

# Axiomatic Semantics

---

Renato Neves



Universidade do Minho



# 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

Motivation

Key Points

Weakest Precondition Semantics

Hoare Calculus

# 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  $x \geq 2$

`if  $x = 1$  then  $x := 2$  else  $x := 3$`

- Show that the following program is the factorial function

`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  $x \geq 2$

`if  $x = 1$  then  $x := 2$  else  $x := 3$`

- Show that the following program is the factorial function

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

Hard ?

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

Motivation

Key Points

Weakest Precondition Semantics

Hoare Calculus



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

# 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

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

Axiomatic semantics essentially about (dis)proving

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

"If  $\Phi$  holds at the input then  $\Psi$  holds at the output"

## Examples

- $\{\text{tt}\} p \{x \geq 2\}$
- $\{x = n \wedge y = 1\} p \{y = n!\}$
- ...

# Meaning of Hoare Triples

Can we state mathematically what a Hoare triple really means ?

# Meaning of Hoare Triples

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 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

Right choice often not obvious ...

Often varies depending on the problem at hand

... but typically the case that  $\Phi$  corresponds to a subset

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

(‘the elements of  $\text{State}_\perp$  at which  $\Phi$  holds’)

Often varies depending on the problem at hand

... but typically the case that  $\Phi$  corresponds to a subset

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

(‘the elements of  $\text{State}_\perp$  at which  $\Phi$  holds’)

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

... which however brings its own set of problems



# Meaning of Hoare Triples

$$\{\Phi\} p \{\Psi\} \quad \text{means} \quad \left( x \in \llbracket \Phi \rrbracket \implies \llbracket p \rrbracket(x) \in \llbracket \Psi \rrbracket \right)$$

# Meaning of Hoare Triples

$$\{\Phi\} p \{\Psi\} \quad \text{means} \quad \left( x \in \llbracket \Phi \rrbracket \implies \llbracket p \rrbracket(x) \in \llbracket \Psi \rrbracket \right)$$

Remarkably note the following equivalence

$$\left( x \in \llbracket \Phi \rrbracket \implies \llbracket p \rrbracket(x) \in \llbracket \Psi \rrbracket \right) \quad \text{iff} \quad \llbracket \Phi \rrbracket \subseteq \llbracket p \rrbracket^{-1}(\llbracket \Psi \rrbracket)$$

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 liberal conditions

... *i.e.* every predicate  $\Phi$  will have  $\perp \in \llbracket \Phi \rrbracket$

Entails that we are working only with partial correctness

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

Argue informally whether the triples below hold

- $\{tt\} \text{ while } tt \text{ skip } \{ff\}$
- $\{tt\} \text{ if } b \text{ then } x := 2 \text{ else } x := 3 \{x \geq 2\}$
- $\{x = a \wedge y = b\} x := y ; y := x \{x = b \wedge y = a\}$
- $\{x = a \wedge y = b\} x := aux ; x := y ; y := aux \{x = b \wedge y = a\}$
- $\{x = n \wedge y = 1\} \text{ fact } \{y = n!\}$

# Table of Contents

Motivation

Key Points

Weakest Precondition Semantics

Hoare Calculus

# What and Why

Focus is on deriving the weakest condition  $\Phi$  such that

$$\{\Phi\} p \{\Psi\} \quad \left( \text{iff } \llbracket \Phi \rrbracket \subseteq \llbracket p \rrbracket^{-1}(\llbracket \Psi \rrbracket) \right)$$

# What and Why

Focus is on deriving the weakest condition  $\Phi$  such that

$$\{\Phi\} p \{\Psi\} \quad \left( \text{iff } \llbracket \Phi \rrbracket \subseteq \llbracket p \rrbracket^{-1}(\llbracket \Psi \rrbracket) \right)$$

$\Phi$  'weaker' (*i.e.* less restrictive) than  $\Phi'$  means  $\llbracket \Phi \rrbracket \supseteq \llbracket \Phi' \rrbracket$

# What and Why

Focus is on deriving the weakest condition  $\Phi$  such that

$$\{\Phi\} p \{\Psi\} \quad \left( \text{iff } \llbracket \Phi \rrbracket \subseteq \llbracket p \rrbracket^{-1}(\llbracket \Psi \rrbracket) \right)$$

$\Phi$  'weaker' (*i.e.* less restrictive) than  $\Phi'$  means  $\llbracket \Phi \rrbracket \supseteq \llbracket \Phi' \rrbracket$

To understand a program amounts to knowing the weakest precondition that ensures a given postcondition





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

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

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

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

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

- $(x := y, x \geq 1)$
- $(\text{if } b \text{ then } x := 2 \text{ else } x := 3, x \geq 2)$
- $(x := y ; y := x, x = b \wedge y = a)$
- $(x := \text{aux} ; x := y ; y := \text{aux}, x = b \wedge y = a)$

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

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

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

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

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

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

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

$$\text{wp}(\text{while } b \text{ do } \{ p \}, \Phi) = \bigwedge_{n \in \mathbb{N}} \psi_n$$

$$\psi_0 = \text{tt}$$

$$\psi_{n+1} = \neg b \wedge \Phi \vee b \wedge \text{wp}(p, \psi_n)$$

# Unfolding While-loops

$$\begin{aligned} & \text{wp}(\text{while } b \text{ do } \{ p \}, \Phi) \\ &= \Psi_0 \quad (\text{trivial})^* \\ & \wedge \neg b \wedge \Phi \vee b \wedge \text{wp}(p, \Psi_0) \quad (\text{terminates with } \Phi \text{ or iterates once and then } \underline{*})^{**} \\ & \wedge \neg b \wedge \Phi \vee b \wedge \text{wp}(p, \Psi_1) \quad (\text{terminates with } \Phi \text{ or iterates once and then } \underline{**}) \\ & \wedge \dots \end{aligned}$$

# Unfolding While-loops

$$\begin{aligned} & \text{wp}(\text{while } b \text{ do } \{ p \}, \Phi) \\ &= \Psi_0 \quad (\text{trivial})^* \\ & \wedge \neg b \wedge \Phi \vee b \wedge \text{wp}(p, \Psi_0) \quad (\text{terminates with } \Phi \text{ or iterates once and then } *)^{**} \\ & \wedge \neg b \wedge \Phi \vee b \wedge \text{wp}(p, \Psi_1) \quad (\text{terminates with } \Phi \text{ or iterates once and then } **) \\ & \wedge \dots \end{aligned}$$

Infinitary formula tracks when the loop terminates

... in which case it enforces  $\Phi$

Each conjunct  $\Psi_{n+1}$  tracks up to  $n$  iterations

# Table of Contents

Motivation

Key Points

Weakest Precondition Semantics

Hoare Calculus