Semantics of Programming Languages

Axiomatic Semantics

Prof. Peter Müller

Software Component Technology

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Motivation

- Developing an axiomatic semantics is difficult

- **Soundness**:
  If a property can be proved then it does indeed hold
  - An unsound inference system is useless

- **Completeness**:
  If a property does hold then it can be proved
  - With an incomplete inference system, a program might be correct, but we cannot prove it

Unsoundness: While Rule

- Why do we need the precondition $\exists Z : \mathbf{V}(Z)$?

$$\frac{\{\ \mathcal{B}[\![b]\!] \wedge \mathbf{P} \wedge \mathbf{V}(Z+1)\ \}\ s\ \{\ \Downarrow \mathbf{P} \wedge \mathbf{V}(Z)\ \}}{\{\ \mathbf{P}\ \}\ \texttt{while}\ b\ \texttt{do}\ s\ \texttt{end}\ \{\ \Downarrow \neg\mathcal{B}[\![b]\!] \wedge \mathbf{P}\ \}}$$

where $Z \in \mathbb{N}$

- With $\mathbf{V}(Z) \equiv \mathsf{x} = Z$, we can derive

$$\frac{\dfrac{\{\ \mathsf{x} - 1 = Z\ \}\ \mathsf{x}\texttt{:=}\mathsf{x} - 1\ \{\ \Downarrow \mathsf{x} = Z\ \}}{\{\ \mathsf{x}\#0 \wedge \mathsf{x} = Z + 1\ \}\ \mathsf{x}\texttt{:=}\mathsf{x} - 1\ \{\ \Downarrow \mathsf{x} = Z\ \}}}{\{\ \textit{true}\ \}\ \texttt{while}\ \mathsf{x}\#0\ \texttt{do}\ \mathsf{x}\texttt{:=}\mathsf{x} - 1\ \texttt{end}\ \{\ \Downarrow \mathsf{x} = 0\ \}}$$

- This derivation is not **sound**
- We cannot prove $\exists Z \in \mathbb{N} : \mathbf{V}(Z)$ for $\mathsf{x} < 0$

Incompleteness: Procedures

$$\frac{\{\,\mathbf{P}\,\}\,\texttt{call}\,p\,\{\,\mathbf{Q}\,\} \vdash \{\,\mathbf{P}\,\}\,s\,\{\,\mathbf{Q}\,\}}{\{\,\mathbf{P}\,\}\,\texttt{call}\,p\,\{\,\mathbf{Q}\,\}}$$

where p is defined by `proc` p `is` s `end`

```
proc p is
  if y >0 then
    y := y - 1;
    x := x - 1; call p; x := x + 1;
  end
end
```

- We cannot prove

  $\{\,\mathtt{x} = N\,\}\,\texttt{call p}\,\{\,\mathtt{x} = N\,\} \vdash \{\,\mathtt{x} = N\,\}\,\mathit{body}(\mathtt{p})\,\{\,\mathtt{x} = N\,\}$

  because the assumption does not match the recursive call

Soundness and Completeness

- Soundess and completeness can be proved w.r.t. an operational or denotational semantics

The partial correctness assertion $\{ \mathbf{P} \}\ s\ \{ \mathbf{Q} \}$ is **valid**—written as $\models \{ \mathbf{P} \}\ s\ \{ \mathbf{Q} \}$— iff

$$\forall \sigma, \sigma' \in \text{State} : \mathbf{P}(\sigma) = tt \wedge \langle s, \sigma \rangle \rightarrow \sigma' \Rightarrow \mathbf{Q}(\sigma') = tt$$

- **Soundness**: $\vdash \{ \mathbf{P} \}\ s\ \{ \mathbf{Q} \} \Rightarrow \models \{ \mathbf{P} \}\ s\ \{ \mathbf{Q} \}$
- **Completeness**: $\models \{ \mathbf{P} \}\ s\ \{ \mathbf{Q} \} \Rightarrow \vdash \{ \mathbf{P} \}\ s\ \{ \mathbf{Q} \}$

Theorem

Soundess and completeness theorem

For all partial correctness assertions $\{ \mathbf{P} \}\ s\ \{ \mathbf{Q} \}$ of IMP we have

$$\vdash \{ \mathbf{P} \}\ s\ \{ \mathbf{Q} \} \Leftrightarrow \vDash \{ \mathbf{P} \}\ s\ \{ \mathbf{Q} \}$$

4. Axiomatic Semantics

4.1 Hoare Logic

4.2 Soundness and Completeness

4.2.1 Proof of Soundness

4.2.2 Proof of Completeness

Soundness Proof

- We prove $\vdash \{\ \mathbf{P}\ \}\ s\ \{\ \mathbf{Q}\ \} \Rightarrow \models \{\ \mathbf{P}\ \}\ s\ \{\ \mathbf{Q}\ \}$
- That is, we have to show

$$\vdash \{\ \mathbf{P}\ \}\ s\ \{\ \mathbf{Q}\ \} \wedge \mathbf{P}(\sigma) = tt \wedge \langle s, \sigma \rangle \rightarrow \sigma' \Rightarrow \mathbf{Q}(\sigma') = tt$$

- The proof runs by induction on the shape of the inference tree for $\vdash \{\ \mathbf{P}\ \}\ s\ \{\ \mathbf{Q}\ \}$

Soundness Proof: Base Cases

- Case assign-axiom
  - Assume $\langle x := e, \sigma \rangle \rightarrow \sigma'$
  - We have to prove $(\mathbf{P}[x \mapsto \mathcal{A}[\![e]\!]])\sigma = tt \Rightarrow \mathbf{P}(\sigma') = tt$
  - From the natural semantics, we get $\langle x := e, \sigma \rangle \rightarrow \sigma[x \mapsto \mathcal{A}[\![e]\!]\sigma]$
  - We have $(\mathbf{P}[x \mapsto \mathcal{A}[\![e]\!]])\sigma = tt \Leftrightarrow \mathbf{P}(\sigma[x \mapsto \mathcal{A}[\![e]\!]\sigma]) = tt$
- Case skip-axiom: Trivial

Soundness Proof: Composition

- Consider arbitrary states σ and σ'' where $\mathbf{P}(\sigma) = tt$ holds and $\langle s_1 ; s_2, \sigma \rangle \rightarrow \sigma''$
- From the natural semantics, we know that there is a state σ' such that $\langle s_1, \sigma \rangle \rightarrow \sigma'$ and $\langle s_2, \sigma' \rangle \rightarrow \sigma''$
- From the induction hypothesis, we get $\models \{ \mathbf{P} \}\ s_1\ \{ \mathbf{Q} \}$ and $\models \{ \mathbf{Q} \}\ s_2\ \{ \mathbf{R} \}$
- From $\models \{ \mathbf{P} \}\ s_1\ \{ \mathbf{Q} \}$, $\langle s_1, \sigma \rangle \rightarrow \sigma'$, and $\mathbf{P}(\sigma) = tt$, we get $\mathbf{Q}(\sigma') = tt$
- From $\models \{ \mathbf{Q} \}\ s_2\ \{ \mathbf{R} \}$, $\langle s_2, \sigma' \rangle \rightarrow \sigma''$, and $\mathbf{Q}(\sigma') = tt$, we get $\mathbf{R}(\sigma'') = tt$

Soundness Proof: Conditional

- Case 1: $\mathcal{B}[\![b]\!]\sigma = tt$
  - Consider arbitrary states σ and σ' where $\mathbf{P}(\sigma) = tt$ holds and $\langle \texttt{if}\ b\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{end}, \sigma\rangle \rightarrow \sigma'$
  - From the natural semantics, we get $\langle s_1, \sigma\rangle \rightarrow \sigma'$
  - From the induction hypothesis, we get $\models \{\ \mathcal{B}[\![b]\!] \wedge \mathbf{P}\ \}\ s_1\ \{\ \mathbf{Q}\ \}$
  - From $\mathbf{P}(\sigma) = tt$ and $\mathcal{B}[\![b]\!]\sigma = tt$, we get $(\mathcal{B}[\![b]\!] \wedge \mathbf{P})\sigma = tt$
  - From $\models \{\ \mathcal{B}[\![b]\!] \wedge \mathbf{P}\ \}\ s_1\ \{\ \mathbf{Q}\ \}$ and $(\mathcal{B}[\![b]\!] \wedge \mathbf{P})\sigma = tt$, we get $\mathbf{Q}(\sigma'') = tt$
- Case 2: $\mathcal{B}[\![b]\!]\sigma = ff$ is analogous

Soundness Proof: Loop

- We have to prove

$$\vdash \{\ \mathbf{P}\ \}\ \texttt{while}\ b\ \texttt{do}\ s\ \texttt{end}\ \{\ \neg\mathcal{B}[\![b]\!] \wedge \mathbf{P}\ \} \wedge$$
$$\mathbf{P}(\sigma) = tt \wedge \langle \texttt{while}\ b\ \texttt{do}\ s\ \texttt{end}, \sigma\rangle \rightarrow \sigma''$$
$$\Rightarrow (\neg\mathcal{B}[\![b]\!] \wedge \mathbf{P})\sigma''$$

  where σ and σ'' are arbitrary states

- The proof runs by induction on the shape of the derivation tree for $\langle \texttt{while}\ b\ \texttt{do}\ s\ \texttt{end}, \sigma\rangle \rightarrow \sigma''$

Soundness Proof: Loop (cont'd)

- Case 1: $\mathcal{B}[\![b]\!]\sigma = tt$
  - From the natural semantics, we get $\langle s, \sigma\rangle \rightarrow \sigma'$ and $\langle \texttt{while}\ b\ \texttt{do}\ s\ \texttt{end}, \sigma'\rangle \rightarrow \sigma''$
  - From $\mathbf{P}(\sigma) = tt$ and $\mathcal{B}[\![b]\!]\sigma = tt$, we get $(\mathcal{B}[\![b]\!] \wedge \mathbf{P})\sigma = tt$
  - By applying the induction hypothesis of the outer induction to $\vDash \{\ \mathcal{B}[\![b]\!] \wedge \mathbf{P}\ \}\ s\ \{\ \mathbf{P}\ \}$, we get $\mathbf{P}(\sigma') = tt$
  - Now we can apply the induction hypothesis of the nested induction to $\langle \texttt{while}\ b\ \texttt{do}\ s\ \texttt{end}, \sigma'\rangle \rightarrow \sigma''$ to get $(\neg\mathcal{B}[\![b]\!] \wedge \mathbf{P})\sigma'' = tt$
- Case 2: $\mathcal{B}[\![b]\!]\sigma = ff$
  - From the natural semantics, we get $\sigma = \sigma''$
  - $\mathbf{P}(\sigma) = tt$ and $\mathcal{B}[\![b]\!]\sigma = ff$ imply $(\neg\mathcal{B}[\![b]\!] \wedge \mathbf{P})\sigma'' = tt$

Soundness Proof: Consequence

- Consider arbitrary states σ and σ' where $\mathbf{P}(\sigma) = tt$ holds and $\langle s, \sigma \rangle \rightarrow \sigma'$
- We have $\vDash \{ \mathbf{P}' \}\ s\ \{ \mathbf{Q}' \}$, $\mathbf{P} \Rightarrow \mathbf{P}'$, and $\mathbf{Q}' \Rightarrow \mathbf{Q}$
- From $\mathbf{P}(\sigma) = tt$ and $\mathbf{P} \Rightarrow \mathbf{P}'$, we get $\mathbf{P}'(\sigma) = tt$
- By applying the induction hypothesis, we get $\mathbf{Q}'(\sigma') = tt$
- From $\mathbf{Q}'(\sigma') = tt$ and $\mathbf{Q}' \Rightarrow \mathbf{Q}$, we get $\mathbf{Q}(\sigma') = tt$

4. Axiomatic Semantics

4.1 Hoare Logic

4.2 Soundness and Completeness

4.2.1 Proof of Soundness

4.2.2 Proof of Completeness

Weakest (Liberal) Preconditions

- The weakest precondition of a statement s and a postcondition $\mathbf{Q}$ is the weakest predicate that has to hold in the initial state of an execution of s to guarantee that $\mathbf{Q}$ holds in the final state
  - The weakest precondition $wp(s, \mathbf{Q})$ guarantees termination
  - The weakest **liberal** precondition $wlp(s, \mathbf{Q})$ does not guarantee termination

$$wp(s, \mathbf{Q})\sigma = tt \quad \Leftrightarrow \exists \sigma' : (\langle s, \sigma \rangle \rightarrow \sigma' \wedge \mathbf{Q}(\sigma'))$$
$$wlp(s, \mathbf{Q})\sigma = tt \quad \Leftrightarrow \forall \sigma' : (\langle s, \sigma \rangle \rightarrow \sigma' \Rightarrow \mathbf{Q}(\sigma'))$$

- In the following, we consider partial correctness

wlp-Lemma

Lemma: For every statement s and predicate $\mathbf{Q}$ we have

1. $\models \{\ \mathit{wlp}(s, \mathbf{Q})\ \}\ s\ \{\ \mathbf{Q}\ \}$
2. $\models \{\ \mathbf{P}\ \}\ s\ \{\ \mathbf{Q}\ \} \Rightarrow (\mathbf{P} \Rightarrow \mathit{wlp}(s, \mathbf{Q}))$

- Proof 1:
  - Let $\mathit{wlp}(s, \mathbf{Q})\sigma = tt$ and $\langle s, \sigma\rangle \rightarrow \sigma'$
  - From the definition of *wlp*, we get $\mathbf{Q}(\sigma')$
- Proof 2:
  - Let $\mathbf{P}(\sigma) = tt$ and $\langle s, \sigma\rangle \rightarrow \sigma'$
  - From $\models \{\ \mathbf{P}\ \}\ s\ \{\ \mathbf{Q}\ \}$, we get $\mathbf{Q}(\sigma') = tt$
  - From the definition of *wlp*, we get $\mathit{wlp}(s, \mathbf{Q})\sigma'$

Completeness Proof

- We prove $\models \{ \mathbf{P} \}\ s\ \{ \mathbf{Q} \} \Rightarrow \vdash \{ \mathbf{P} \}\ s\ \{ \mathbf{Q} \}$
- It suffices to infer $\vdash \{ \mathit{wlp}(s, \mathbf{Q}) \}\ s\ \{ \mathbf{Q} \}$

  - By $\models \{ \mathbf{P} \}\ s\ \{ \mathbf{Q} \}$, the *wlp*-lemma implies $\mathbf{P} \Rightarrow \mathit{wlp}(s, \mathbf{Q})$

$$\frac{\{ \mathit{wlp}(s, \mathbf{Q}) \}\ s\ \{ \mathbf{Q} \}}{\{ \mathbf{P} \}\ s\ \{ \mathbf{Q} \}}$$

- We prove $\vdash \{ \mathit{wlp}(s, \mathbf{Q}) \}\ s\ \{ \mathbf{Q} \}$ by structural induction on s

Completeness Proof: Base Cases

- Case assign-axiom
  - From the natural semantics, we get $\langle x := e, \sigma \rangle = \sigma[x \mapsto \mathcal{A}[\![e]\!]\sigma]$
  - From the definition of *wlp*, we get $\mathit{wlp}(x := e, \mathbf{Q})\sigma \Leftrightarrow \mathbf{Q}(\sigma[x \mapsto \mathcal{A}[\![e]\!]\sigma])$
  - Therefore, we get $\mathit{wlp}(x := e, \mathbf{Q}) = \mathbf{Q}[x \mapsto \mathcal{A}[\![e]\!]]$
  - We can infer $\vdash \{ \mathbf{Q}[x \mapsto \mathcal{A}[\![e]\!]] \}\ x := e\ \{ \mathbf{Q} \}$

- Case skip-axiom:
  - From the natural semantics, we get $\mathit{wlp}(\texttt{skip}, \mathbf{Q}) = \mathbf{Q}$
  - We can infer $\vdash \{ \mathbf{Q} \}\ \texttt{skip}\ \{ \mathbf{Q} \}$

Completeness Proof: Composition

- By the induction hypothesis, we get
  $\vdash \{\ \mathit{wlp}(s_2, \mathbf{Q})\ \}\ s_2\ \{\ \mathbf{Q}\ \}$ and
  $\vdash \{\ \mathit{wlp}(s_1, \mathit{wlp}(s_2, \mathbf{Q}))\ \}\ s_1\ \{\ \mathit{wlp}(s_2, \mathbf{Q})\ \}$
- We can infer $\vdash \{\ \mathit{wlp}(s_1, \mathit{wlp}(s_2, \mathbf{Q}))\ \}\ s_1 ; s_2\ \{\ \mathbf{Q}\ \}$
- It remains to prove that
  $\mathit{wlp}(s_1 ; s_2, \mathbf{Q}) \Rightarrow \mathit{wlp}(s_1, \mathit{wlp}(s_2, \mathbf{Q}))$
- We assume that $\mathit{wlp}(s_1 ; s_2, \mathbf{Q})\sigma = tt$ and show that
  $\mathit{wlp}(s_1, \mathit{wlp}(s_2, \mathbf{Q}))\sigma = tt$

Completeness Proof: Composition (2)

- If there is no σ' such that $\langle s_1, \sigma \rangle \rightarrow \sigma'$ then *wlp*$(s_1,$ *wlp*$(s_2, \mathbf{Q}))\sigma = tt$ follows immediately from the definition of *wlp*
- Otherwise, we have to show *wlp*$(s_2, \mathbf{Q})\sigma' = tt$
- Again, if there is no σ'' such that $\langle s_2, \sigma' \rangle \rightarrow \sigma''$ then *wlp*$(s_2, \mathbf{Q})\sigma' = tt$ follows immediately from the definition of *wlp*
- Otherwise, we have to show $\mathbf{Q}(\sigma'')$
- $\mathbf{Q}(\sigma'')$ follows from *wlp*$(s_1 ; s_2, \mathbf{Q})\sigma = tt$ and $\langle s_1 ; s_2, \sigma \rangle \rightarrow \sigma''$

Completeness Proof: Conditional

- By the induction hypothesis, we get
  $\vdash \{\ \mathit{wlp}(s_1, \mathbf{Q})\ \}\ s_1\ \{\ \mathbf{Q}\ \}$ and
  $\vdash \{\ \mathit{wlp}(s_2, \mathbf{Q})\ \}\ s_2\ \{\ \mathbf{Q}\ \}$
- Define $\mathbf{P} \equiv (\mathcal{B}[\![b]\!] \wedge \mathit{wlp}(s_1, \mathbf{Q})) \vee (\neg\mathcal{B}[\![b]\!] \wedge \mathit{wlp}(s_2, \mathbf{Q}))$
- We have $\mathcal{B}[\![b]\!] \wedge \mathbf{P} \Rightarrow \mathit{wlp}(s_1, \mathbf{Q})$ and
  $\neg\mathcal{B}[\![b]\!] \wedge \mathbf{P} \Rightarrow \mathit{wlp}(s_2, \mathbf{Q})$
- We derive

$$\frac{\dfrac{\{\ \mathit{wlp}(s_1, \mathbf{Q})\ \}\ s_1\ \{\ \mathbf{Q}\ \}}{\{\ \mathcal{B}[\![b]\!] \wedge \mathbf{P}\ \}\ s_1\ \{\ \mathbf{Q}\ \}} \qquad \frac{\{\ \mathit{wlp}(s_2, \mathbf{Q})\ \}\ s_2\ \{\ \mathbf{Q}\ \}}{\{\ \neg\mathcal{B}[\![b]\!] \wedge \mathbf{P}\ \}\ s_2\ \{\ \mathbf{Q}\ \}}}{\{\ \mathbf{P}\ \}\ \texttt{if}\ b\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{end}\ \{\ \mathbf{Q}\ \}}$$

Completeness Proof: Conditional (2)

- We have
  $\mathbf{P} \equiv (\mathcal{B}[\![b]\!] \wedge \mathit{wlp}(s_1, \mathbf{Q})) \vee (\neg\mathcal{B}[\![b]\!] \wedge \mathit{wlp}(s_2, \mathbf{Q}))$
- It remains to show that
  $\mathit{wlp}(\texttt{if}\ b\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{end}, \mathbf{Q})\sigma = tt \Rightarrow \mathbf{P}(\sigma) = tt$
- Case 1: $\mathcal{B}[\![b]\!]\sigma = tt$
  - If there is no σ' such that $\langle s_1, \sigma\rangle \rightarrow \sigma'$ then $\mathit{wlp}(s_1, \mathbf{Q})\sigma = tt$ follows immediately from the definition of *wlp*
  - Otherwise, we have to prove $\mathbf{Q}(\sigma')$
  - From $\mathit{wlp}(\texttt{if}\ b\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{end}, \mathbf{Q})\sigma = tt$ and $\langle \texttt{if}\ b\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{end}, \sigma\rangle \rightarrow \sigma'$, we get $\mathbf{Q}(\sigma')$
- Case 2: $\mathcal{B}[\![b]\!]\sigma = ff$ is analogous

Completeness Proof: Loop

- Define $\mathbf{P} \equiv \mathit{wlp}(\texttt{while}\ b\ \texttt{do}\ s\ \texttt{end}, \mathbf{Q})$
- We will prove

  (1) $(\neg \mathcal{B}[\![b]\!] \wedge \mathbf{P}) \Rightarrow \mathbf{Q}$

  (2) $(\mathcal{B}[\![b]\!] \wedge \mathbf{P}) \Rightarrow \mathit{wlp}(s, \mathbf{P})$
- By the induction hypothesis, we get
  $\vdash \{\ \mathit{wlp}(s, \mathbf{P})\ \}\ s\ \{\ \mathbf{P}\ \}$
- From (2), we get $\vdash \{\ \mathcal{B}[\![b]\!] \wedge \mathbf{P}\ \}\ s\ \{\ \mathbf{P}\ \}$
- Be the while rule, we get
  $\vdash \{\ \mathbf{P}\ \}\ \texttt{while}\ b\ \texttt{do}\ s\ \texttt{end}\ \{\ \neg \mathcal{B}[\![b]\!] \wedge \mathbf{P}\ \}$
- From (1), we get $\vdash \{\ \mathbf{P}\ \}\ \texttt{while}\ b\ \texttt{do}\ s\ \texttt{end}\ \{\ \mathbf{Q}\ \}$

Completeness Proof: Loop (2)

- We prove (1): $(\neg \mathcal{B}[\![b]\!] \wedge \mathbf{P}) \Rightarrow \mathbf{Q}$
- Assume $(\neg \mathcal{B}[\![b]\!] \wedge \mathbf{P})\sigma = tt$
- Then we have $\langle \texttt{while}\ b\ \texttt{do}\ s\ \texttt{end}, \sigma \rangle = \sigma$
- By *wlp*$(\texttt{while}\ b\ \texttt{do}\ s\ \texttt{end}, \mathbf{Q})\sigma = tt$ and the definition of *wlp*, we get $\mathbf{Q}(\sigma) = tt$

Completeness Proof: Loop (3)

- We prove (2): $(\mathcal{B}[\![b]\!] \wedge \mathbf{P}) \Rightarrow \mathit{wlp}(s, \mathbf{P})$
- We assume $(\mathcal{B}[\![b]\!] \wedge \mathbf{P})\sigma = tt$ and show that $\mathit{wlp}(s, \mathbf{P})\sigma = tt$
- If there is no σ' such that $\langle s, \sigma \rangle \rightarrow \sigma'$ then $\mathit{wlp}(s, \mathbf{P})\sigma = tt$ follows immediately from the definition of *wlp*
- Otherwise, we have to show $\mathbf{P}(\sigma') = tt$

Completeness Proof: Loop (4)

- Case 1: There is no σ'' such that $\langle \texttt{while}\ b\ \texttt{do}\ s\ \texttt{end}, \sigma' \rangle = \sigma''$
  - By the definition of *wlp*, we get that $\mathit{wlp}(\texttt{while}\ b\ \texttt{do}\ s\ \texttt{end}, \mathbf{Q})\sigma' = tt$ and, thus, $\mathbf{P}(\sigma') = tt$
- Case 2: There is a σ'' such that $\langle \texttt{while}\ b\ \texttt{do}\ s\ \texttt{end}, \sigma' \rangle = \sigma''$
  - From $\langle s, \sigma \rangle \rightarrow \sigma'$ and $\langle \texttt{while}\ b\ \texttt{do}\ s\ \texttt{end}, \sigma' \rangle = \sigma''$, we get $\langle \texttt{while}\ b\ \texttt{do}\ s\ \texttt{end}, \sigma \rangle = \sigma''$
  - By $\mathbf{P}(\sigma) = tt$ and $\langle \texttt{while}\ b\ \texttt{do}\ s\ \texttt{end}, \sigma \rangle = \sigma''$, we get $\mathbf{Q}(\sigma'') = tt$
  - By $\mathbf{Q}(\sigma'') = tt$ and $\langle \texttt{while}\ b\ \texttt{do}\ s\ \texttt{end}, \sigma' \rangle = \sigma''$, we get $\mathit{wlp}(\texttt{while}\ b\ \texttt{do}\ s\ \texttt{end}, \mathbf{Q})\sigma' = tt$ and, thus, $\mathbf{P}(\sigma') = tt$

Summary: Axiomatic Semantics

- Axiomatic semantics
  - expresses **specific properties** of the effect of executing a program
  - Some aspects of the computation may be ignored
- Axiomatic semantics is used to verify programs
  - Partial correctness
  - Total correctness
  - Other properties, e.g., resource consumption
- The inference system should be **sound** and **complete**