathtt{x} \mapsto 6}$$

Exercise

Provide a big-step semantics to the propositional language

$$\mathtt{b} ::= \mathtt{x} \mid \mathtt{b} \wedge \mathtt{b} \mid \neg \mathtt{b}$$

Equivalence of while-programs

The previous semantics yields the following notion of equivalence
$\mathtt{p} \equiv \mathtt{q}$ if for all environments σ

$$\langle \mathtt{p}, \sigma \rangle \Downarrow \sigma' \text{ iff } \langle \mathtt{q}, \sigma \rangle \Downarrow \sigma'$$

Examples of equivalent terms

- $(\mathtt{p} \mathbin{;} \mathtt{q}) \mathbin{;} \mathtt{r} \equiv \mathtt{p} \mathbin{;} (\mathtt{q} \mathbin{;} \mathtt{r})$
- $(\mathtt{if}\ \mathtt{b}\ \mathtt{then}\ \mathtt{p}\ \mathtt{else}\ \mathtt{q}) \mathbin{;} \mathtt{r} \equiv \mathtt{if}\ \mathtt{b}\ \mathtt{then}\ \mathtt{p} \mathbin{;} \mathtt{r}\ \mathtt{else}\ \mathtt{q} \mathbin{;} \mathtt{r}$

The relation to the small-step semantics

Lemma

$$\langle \mathtt{p}, \sigma \rangle \longrightarrow \langle \mathtt{p}', \sigma' \rangle \Downarrow \sigma'' \textit{ implies } \langle \mathtt{p}, \sigma \rangle \Downarrow \sigma''$$

Proof.

Induction over the rules concerning $\longrightarrow$ □

The relation to the small-step semantics

Lemma

$$\langle \mathrm{p}, \sigma \rangle \longrightarrow \langle \mathrm{p}', \sigma' \rangle \Downarrow \sigma'' \text{ implies } \langle \mathrm{p}, \sigma \rangle \Downarrow \sigma''$$

Proof.

Induction over the rules concerning $\longrightarrow$ $\square$

Theorem

$$\langle \mathrm{p}, \sigma \rangle \longrightarrow^{\star} \sigma' \text{ iff } \langle \mathrm{p}, \sigma \rangle \Downarrow \sigma'$$

Proof.

Left-to-right-direction: previous lemma and induction over the rules concerning $\longrightarrow^{\star}$

Right-to-left direction: induction over the rules concerning $\Downarrow$ $\square$