

Semantics and Verification of Software

Lecture 11: Axiomatic Semantics of WHILE

Thomas Noll

Lehrstuhl für Informatik 2
RWTH Aachen University
noll@cs.rwth-aachen.de

<http://www-i2.informatik.rwth-aachen.de/i2/svsw/>

Summer semester 2007

- 1 Repetition: Correctness of Hoare Logic
- 2 Equivalence of Axiomatic and Operational/Denotational Semantics
- 3 Total Correctness

Theorem (Soundness of Hoare Logic)

For every partial correctness property $\{A\} c \{B\}$,
$$\vdash \{A\} c \{B\} \implies \models \{A\} c \{B\}.$$

Theorem (Gödel's Incompleteness Theorem)

The set of all valid assertions

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

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

Theorem (Cook's Completeness Theorem)

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

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

- 1 Repetition: Correctness of Hoare Logic
- 2 Equivalence of Axiomatic and Operational/Denotational Semantics
- 3 Total Correctness

Def. 4.3: $\mathfrak{D}[\![\cdot]\!]$: $Cmd \rightarrow (\Sigma \multimap \Sigma)$ given by

$$\mathfrak{D}[\![c]\!](\sigma) = \sigma' \iff \langle c, \sigma \rangle \rightarrow \sigma'$$

Def. 4.4: Two statements $c_1, c_2 \in Cmd$ are called **operationally equivalent** (notation: $c_1 \sim c_2$) if

$$\mathfrak{D}[\![c_1]\!] = \mathfrak{D}[\![c_2]\!].$$

Theorem 7.4: For every $c \in Cmd$,

$$\mathfrak{D}[\![c]\!] = \mathfrak{C}[\![c]\!],$$

$$\text{i.e., } \mathfrak{D}[\![\cdot]\!] = \mathfrak{C}[\![\cdot]\!].$$

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 11.1 (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 $A, B \in \text{Assn}$,

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

Example 11.2

We show that $c_1; (c_2; c_3) \approx (c_1; c_2); c_3$. Let $A, B \in \text{Assn}$:

$$\begin{aligned} & \models \{A\} c_1; (c_2; c_3) \{B\} \\ \iff & \vdash \{A\} c_1; (c_2; c_3) \{B\} \text{ (Theorem 9.5, 10.3)} \\ \iff & \text{ex. } C_1 \in \text{Assn} \text{ such that } \vdash \{A\} c_1 \{C_1\}, \vdash \{C_1\} c_2; c_3 \{B\} \text{ (rule (seq))} \\ \iff & \text{ex. } C_1, C_2 \in \text{Assn} \text{ such that } \vdash \{A\} c_1 \{C_1\}, \vdash \{C_1\} c_2 \{C_2\}, \\ & \vdash \{C_2\} c_3 \{B\} \text{ (rule (seq))} \\ \iff & \text{ex. } C_2 \in \text{Assn} \text{ such that } \vdash \{A\} c_1; c_2 \{C_2\}, \vdash \{C_2\} c_3 \{B\} \text{ (rule (seq))} \\ \iff & \vdash \{A\} (c_1; c_2); c_3 \{B\} \text{ (rule (seq))} \\ \iff & \models \{A\} (c_1; c_2); c_3 \{B\} \text{ (Theorem 9.5, 10.3)} \end{aligned}$$

Theorem 11.3

Axiomatic and denotational/operational equivalence coincide, i.e., for all $c_1, c_2 \in \text{Cmd}$,

$$c_1 \approx c_2 \iff c_1 \sim c_2.$$

Proof.

on the board



- 1 Repetition: Correctness of Hoare Logic
- 2 Equivalence of Axiomatic and Operational/Denotational Semantics
- 3 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 specified 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 σ terminates and yields a state which satisfies B .

Definition 11.4 (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: $\sigma \models^I \{A\} c \{\Downarrow B\}$) if $\sigma \models^I A$ implies that $\mathfrak{C}[[c]]\sigma \neq \perp$ and $\mathfrak{C}[[c]]\sigma \models^I B$.
- $\{A\} c \{\Downarrow B\}$ is called **valid in $I \in Int$** (notation: $\models^I \{A\} c \{\Downarrow B\}$) if $\sigma \models^I \{A\} c \{\Downarrow B\}$ for every $\sigma \in \Sigma$.
- $\{A\} c \{\Downarrow B\}$ is called **valid** (notation: $\models \{A\} c \{\Downarrow B\}$) if $\models^I \{A\} c \{\Downarrow B\}$ for every $I \in Int$.

Proving Total Correctness I

Goal: syntactic derivation of valid total correctness properties

Definition 11.5 (Hoare Logic for total correctness)

The **Hoare rules** for total correctness are given by

$$\begin{array}{c} \frac{}{\{A\} \text{skip} \{\Downarrow A\}} \text{ (skip)} \qquad \frac{}{\{A[x \mapsto a]\} x := a \{\Downarrow A\}} \text{ (asgn)} \\[10pt] \frac{\{A\} c_1 \{\Downarrow C\} \quad \{C\} c_2 \{\Downarrow B\}}{\{A\} c_1; c_2 \{\Downarrow B\}} \text{ (seq)} \qquad \frac{\{A \wedge b\} c_1 \{\Downarrow B\} \quad \{A \wedge \neg b\} c_2 \{\Downarrow B\}}{\{A\} \text{if } b \text{ then } c_1 \text{ else } c_2 \{\Downarrow B\}} \text{ (if)} \\[10pt] \frac{\{i \geq 0 \wedge A(i+1)\} c \{\Downarrow A(i)\}}{\{\exists i. i \geq 0 \wedge A(i)\} \text{while } b \text{ do } c \{\Downarrow A(0)\}} \text{ (while)} \\[10pt] \frac{\models (A \implies A') \quad \{A'\} c \{\Downarrow B'\} \models (B' \implies B)}{\{A\} c \{\Downarrow B\}} \text{ (cons)} \end{array}$$

where $i \in LVar$, $\models (i \geq 0 \wedge A(i+1) \implies b)$, and $\models (A(0) \implies \neg b)$.

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**.

Proving Total Correctness II

- In rule

$$\frac{\{i \geq 0 \wedge A(i+1)\} c \{\Downarrow A(i)\}}{\{\exists i. i \geq 0 \wedge A(i)\} \textbf{while } b \textbf{ do } c \{\Downarrow A(0)\}} \text{ (while)}$$

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**
- Execution terminated
 - $\implies A(0)$ holds
 - $\implies$ execution condition b false

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

- Loop to be traversed $i+1$ times ($i \geq 0$)
 - $\implies A(i+1)$ holds
 - $\implies$ execution condition b true

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

Example 11.6

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

$$A := (x > 0 \wedge x = i)$$
$$c := \text{while } \neg(x=1) \text{ do } (y:=y*x; x:=x-1)$$
$$B := (y = i!)$$

(on the board)