

Semantics and Verification of Software

Lecture 10: Axiomatic Semantics of WHILE II (Hoare Logic)

Thomas Noll

Lehrstuhl für Informatik 2
(Software Modeling and Verification)



noll@cs.rwth-aachen.de

<http://www-i2.informatik.rwth-aachen.de/i2/svsw11/>

Winter Semester 2011/12

1 Repetition: The Axiomatic Approach

2 Proof Rules for Partial Correctness

Validity of property $\{A\} \subset \{B\}$

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

if the execution of c in σ terminates in $\sigma' \in \Sigma$, then σ' satisfies B .

Definition (Syntax of assertions)

The **syntax of *Assn*** is defined by the following context-free grammar:

$$a ::= z \mid x \mid i \mid a_1 + a_2 \mid a_1 - a_2 \mid a_1 * a_2 \in LExp$$
$$A ::= t \mid a_1 = a_2 \mid a_1 > a_2 \mid \neg A \mid A_1 \wedge A_2 \mid A_1 \vee A_2 \mid \forall i. A \in Assn$$

Abbreviations:

$$A_1 \implies A_2 := \neg A_1 \vee A_2$$
$$\exists i. A := \neg (\forall i. \neg A)$$
$$a_1 \geq a_2 := a_1 > a_2 \vee a_1 = a_2$$
$$\vdots$$

The semantics now additionally depends on values of logical variables:

Definition (Semantics of $LExp$)

An **interpretation** is an element of the set

$$Int := \{I \mid I : LVar \rightarrow \mathbb{Z}\}.$$

The **value** of an arithmetic expressions with logical variables is given by the functional

$$\mathfrak{L}[\cdot] : LExp \rightarrow (Int \rightarrow (\Sigma \rightarrow \mathbb{Z}))$$

where

$$\begin{array}{ll} \mathfrak{L}[z]I\sigma := z & \mathfrak{L}[a_1 + a_2]I\sigma := \mathfrak{L}[a_1]I\sigma + \mathfrak{L}[a_2]I\sigma \\ \mathfrak{L}[x]I\sigma := \sigma(x) & \mathfrak{L}[a_1 - a_2]I\sigma := \mathfrak{L}[a_1]I\sigma - \mathfrak{L}[a_2]I\sigma \\ \mathfrak{L}[i]I\sigma := I(i) & \mathfrak{L}[a_1 * a_2]I\sigma := \mathfrak{L}[a_1]I\sigma * \mathfrak{L}[a_2]I\sigma \end{array}$$

Reminder: $A ::= t \mid a_1=a_2 \mid a_1>a_2 \mid \neg A \mid A_1 \wedge A_2 \mid A_1 \vee A_2 \mid \forall i. A \in \text{Assn}$

Definition (Semantics of assertions)

Let $A \in \text{Assn}$, $\sigma \in \Sigma_{\perp}$, and $I \in \text{Int}$. The relation “ σ satisfies A in I ” (notation: $\sigma \models^I A$) is inductively defined by:

$$\begin{aligned}\sigma &\models^I \text{true} \\ \sigma &\models^I a_1=a_2 \quad \text{if } \mathcal{L}[[a_1]]_I \sigma = \mathcal{L}[[a_2]]_I \sigma \\ \sigma &\models^I a_1>a_2 \quad \text{if } \mathcal{L}[[a_1]]_I \sigma > \mathcal{L}[[a_2]]_I \sigma \\ \sigma &\models^I \neg A \quad \text{if not } \sigma \models^I A \\ \sigma &\models^I A_1 \wedge A_2 \quad \text{if } \sigma \models^I A_1 \text{ and } \sigma \models^I A_2 \\ \sigma &\models^I A_1 \vee A_2 \quad \text{if } \sigma \models^I A_1 \text{ or } \sigma \models^I A_2 \\ \sigma &\models^I \forall i. A \quad \text{if } \sigma \models^{I[i \mapsto z]} A \text{ for every } z \in \mathbb{Z} \\ \perp &\models^I A\end{aligned}$$

Furthermore σ satisfies A ($\sigma \models A$) if $\sigma \models^I A$ for every interpretation $I \in \text{Int}$, and A is called **valid** ($\models A$) if $\sigma \models A$ for every state $\sigma \in \Sigma$.

Definition (Partial correctness properties)

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

- An expression of the form $\{A\} c \{B\}$ is called a **partial correctness property** with **precondition** A and **postcondition** B .
- Given $\sigma \in \Sigma_{\perp}$ and $I \in \text{Int}$, we let

$$\sigma \models^I \{A\} c \{B\}$$

if $\sigma \models^I A$ implies $\mathfrak{C}[c]\sigma \models^I B$
(or equivalently: $\sigma \in A^I \implies \mathfrak{C}[c]\sigma \in B^I$).

- $\{A\} c \{B\}$ is called **valid in** I (notation: $\models^I \{A\} c \{B\}$) if $\sigma \models^I \{A\} c \{B\}$ for every $\sigma \in \Sigma_{\perp}$ (or equivalently: $\mathfrak{C}[c]A^I \subseteq B^I$).
- $\{A\} c \{B\}$ is called **valid** (notation: $\models \{A\} c \{B\}$) if $\models^I \{A\} c \{B\}$ for every $I \in \text{Int}$.

1 Repetition: The Axiomatic Approach

2 Proof Rules for Partial Correctness

Hoare Logic I

Goal: syntactic derivation of valid partial correctness properties

Definition 10.1 (Hoare Logic)

The **Hoare rules** are given by

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

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

Here $A[x \mapsto a]$ denotes the syntactic replacement of every occurrence of x by a in A .

Example 10.2

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

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

(on the board)

Example 10.2

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

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

(on the board)

Structure of the proof:

$$\frac{\text{(seq)} \frac{\text{(cons)} \frac{\text{(asgn)} \frac{4}{5} \text{(asgn)} \frac{6}{6}}{2} \text{(cons)} \frac{\text{(while)}}{7}}{1} \text{(cons)} \frac{\text{(seq)} \frac{\text{(asgn)} \frac{14}{14} \text{(asgn)} \frac{15}{15}}{12}}{10}}{3} \frac{8}{9}$$

Example 10.2 (continued)

Here the respective propositions are given by:

- ① $C := (x > 0 \implies y * x! = i!)$
- ② $\{A\} y := 1; c \{B\}$
- ③ $\{A\} y := 1 \{C\}$
- ④ $\{C\} c \{B\}$
- ⑤ $\models (A \implies C[y \mapsto 1])$
- ⑥ $\{C[y \mapsto 1]\} y := 1 \{C\}$
- ⑦ $\models (C \implies C)$
- ⑧ $\models (C \implies C)$
- ⑨ $\{C\} c \{\neg(\neg(x = 1)) \wedge C\}$
- ⑩ $\models (\neg(\neg(x = 1)) \wedge C \implies B)$
- ⑪ $\{ \neg(x = 1) \wedge C \} y := y * x; x := x - 1 \{C\}$
- ⑫ $\models (\neg(x = 1) \wedge C \implies C[x \mapsto x - 1, y \mapsto y * x])$
- ⑬ $\{C[x \mapsto x - 1, y \mapsto y * x]\} y := y * x; x := x - 1 \{C\}$
- ⑭ $\models (C \implies C)$
- ⑮ $\{C[x \mapsto x - 1, y \mapsto y * x]\} y := y * x \{C[x \mapsto x - 1]\}$
- ⑯ $\{C[x \mapsto x - 1]\} x := x - 1 \{C\}$