



# Semantics and Verification of Software

Winter Semester 2017/18

Lecture 11: Axiomatic Semantics of WHILE III  
(Total Correctness & Axiomatic Equivalence)

Thomas Noll

Software Modeling and Verification Group

RWTH Aachen University

<http://moves.rwth-aachen.de/teaching/ws-1718/sv-sw/>

# Recap: Hoare Logic

## Hoare Logic

**Goal:** syntactic derivation of valid partial correctness properties. Here  $A[x \mapsto a]$  denotes the syntactic replacement of every occurrence of  $x$  by  $a$  in  $A$ .



Tony Hoare (\* 1934)

### Definition (Hoare Logic)

The **Hoare rules** are given by

$$\begin{array}{c} \text{(skip)} \frac{}{\{A\} \text{ skip } \{A\}} \\ \text{(seq)} \frac{\{A\} c_1 \{C\} \quad \{C\} c_2 \{B\}}{\{A\} c_1 ; c_2 \{B\}} \\ \text{(while)} \frac{\{A \wedge b\} c \{A\}}{\{A\} \text{ while } b \text{ do } c \text{ end } \{A \wedge \neg b\}} \\ \text{(asgn)} \frac{}{\{A[x \mapsto a]\} x := a \{A\}} \\ \text{(if)} \frac{\{A \wedge b\} c_1 \{B\} \quad \{A \wedge \neg b\} c_2 \{B\}}{\{A\} \text{ if } b \text{ then } c_1 \text{ else } c_2 \text{ end } \{B\}} \\ \text{(cons)} \frac{\models (A \Rightarrow A') \quad \{A'\} c \{B'\} \quad \models (B' \Rightarrow B)}{\{A\} c \{B\}} \end{array}$$

A partial correctness property is **provable** (notation:  $\vdash \{A\} c \{B\}$ ) if it is derivable by the Hoare rules. In (while),  $A$  is called a **(loop) invariant**.

## Recap: Hoare Logic

---

### Soundness of Hoare Logic

#### Theorem (Soundness of Hoare Logic)

For every partial correctness property  $\{A\} c \{B\}$ ,

$$\vdash \{A\} c \{B\} \quad \Rightarrow \quad \models \{A\} c \{B\}.$$

#### Proof.

Let  $\vdash \{A\} c \{B\}$ . By induction over the structure of the corresponding proof tree we show that, for every  $\sigma \in \Sigma$  and  $I \in Int$  such that  $\sigma \models' A$ ,  $\mathcal{C}[[c]]\sigma \models' B$  (on the board). (If  $\sigma = \perp$ , then  $\mathcal{C}[[c]]\sigma = \perp \models' B$  holds trivially.)  $\square$

## Recap: Hoare Logic

### Incompleteness of Hoare Logic I

**Soundness:** only valid partial correctness properties are provable ✓

**Completeness:** all valid partial correctness properties are systematically derivable ⚡

#### Theorem (Gödel's Incompleteness Theorem)

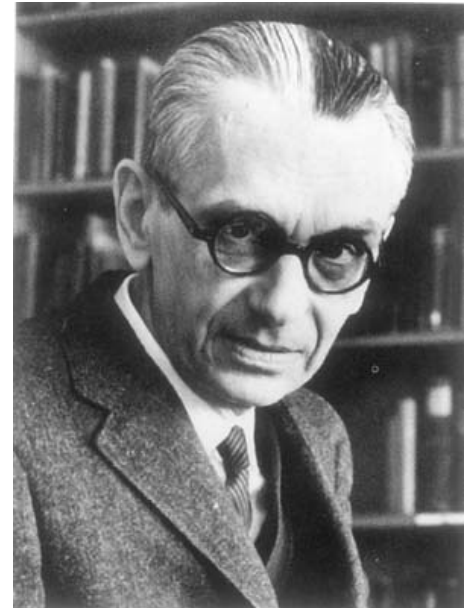
*The set of all valid assertions*

$$\{A \in Assn \mid \models A\}$$

*is not recursively enumerable, i.e., there exists no proof system for  $Assn$  in which all valid assertions are systematically derivable.*

#### Proof.

see [Winskel 1996, p. 110 ff] □



Kurt Gödel  
(1906–1978)

## Recap: Hoare Logic

---

### Incompleteness of Hoare Logic II

#### Corollary

*There is no proof system in which all valid partial correctness properties can be enumerated.*

#### Proof.

Given  $A \in Assn$ ,  $\models A$  is obviously equivalent to  $\{\text{true}\} \text{skip} \{A\}$ . Thus the enumerability of all valid partial correctness properties would imply the enumerability of all valid assertions. □

**Remark:** alternative proof (using computability theory):

$\{\text{true}\} c \{\text{false}\}$  is valid iff  $c$  does not terminate on any input state. But the set of all non-terminating WHILE statements is not enumerable.

## Recap: Hoare Logic

---

### Relative Completeness of Hoare Logic II

#### Theorem (Cook's Completeness Theorem)

Hoare Logic is *relatively complete*, i.e., for every partial correctness property  $\{A\} c \{B\}$ :

$$\models \{A\} c \{B\} \Rightarrow \vdash \{A\} c \{B\}.$$



Stephen A. Cook (\* 1939)

Thus: if we know that a partial correctness property is valid, then we know that there is a corresponding derivation.

# Total Correctness

---

## Total Correctness

- **Observation:** partial correctness properties only speak about **terminating** computations of a given program
- **Total correctness** additionally requires the proof that the program indeed stops (on the input states admitted by the precondition)
- Consider **total correctness properties** of the form

$$\{A\} c \{\Downarrow B\}$$

where  $c \in \text{Cmd}$  and  $A, B \in \text{Assn}$

- Interpretation:

Validity of property  $\{A\} c \{\Downarrow B\}$

For all states  $\sigma \in \Sigma$  which satisfy  $A$ :

the execution of  $c$  in  $\sigma$  **terminates** and yields a state which satisfies  $B$ .

# Total Correctness

## Semantics of Total Correctness Properties

### Definition 11.1 (Semantics of total correctness properties)

Let  $A, B \in Assn$  and  $c \in Cmd$ .

- $\{A\} c \{\Downarrow B\}$  is called **valid in**  $\sigma \in \Sigma$  and  $I \in Int$  (notation:  $\sigma \models' \{A\} c \{\Downarrow B\}$ ) if  $\sigma \models' A$  implies that  $\mathcal{C}[[c]]\sigma \neq \perp$  and  $\mathcal{C}[[c]]\sigma \models' B$ .
- $\{A\} c \{\Downarrow B\}$  is called **valid in**  $I \in Int$  (notation:  $\models' \{A\} c \{\Downarrow B\}$ ) if  $\sigma \models' \{A\} c \{\Downarrow B\}$  for every  $\sigma \in \Sigma$ .
- $\{A\} c \{\Downarrow B\}$  is called **valid** (notation:  $\models \{A\} c \{\Downarrow B\}$ ) if  $\models' \{A\} c \{\Downarrow B\}$  for every  $I \in Int$ .

Obviously, total implies partial correctness (but not vice versa):

### Corollary 11.2

For all  $A, B \in Assn$  and  $c \in Cmd$ ,

$$\models \{A\} c \{\Downarrow B\} \Rightarrow \models \{A\} c \{B\}.$$



# Total Correctness

## Proving Total Correctness I

**Goal:** syntactic derivation of valid total correctness properties

Definition 11.3 (Hoare Logic for total correctness)

The **Hoare rules for total correctness** are given by (where  $i \in LVar$ )

$$\begin{array}{c} \text{(skip)} \frac{}{\{A\} \text{ skip } \{\Downarrow A\}} \\ \text{(seq)} \frac{\{A\} c_1 \{\Downarrow C\} \quad \{C\} c_2 \{\Downarrow B\}}{\{A\} c_1 ; c_2 \{\Downarrow B\}} \\ \text{(if)} \frac{\{A \wedge b\} c_1 \{\Downarrow B\} \quad \{A \wedge \neg b\} c_2 \{\Downarrow B\}}{\{A\} \text{ if } b \text{ then } c_1 \text{ else } c_2 \text{ end } \{\Downarrow B\}} \\ \text{(while)} \frac{\vdash (i \geq 0 \wedge A(i+1) \Rightarrow b) \quad \{i \geq 0 \wedge A(i+1)\} c \{\Downarrow A(i)\} \quad \vdash (A(0) \Rightarrow \neg b)}{\{\exists i. i \geq 0 \wedge A(i)\} \text{ while } b \text{ do } c \text{ end } \{\Downarrow A(0)\}} \\ \text{(cons)} \frac{\vdash (A \Rightarrow A') \quad \{A'\} c \{\Downarrow B'\} \quad \vdash (B' \Rightarrow B)}{\{A\} c \{\Downarrow B\}} \end{array}$$

A total correctness property is **provable** (notation:  $\vdash \{A\} c \{\Downarrow B\}$ ) if it is derivable by the Hoare rules. In case of (while),  $A(i)$  is called a **(loop) invariant**.

# Total Correctness

---

## Proving Total Correctness II

- In rule

$$\frac{\models (i \geq 0 \wedge A(i+1) \Rightarrow b) \quad \{i \geq 0 \wedge A(i+1)\} c \{\Downarrow A(i)\} \quad \models (A(0) \Rightarrow \neg b)}{\text{(while)} \quad \frac{}{\models \{\exists i. i \geq 0 \wedge A(i)\} \text{ while } b \text{ do } c \text{ end } \{\Downarrow A(0)\}}}$$

the notation  $A(i)$  indicates that assertion  $A$  **parametrically depends** on the value of the logical variable  $i \in LVar$ .

- Idea:  $i$  represents the **remaining number of loop iterations**
- Loop to be traversed  $i + 1$  times ( $i \geq 0$ )
  - $\Rightarrow A(i + 1)$  holds
  - $\Rightarrow$  execution condition  $b$  satisfied

Thus:  $\models (i \geq 0 \wedge A(i + 1) \Rightarrow b)$ , and  $i + 1$  decreased to  $i$  after execution of  $c$

- Execution terminated
  - $\Rightarrow A(0)$  holds
  - $\Rightarrow$  execution condition  $b$  violated

Thus:  $\models (A(0) \Rightarrow \neg b)$

## Total Correctness of Factorial Program I

### Example 11.4

Proof of  $\{A\} y:=1; c \{\Downarrow B\}$  where

$$A := (x > 0 \wedge x = i)$$

$$c := \text{while } \neg(x=1) \text{ do } y:=y*x; x:=x-1 \text{ end}$$

$$B := (y = i!)$$

First we show that the assertion  $C(j) = (x > 0 \wedge y * x! = i! \wedge x = j + 1)$  is an invariant of  $c$ . Applying (asgn) twice yields

$$\vdash \{j \geq 0 \wedge C(j)[x \mapsto x-1]\} x:=x-1 \{\Downarrow j \geq 0 \wedge C(j)\} \quad \text{and}$$

$$\vdash \{j \geq 0 \wedge C(j)[x \mapsto x-1][y \mapsto y*x]\} y:=y*x \{\Downarrow j \geq 0 \wedge C(j)[x \mapsto x-1]\}$$

such that (seq) implies

$$\vdash \{j \geq 0 \wedge C(j)[x \mapsto x-1][y \mapsto y*x]\} y:=y*x; x:=x-1 \{\Downarrow j \geq 0 \wedge C(j)\}.$$

Now  $C(j+1) = (x > 0 \wedge y*x! = i! \wedge x = j+2)$  and

$$C(j)[x \mapsto x-1][y \mapsto y*x] = (x-1 > 0 \wedge y * x * (x-1)! = i! \wedge x-1 = j+1)$$

such that

$$\models ((j \geq 0 \wedge C(j+1)) \Rightarrow (j \geq 0 \wedge C(j)[x \mapsto x-1][y \mapsto y*x])) \text{ and}$$

$$\models ((j \geq 0 \wedge C(j)) \Rightarrow C(j)).$$

# Total Correctness

## Total Correctness of Factorial Program II

Example 11.4 (continued;  $C(j) = (x > 0 \wedge y * x! = i! \wedge x = j + 1)$ )

Hence (cons) implies

$$\vdash \{j \geq 0 \wedge C(j+1)\} y := y * x; x := x - 1 \{\Downarrow C(j)\}.$$

Moreover we have

$$\models ((j \geq 0 \wedge C(j+1)) \Rightarrow \neg(x = 1)) \text{ and } \models (C(0) \Rightarrow \neg(\neg(x = 1)))$$

such that (while) yields

$$\vdash \{\exists j. j \geq 0 \wedge C(j)\} c \{\Downarrow C(0)\}.$$

For the initializing assignment, (asgn) implies

$$\vdash \{\exists j. j \geq 0 \wedge C(j)[y \mapsto 1]\} y := 1 \{\Downarrow \exists j. j \geq 0 \wedge C(j)\},$$

such that (seq) allows to conclude

$$\vdash \{\exists j. j \geq 0 \wedge C(j)[y \mapsto 1]\} y := 1; c \{\Downarrow C(0)\}.$$

On the other hand we have (choose  $j := i - 1$ ):

$$\models ((x > 0 \wedge x = i) \Rightarrow (\exists j. j \geq 0 \wedge C(j)[y \mapsto 1])) \text{ and } \models (C(0) \Rightarrow y = i!)$$

such that (cons) yields the desired result:

$$\vdash \{x > 0 \wedge x = i\} y := 1; c \{\Downarrow y = i!\}.$$

# Soundness and Completeness of Hoare Logic for Total Correctness

---

## Soundness

In analogy to Theorem 10.2 we can show that the Hoare Logic for total correctness properties is also sound:

### Theorem 11.5 (Soundness)

For every total correctness property  $\{A\} c \{\Downarrow B\}$ ,

$$\vdash \{A\} c \{\Downarrow B\} \quad \Rightarrow \quad \models \{A\} c \{\Downarrow B\}.$$

### Proof.

again by structural induction over the derivation tree of  $\vdash \{A\} c \{\Downarrow B\}$   
(here only (while) case; on the board) □

# Soundness and Completeness of Hoare Logic for Total Correctness

---

## Relative Completeness

Also the counterpart to Cook's Completeness Theorem 10.5 applies:

### Theorem 11.6 (Completeness)

The Hoare Logic for total correctness properties is *relatively complete*, i.e., for every  $\{A\} c \{\Downarrow B\}$ :

$$\models \{A\} c \{\Downarrow B\} \quad \Rightarrow \quad \vdash \{A\} c \{\Downarrow B\}.$$

Proof.

omitted □

# Axiomatic Equivalence

---

## Operational and Denotational Equivalence

Definition 4.1:  $\mathcal{D}[\cdot]$  :  $Cmd \rightarrow (\Sigma \dashrightarrow \Sigma)$  given by

$$\mathcal{D}[c]\sigma = \sigma' \iff \langle c, \sigma \rangle \rightarrow \sigma'$$

Definition 4.2: Two statements  $c_1, c_2 \in Cmd$  are **operationally equivalent** (notation:  $c_1 \sim c_2$ ) if

$$\mathcal{D}[c_1] = \mathcal{D}[c_2]$$

Theorem 8.5: For every  $c \in Cmd$ ,

$$\mathcal{D}[c] = \mathcal{C}[c]$$

# Axiomatic Equivalence

---

## Axiomatic Equivalence I

In the axiomatic semantics, two statements have to be considered equivalent if they are **indistinguishable** w.r.t. (partial) correctness properties:

### Definition 11.7 (Axiomatic equivalence)

Two statements  $c_1, c_2 \in \text{Cmd}$  are called **axiomatically equivalent** (notation:  $c_1 \approx c_2$ ) if, for all assertions  $A, B \in \text{Assn}$ ,

$$\models \{A\} c_1 \{B\} \iff \models \{A\} c_2 \{B\}.$$

(later: total correctness yields same notion of equivalence)



# Axiomatic Equivalence

## Axiomatic Equivalence II

### Example 11.8

We show that  $\text{while } b \text{ do } c \text{ end} \approx \text{if } b \text{ then } c; \text{while } b \text{ do } c \text{ end else skip end}$  (cf. Lemma 4.3). Let  $A, B \in \text{Assn}$ :

$$\begin{aligned} & \models \{A\} \text{ while } b \text{ do } c \text{ end } \{B\} \\ \iff & \vdash \{A\} \text{ while } b \text{ do } c \text{ end } \{B\} \quad (\text{Theorem 10.2, 10.5}) \\ \iff & \text{ex. } C \in \text{Assn} \text{ such that } \models (A \Rightarrow C), \models (C \wedge \neg b \Rightarrow B), \\ & \vdash \{C\} \text{ while } b \text{ do } c \text{ end } \{C \wedge \neg b\} \quad (\text{rule (cons)}) \\ \iff & \text{ex. } C \in \text{Assn} \text{ such that } \models (A \Rightarrow C), \models (C \wedge \neg b \Rightarrow B), \\ & \vdash \{C \wedge b\} c \{C\} \quad (\text{rule (while)}) \\ \iff & \text{ex. } C \in \text{Assn} \text{ such that } \models (A \Rightarrow C), \models (C \wedge \neg b \Rightarrow B), \\ & \vdash \{C \wedge b\} c; \text{while } b \text{ do } c \text{ end } \{C \wedge \neg b\} \quad (\text{rule (seq)}), \\ & \vdash \{C \wedge \neg b\} \text{ skip } \{C \wedge \neg b\} \quad (\text{rule (skip)}) \\ \iff & \text{ex. } C \in \text{Assn} \text{ such that } \models (A \Rightarrow C), \models (C \wedge \neg b \Rightarrow B), \\ & \vdash \{C\} \text{ if } b \text{ then } c; \text{while } b \text{ do } c \text{ end else skip end } \{C \wedge \neg b\} \quad (\text{rule (if)}) \\ \iff & \vdash \{A\} \text{ if } b \text{ then } c; \text{while } b \text{ do } c \text{ end else skip end } \{B\} \quad (\text{rule (cons)}) \\ \iff & \models \{A\} \text{ if } b \text{ then } c; \text{while } b \text{ do } c \text{ end else skip end } \{B\} \quad (\text{Theorem 10.2, 10.5}) \end{aligned}$$