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
Axiomatic semantics	Which logical properties it satisfies

Table of Contents

A Brief Warm-up

Solve the following exercises via your favorite semantics

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

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

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

- Show that the following program is the factorial function

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

A Brief Warm-up

Solve the following exercises via your favorite semantics

- Calculate the output of $\mathtt{x} := 1\ ;\ \mathtt{x} := 2$
- Show that the following program outputs a state with $\mathtt{x} \geq 2$

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

- Show that the following program is the factorial function

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

Hard ?

The Right Tools

Two last exercises were about post-conditions . . .

not exactly about determining output . . .

nor about program equivalence

The Right Tools

Two last exercises were about post-conditions . . .

not exactly about determining output . . .

nor about program equivalence

$\boxed{\text{Do we have the right semantics for solving them ?}}$

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

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’)

Hoare Triples

Axiomatic semantics essentially about (dis)proving

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

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

Examples

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

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 ...

The Choice

Often varies depending on the problem at hand

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

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

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

The Choice

Often varies depending on the problem at hand

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

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

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

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

... which however brings its own set of problems

Meaning of Hoare Triples

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

Meaning of Hoare Triples

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

Remarkably note the following equivalence

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

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 liberal conditions

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

Entails that we are working only with partial correctness

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

Exercises

Argue informally whether the triples below hold

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

Table of Contents

What and Why

Focus is on deriving the weakest condition Φ such that

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

What and Why

Focus is on deriving the weakest condition Φ such that

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

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

What and Why

Focus is on deriving the weakest condition Φ such that

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

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

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

The Semantics

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

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

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

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

Exercises

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

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

The Semantics

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

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

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

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

The Semantics

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

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

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

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

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

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

Unfolding While-loops

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

$$= \Psi_0 \qquad \text{(trivial)}^{*}$$

$$\wedge\ \neg \mathrm{b} \wedge \Phi\ \vee\ \mathrm{b} \wedge \mathrm{wp}\left(\mathrm{p}, \Psi_0\right) \qquad \text{(terminates with } \Phi \text{ or iterates once and then } \underline{*}\text{)}^{**}$$

$$\wedge\ \neg \mathrm{b} \wedge \Phi\ \vee\ \mathrm{b} \wedge \mathrm{wp}\left(\mathrm{p}, \Psi_1\right) \qquad \text{(terminates with } \Phi \text{ or iterates once and then } \underline{**}\text{)}$$

$$\wedge\ \ldots$$

Unfolding While-loops

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

$= \Psi_0$ (trivial)*

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

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

$\wedge\ \ldots$

Infinitary formula tracks when the loop terminates

... in which case it enforces Φ

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

Unfolding While-loops (The Case of Divergence)

$$
\begin{aligned}
&\mathrm{wp}\,(\texttt{while}\,\texttt{tt}\,\texttt{do}\,\{\,\mathrm{p}\,\},\Phi) \\
&= \Psi_0 \qquad (= \texttt{tt}) \\
&\;\wedge \neg\texttt{tt} \wedge \Phi \;\vee\; \texttt{tt} \wedge \mathrm{wp}\,(\mathrm{p},\texttt{tt}) \qquad (= \texttt{tt}) \\
&\;\wedge \neg\texttt{tt} \wedge \Phi \;\vee\; \texttt{tt} \wedge \mathrm{wp}\,(\mathrm{p},\texttt{tt}) \qquad (= \texttt{tt}) \\
&\;\wedge \;\ldots \\
&\;= \texttt{tt}
\end{aligned}
$$

Exercises

Prove that the following equations hold

- $\mathrm{wp}\,(\mathtt{p}, \mathtt{tt}) = \mathtt{tt}$
- $\mathrm{wp}\,(\mathtt{p}, \Phi \wedge \Psi) = \mathrm{wp}\,(\mathtt{p}, \Phi) \wedge \mathrm{wp}\,(\mathtt{p}, \Psi)$
- $\mathrm{wp}\,(\mathtt{p}, \bigwedge_{i \in I} \Phi_i) = \bigwedge_{i \in I} \mathrm{wp}\,(\mathtt{p}, \Phi_i)$

Pre-condition and Denotational Semantics

Theorem

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

Proof.

By induction. Case of while-loops proved neatly via domain theory $\square$

Corollary

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

Expressivity Matters

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

Expressivity Matters

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

Well . . .

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

Expressivity Matters

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

Well . . .

$$
\begin{aligned}
& \forall \Phi.\, \mathrm{wp}\,(\mathrm{p}, \Phi) \equiv \mathrm{wp}\,(\mathrm{q}, \Phi) \\
& \implies \forall \Phi.\, [\![\mathrm{wp}\,(\mathrm{p}, \Phi)]\!] = [\![\mathrm{wp}\,(\mathrm{q}, \Phi)]\!] \\
& \implies \forall \Phi.\, [\![\mathrm{p}]\!]^{-1}([\![\Phi]\!]) = [\![\mathrm{q}]\!]^{-1}([\![\Phi]\!]) \\
& \implies [\![\mathrm{p}]\!] = [\![\mathrm{q}]\!]
\end{aligned}
$$

Counter-example (the simplest grammar of propositions)

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

Calculate all possible interpretations $[\![\mathtt{b}]\!]$

From Weakest Pre-conditions to Hoare Triples

We wish to prove the validity of Hoare triples

... just like in 'algorithms and complexity'

We use a calculus from the precondition semantics

... with merely one rule

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

The Quest for Soundness

Is our calculus correct ?

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

The Quest for Soundness

Is our calculus correct ?

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

Yes and moreover the proof is super easy !!

The Quest for Completeness

Is our calculus complete ?

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

The Quest for Completeness

Is our calculus complete ?

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

$\boxed{\text{It depends on whether the logic is complete} \ldots}$

$$\Big(\ldots \ \textit{i.e.}\ [\![\Phi_1]\!] \subseteq [\![\Phi_2]\!] \Longrightarrow \vdash \Phi_1 \to \Phi_2\Big)$$

Table of Contents

Motivation

Let us try to establish

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

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

Motivation

Let us try to establish

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

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

Hard ?

The calculus is strictly based on obtaining weakest preconditions

... which renders it hard to use in practice

Back to Old Friends

$$\frac{}{\vdash_H \{\Phi\}\,\mathtt{skip}\,\{\Phi\}} \qquad \frac{}{\vdash_H \{\Phi[e/x]\}\,x := e\,\{\Phi\}}$$

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

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

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

No Such Thing as a Free Lunch

Let us try now to establish

$$\vdash_H \{\mathtt{x} = \mathtt{n} \geq 0 \land \mathtt{y} = 1\} \ \mathtt{fact} \ \{\mathtt{y} = \mathtt{n}!\}$$

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

No Such Thing as a Free Lunch

Let us try now to establish

$$\vdash_H \{\mathtt{x} = \mathtt{n} \geq 0 \wedge \mathtt{y} = 1\} \ \mathtt{fact} \ \{\mathtt{y} = \mathtt{n}!\}$$

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

Much easier

... but only if we a find a suitable invariant

e.g. $\mathtt{y} \times \mathtt{x}! = \mathtt{n}! \wedge \mathtt{x} \geq 0$

No Such Thing as a Free Lunch

Let us try now to establish

$$\vdash_H \{\mathtt{x} = \mathtt{n} \geq 0 \wedge \mathtt{y} = 1\}\ \mathtt{fact}\ \{\mathtt{y} = \mathtt{n}!\}$$

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

Much easier

... but only if we a find a suitable invariant

e.g. $\mathtt{y} \times \mathtt{x}! = \mathtt{n}! \wedge \mathtt{x} \geq 0$

$\boxed{\text{Finding suitable invariants is now the hard part}}$

Hoare Calculus and its Correctness

Theorem (Soundness)

$\vdash_H \{\Phi\}\, \mathrm{p}\, \{\Psi\} \Longrightarrow [\![\Phi]\!] \subseteq [\![\mathrm{p}]\!]^{-1}([\![\Psi]\!])$

Proof.

By induction on $\vdash_H$. Case of while-loops proved neatly via domain theory □

Hoare Calculus and its Relative Completeness

Lemma

$\vdash_H \{\mathrm{wp}\,(\mathrm{p}, \Phi)\}\,\mathrm{p}\,\{\Phi\}$

Proof.

Induction on the structure of programs. $\square$

Corollary

$\vdash \{\Phi\}\,\mathrm{p}\,\{\Psi\} \Longrightarrow \vdash_H \{\Phi\}\,\mathrm{p}\,\{\Psi\}$

Hoare Calculus and its Relative Completeness

Theorem (Relative Completeness)

If underlying logic is complete

$$[\![\Phi]\!] \subseteq [\![\mathrm{p}]\!]^{-1}([\![\Psi]\!]) \implies \vdash_H \{\Phi\}\,\mathrm{p}\,\{\Psi\}$$

Proof.

Uses (relative) completeness of weakest preconditions and previous corollary $\square$

Corollary

$$[\![\mathrm{p}]\!] = [\![\mathrm{q}]\!] \implies \Big(\vdash_H \{\Phi\}\,\mathrm{p}\,\{\Psi\} \textit{ iff } \vdash_H \{\Phi\}\,\mathrm{q}\,\{\Psi\}\Big)$$

Table of Contents

Conclusions

Briefly studied axiomatic semantics

Suitable for program correctness

. . . more deeply, for studying programs from a logical perspective

Conclusions

Briefly studied axiomatic semantics

Suitable for program correctness

. . . more deeply, for studying programs from a logical perspective

Hints at profound connections between programming and logic

. . . one has much to learn from the other

Conclusions (Choosing a Logic)

Did not fix any logic

... although assumed strong constructs (infinite conjunctions)

Typical choice is first-order logic

... absence of infinite conjunctions circumvented by universal quantification

Conclusions (Choosing a Logic)

Computer states involve numbers

... and thus one typically uses FOL + natural numbers arithmetic

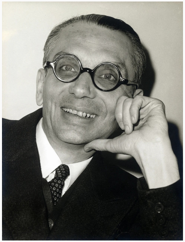
Conclusions (Choosing a Logic)

Computer states involve numbers

... and thus one typically uses FOL + natural numbers arithmetic

But Gödel's incompleteness theorem asserts that

Natural numbers arithmetic is incomplete



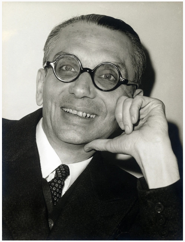
Conclusions (Choosing a Logic)

Computer states involve numbers

. . . and thus one typically uses FOL + natural numbers arithmetic

But Gödel's incompleteness theorem asserts that

Natural numbers arithmetic is incomplete



Which logic would you choose ?

Further Reading

Further details in [Rey98, Chapter 3] and [Win93, Chapter 6 and 7]

- John C Reynolds, *Theories of programming languages*, Cambridge University Press, 1998.
- Glynn Winskel, *The formal semantics of programming languages - an introduction*, Foundation of computing series, MIT Press, 1993.