Axiomatic Semantics

Renato Neves

Universidade do Minho

HASLab
HIGH-ASSURANCE
SOFTWARE LABORATORY

Semantics for Every Season

Operational semantics   How a program operates
Denotational semantics   What a program is
<u>Axiomatic semantics</u>   Which logical properties it satisfies

Table of Contents

A Brief Warm-up

Solve the following exercises via your favorite semantics

- Calculate the output of $x := 1 \, ; x := 2$

- Show that the following program outputs a state with $x \geq 2$

$$\text{if } x = 1 \text{ then } x := 2 \text{ else } x := 3$$

- Show that the following program is the factorial function

$$\text{while } x > 0 \, \{y := x \times y \, ; x := x - 1\}$$

A Brief Warm-up

Solve the following exercises via your favorite semantics

- Calculate the output of $x := 1 \, ; x := 2$

- Show that the following program outputs a state with $x \geq 2$
$$\text{if } x = 1 \text{ then } x := 2 \text{ else } x := 3$$

- Show that the following program is the factorial function
$$\text{while } x > 0 \, \{y := x \times y \, ; x := x - 1\}$$

Hard ?

The Right Tools

Two last exercises were about post-conditions . . .

not exactly about determining output . . .

nor about program equivalence

Two last exercises were about post-conditions ...

not exactly about determining output ...

nor about program equivalence

Do we have the right semantics for solving them ?

Table of Contents

Key Points

Focussed on output properties and less on outputs themselves

Centred around a logic (for reasoning about these properties)

Semantic rules are thus more logic oriented

Focussed on output properties and less on outputs themselves

Centred around a logic (for reasoning about these properties)

Semantic rules are thus more logic oriented

Good for program correctness (recall 'algorithms and complexity')

Hoare Triples

Axiomatic semantics essentially about (dis)proving

$$\{\Phi\} \, p \, \{\Psi\}$$

"If Φ holds at the input then Ψ holds at the output"

Examples

- $\{\mathtt{tt}\} \, p \, \{x \geq 2\}$
- $\{x = n \wedge y = 1\} \, p \, \{y = n!\}$
- $\dots$

Can we state mathematically what a Hoare triple really means ?

Can we state mathematically what a Hoare triple really means ?

Question rooted on what a program means (recall our lectures)

. . . and of course on the choice of a logic for properties

Meaning of Hoare Triples

Can we state <u>mathematically</u> what a Hoare triple really means ?

Question rooted on what a program means (recall our lectures)

. . . and of course on the <u>choice of a logic</u> for properties

Right choice often not obvious . . .

The Choice

Often varies depending on the problem at hand

. . . but typically the case that Φ corresponds to a subset

$$[\![\Phi]\!] \subseteq \text{State}_\perp$$

('the elements of $\text{State}_\perp$ at which Φ holds')

The Choice

Often varies depending on the problem at hand

... but typically the case that Φ corresponds to a subset

$$\llbracket \Phi \rrbracket \subseteq \text{State}_\perp$$

('the elements of $\text{State}_\perp$ at which Φ holds')

Scientists typically fix on the well-established first-order-logic

... which however brings its own set of problems

$$\{\Phi\}\, p\, \{\Psi\} \qquad \text{means} \qquad \Big(x \in [\![\Phi]\!] \implies [\![p]\!](x) \in [\![\Psi]\!] \Big)$$

$$\{\Phi\}\, \mathrm{p}\, \{\Psi\} \qquad \text{means} \qquad \Big(x \in [\![\Phi]\!] \implies [\![\mathrm{p}]\!](x) \in [\![\Psi]\!] \Big)$$

Remarkably note the following equivalence

$$\Big(x \in [\![\Phi]\!] \implies [\![\mathrm{p}]\!](x) \in [\![\Psi]\!] \Big) \quad \text{iff} \quad [\![\Phi]\!] \subseteq [\![\mathrm{p}]\!]^{-1}([\![\psi]\!])$$

It is at the root of a rich theory of

'backward transformations' known as predicate transformers

Liberals vs. Conservatives

In the sequel we will consider only <u>liberal conditions</u>

... *i.e.* every predicate Φ will have $\bot \in [\![\Phi]\!]$

Entails that we are working only with <u>partial correctness</u>

... *i.e.* no predicate enforces termination

Argue informally whether the triples below hold

- $\{\mathtt{tt}\}$ while tt skip $\{\mathtt{ff}\}$

- $\{\mathtt{tt}\}$ if b then x := 2 else x := 3 $\{x \geq 2\}$

- $\{x = a \wedge y = b\}$ x := y ; y := x $\{x = b \wedge y = a\}$

- $\{x = a \wedge y = b\}$ aux := x ; x := y ; y := aux $\{x = b \wedge y = a\}$

- $\{x = n \wedge y = 1\}$ fact $\{y = n!\}$

Table of Contents

Focus is on deriving the <u>weakest</u> condition Φ such that

$$\{\Phi\}\,p\,\{\Psi\} \qquad \left(\text{ iff } [\![\Phi]\!] \subseteq [\![p]\!]^{-1}([\![\Psi]\!]) \right)$$

Focus is on deriving the <u>weakest</u> condition Φ such that

$$\{\Phi\}\, p\, \{\Psi\} \qquad \left(\text{iff } [\![\Phi]\!] \subseteq [\![p]\!]^{-1}([\![\Psi]\!])\right)$$

Φ 'weaker' (*i.e.* less restrictive) than Φ' means $[\![\Phi]\!] \supseteq [\![\Phi']\!]$

Focus is on deriving the <u>weakest</u> condition Φ such that

$$\{\Phi\}\, p\, \{\Psi\} \qquad \Big(\text{ iff } [\![\Phi]\!] \subseteq [\![p]\!]^{-1}([\![\Psi]\!])\,\Big)$$

Φ 'weaker' (*i.e.* less restrictive) than Φ' means $[\![\Phi]\!] \supseteq [\![\Phi']\!]$

To <u>understand</u> a program amounts to knowing
the weakest precondition that ensures a given
postcondition

$$\mathrm{wp}\,(\mathtt{x} := \mathtt{e}, \Phi) = \Phi[\mathtt{e}/\mathtt{x}]$$

$$\mathrm{wp}\,(\mathtt{p}\,;\mathtt{q}, \Phi) = \mathrm{wp}\,(\mathtt{p}, \mathrm{wp}\,(\mathtt{q}, \Phi))$$

$$\mathrm{wp}\,(\mathtt{if}\ \mathtt{b}\ \mathtt{then}\ \mathtt{p}\ \mathtt{else}\ \mathtt{q}, \Phi) = \mathtt{b} \wedge \mathrm{wp}\,(\mathtt{p}, \Phi)\ \vee\ \neg\mathtt{b} \wedge \mathrm{wp}\,(\mathtt{q}, \Phi)$$

$$\mathrm{wp}\,(\mathtt{while}\ \mathtt{b}\ \mathtt{do}\ \{\,\mathtt{p}\,\}, \Phi) = \dots$$

Calculate the weakest preconditions w.r.t. the following pairs

- $(x := y,\ x \geq 1)$

- $(\text{if } b \text{ then } x := 2 \text{ else } x := 3,\ x \geq 2)$

- $(x := y\ ;\ y := x,\ x = b \wedge y = a)$

- $(\text{aux} := x;\ x := y\ ;\ y := \text{aux},\ x = b \wedge y = a)$

$$\mathrm{wp}\left(\mathtt{x} := \mathtt{e}, \Phi\right) = \Phi[\mathtt{e}/\mathtt{x}]$$

$$\mathrm{wp}\left(\mathtt{p} \,;\, \mathtt{q}, \Phi\right) = \mathrm{wp}\left(\mathtt{p}, \mathrm{wp}\left(\mathtt{q}, \Phi\right)\right)$$

$$\mathrm{wp}\left(\texttt{if}\ \mathtt{b}\ \texttt{then}\ \mathtt{p}\ \texttt{else}\ \mathtt{q}, \Phi\right) = \mathtt{b} \wedge \mathrm{wp}\left(\mathtt{p}, \Phi\right) \,\vee\, \neg\mathtt{b} \wedge \mathrm{wp}\left(\mathtt{q}, \Phi\right)$$

$$\mathrm{wp}\left(\texttt{while}\ \mathtt{b}\ \texttt{do}\ \{\,\mathtt{p}\,\}, \Phi\right) = \bigwedge_{n \in \mathbb{N}} \Psi_n$$

$$\mathrm{wp}\,(\mathtt{x} := \mathtt{e}, \Phi) = \Phi[\mathtt{e}/\mathtt{x}]$$

$$\mathrm{wp}\,(\mathtt{p}\,;\mathtt{q}, \Phi) = \mathrm{wp}\,(\mathtt{p}, \mathrm{wp}\,(\mathtt{q}, \Phi))$$

$$\mathrm{wp}\,(\mathtt{if}\ \mathtt{b}\ \mathtt{then}\ \mathtt{p}\ \mathtt{else}\ \mathtt{q}, \Phi) = \mathtt{b} \wedge \mathrm{wp}\,(\mathtt{p}, \Phi)\ \vee\ \neg\mathtt{b} \wedge \mathrm{wp}\,(\mathtt{q}, \Phi)$$

$$\mathrm{wp}\,(\mathtt{while}\ \mathtt{b}\ \mathtt{do}\ \{\,\mathtt{p}\,\}, \Phi) = \bigwedge_{n\in\mathbb{N}} \Psi_n$$

$$\Psi_0 = \mathtt{tt}$$

$$\Psi_{n+1} = \neg\mathtt{b} \wedge \Phi\ \vee\ \mathtt{b} \wedge \mathrm{wp}\,(\mathtt{p}, \Psi_n)$$

$\mathrm{wp}\,(\texttt{while}\,\mathrm{b}\,\texttt{do}\,\{\,\mathrm{p}\,\},\Phi)$

$=\Psi_0$ (trivial)*

$\wedge\,\neg\mathrm{b}\wedge\Phi\,\vee\,\mathrm{b}\wedge\mathrm{wp}\,(\mathrm{p},\Psi_0)$ (<u>terminates</u> with Φ <u>or iterates once</u> and then <u>$*$</u>)**

$\wedge\,\neg\mathrm{b}\wedge\Phi\,\vee\,\mathrm{b}\wedge\mathrm{wp}\,(\mathrm{p},\Psi_1)$ (<u>terminates</u> with Φ <u>or iterates once</u> and then <u>$**$</u>)

$\wedge\,\ldots$

$\mathrm{wp}\,(\texttt{while}\,\mathrm{b}\,\texttt{do}\,\{\,\mathrm{p}\,\},\Phi)$

$= \Psi_0$ (trivial)*

$\land\,\neg\mathrm{b}\land\Phi\,\lor\,\mathrm{b}\land\mathrm{wp}\,(\mathrm{p},\Psi_0)$ (<u>terminates</u> with Φ <u>or iterates once</u> and then <u>*</u>)**

$\land\,\neg\mathrm{b}\land\Phi\,\lor\,\mathrm{b}\land\mathrm{wp}\,(\mathrm{p},\Psi_1)$ (<u>terminates</u> with Φ <u>or iterates once</u> and then <u>**</u>)

$\land\,\ldots$

<u>Infinitary</u> formula tracks when the loop terminates

. . . in which case it enforces Φ

Each conjunct Ψ_{n+1} tracks up to n iterations

$$\mathrm{wp}\,(\texttt{while}\,\texttt{tt}\,\texttt{do}\,\{\,\texttt{p}\,\},\Phi)$$

$$= \Psi_0 \quad\quad (= \texttt{tt})$$

$$\wedge\,\neg\texttt{tt}\wedge\Phi\,\vee\,\texttt{tt}\wedge\mathrm{wp}\,(\texttt{p},\texttt{tt}) \quad\quad (= \texttt{tt})$$

$$\wedge\,\neg\texttt{tt}\wedge\Phi\,\vee\,\texttt{tt}\wedge\mathrm{wp}\,(\texttt{p},\texttt{tt}) \quad\quad (= \texttt{tt})$$

$$\wedge\,\ldots$$

$$= \texttt{tt}$$

Exercises

Prove that the following equations hold

- $\operatorname{wp}(\mathtt{p}, \mathtt{tt}) = \mathtt{tt}$

- $\operatorname{wp}(\mathtt{p}, \Phi \wedge \Psi) = \operatorname{wp}(\mathtt{p}, \Phi) \wedge \operatorname{wp}(\mathtt{p}, \Psi)$

- $\operatorname{wp}(\mathtt{p}, \bigwedge_{i \in I} \Phi_i) = \bigwedge_{i \in I} \operatorname{wp}(\mathtt{p}, \Phi_i)$

Pre-condition and Denotational Semantics

Theorem

$$[\![\mathrm{wp}\,(\mathrm{p}, \Phi)]\!] = [\![\mathrm{p}]\!]^{-1}([\![\Phi]\!])$$

Proof.

By <u>induction</u>. Case of while-loops proved neatly via domain
theory $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ □

Corollary

$$[\![\mathrm{p}]\!] = [\![\mathrm{q}]\!] \implies \forall \Phi.\, \mathrm{wp}\,(\mathrm{p}, \Phi) \equiv \mathrm{wp}\,(\mathrm{q}, \Phi)$$

Is it true that $\Big(\forall \Phi.\, \mathrm{wp}(p, \Phi) \equiv \mathrm{wp}\,(q, \Phi)\Big) \implies [\![p]\!] = [\![q]\!]$?

Expressivity Matters

Is it true that $\left(\forall \Phi.\, \mathrm{wp}(p, \Phi) \equiv \mathrm{wp}\,(q, \Phi) \right) \Longrightarrow [\![p]\!] = [\![q]\!]$?

Well . . .

$$\forall \Phi.\, \mathrm{wp}\,(p, \Phi) \equiv \mathrm{wp}\,(q, \Phi)$$
$$\Longrightarrow \forall \Phi.\, [\![\mathrm{wp}\,(p, \Phi)]\!] = [\![\mathrm{wp}\,(q, \Phi)]\!]$$
$$\Longrightarrow \forall \Phi.\, [\![p]\!]^{-1}([\![\Phi]\!]) = [\![q]\!]^{-1}([\![\Phi]\!])$$
$$\Longrightarrow [\![p]\!] = [\![q]\!]$$

Expressivity Matters

Is it true that $\left(\forall \Phi. \operatorname{wp}(p, \Phi) \equiv \operatorname{wp}(q, \Phi)\right) \Longrightarrow [\![p]\!] = [\![q]\!]$?

Well ...

$$\forall \Phi. \operatorname{wp}(p, \Phi) \equiv \operatorname{wp}(q, \Phi)$$
$$\Longrightarrow \forall \Phi. [\![\operatorname{wp}(p, \Phi)]\!] = [\![\operatorname{wp}(q, \Phi)]\!]$$
$$\Longrightarrow \forall \Phi. [\![p]\!]^{-1}([\![\Phi]\!]) = [\![q]\!]^{-1}([\![\Phi]\!])$$
$$\Longrightarrow [\![p]\!] = [\![q]\!]$$

A counter-example (the simplest grammar of propositions)

$$b ::= \mathtt{tt} \mid \neg b \mid b \vee b \mid \bigwedge b$$

Calculate all possible interpretations $[\![b]\!]$

From Weakest Pre-conditions to Hoare Triples

We wish to prove the validity of Hoare triples

. . . like in 'algorithms and complexity'

For this we use a simple calculus from the semantics

. . . with merely one rule

$$\frac{\vdash \Phi \rightarrow \mathrm{wp}\,(\mathrm{p}, \Psi)}{\vdash \{\Phi\}\,\mathrm{p}\,\{\Psi\}}$$

Is our calculus correct ?

$\dots$ *i.e.* $\vdash \{\Phi\} \, \mathrm{p} \, \{\Psi\} \implies [\![\Phi]\!] \subseteq [\![\mathrm{p}]\!]^{-1}([\![\Psi]\!])$

Soundness

Is our calculus correct ?

... *i.e.* $\vdash \{\Phi\} \, p \, \{\Psi\} \implies [\![\Phi]\!] \subseteq [\![p]\!]^{-1}([\![\Psi]\!])$

Yes, and the proof is super easy !!

Completeness

Is our calculus complete ?

... *i.e.* $\llbracket \Phi \rrbracket \subseteq \llbracket p \rrbracket^{-1}(\llbracket \Psi \rrbracket) \Longrightarrow \vdash \{\Phi\} \, p \, \{\Psi\}$

Is our calculus complete ?

... *i.e.* $\llbracket \Phi \rrbracket \subseteq \llbracket p \rrbracket^{-1}(\llbracket \Psi \rrbracket) \Longrightarrow \vdash \{\Phi\}\, p\, \{\Psi\}$

Well, it depends on whether the logic is complete

... *i.e.* $\llbracket \Phi_1 \rrbracket \subseteq \llbracket \Phi_2 \rrbracket \Longrightarrow \vdash \Phi_1 \rightarrow \Phi_2$ for all formulae Φ_1, Φ_2

Table of Contents

Motivation

Let us now try to establish

$$\vdash \{x = n \land y = 1\} \; \texttt{fact} \; \{y = n!\}$$

$$\left(\texttt{fact} = \texttt{while} \; x > 0 \; \{y := x \times y \, ; x := x - 1\}\right)$$

Motivation

Let us now try to establish

$$\vdash \{x = n \wedge y = 1\} \; \texttt{fact} \; \{y = n!\}$$

$$\Big(\texttt{fact} = \texttt{while}\, x > 0 \,\{y := x \times y \,;\, x := x - 1\}\Big)$$

Still hard ?

The calculus is for obtaining <u>weakest preconditions</u>

. . . and less about proving Hoare triples

. . . besides infinite conjunctions are hard !

$$\overline{\vdash \{\Phi\} \, \texttt{skip} \, \{\Phi\}} \qquad\qquad \overline{\vdash \{\Phi[e/x]\} \, \texttt{x} := \texttt{e} \, \{\Phi\}}$$

$$\frac{\vdash \{\Phi\} \, \texttt{p} \, \{\Psi\} \qquad \vdash \{\Psi\} \, \texttt{q} \, \{\Xi\}}{\vdash \{\Phi\} \, \texttt{p} \, ; \texttt{q} \, \{\Xi\}} \qquad\qquad \frac{\vdash \{\Phi \wedge \texttt{b}\} \, \texttt{p} \, \{\Phi\}}{\vdash \{\Phi\} \, \texttt{while} \, \texttt{b} \, \texttt{p} \, \{\neg \texttt{b} \wedge \Phi\}}$$

$$\frac{\vdash \{\texttt{b} \wedge \Phi\} \, \texttt{p} \, \{\Psi\} \qquad \vdash \{\neg \texttt{b} \wedge \Phi\} \, \texttt{q} \, \{\Psi\}}{\vdash \{\Phi\} \, \texttt{if} \, \texttt{b} \, \texttt{then} \, \texttt{p} \, \texttt{else} \, \texttt{q} \, \{\Psi\}}$$

$$\frac{\vdash \Phi \to \Psi \qquad \vdash \{\Psi\} \, \texttt{p} \, \{\Xi\} \qquad \vdash \Xi \to \Omega}{\vdash \{\Phi\} \, \texttt{p} \, \{\Psi\}}$$