Semantics and Verification of Software

Summer Semester 2019

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

Thomas Noll
Software Modeling and Verification Group
RWTH Aachen University

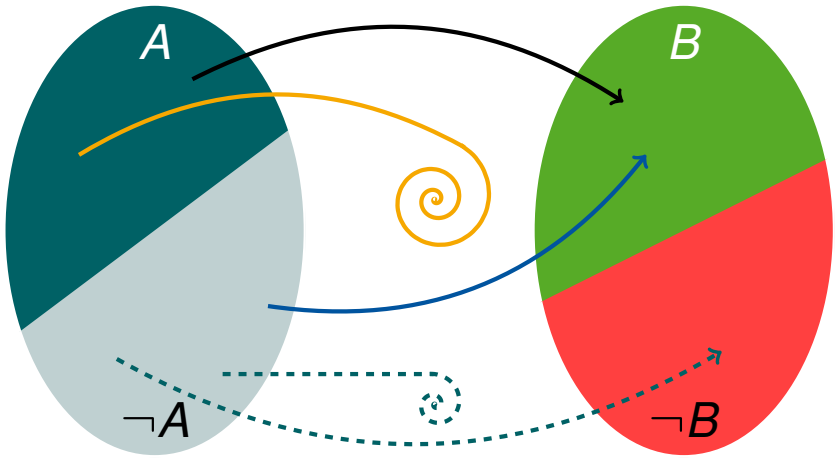
https://moves.rwth-aachen.de/teaching/ss-19/sv-sw/

Software Modeling and Verification Chair

RWTH AACHEN UNIVERSITY

Outline of Lecture 10

The Axiomatic Approach IV



(picture courtesy of C. Matheja)

Partial correctness properties

$\{A\}\ c\ \{B\}$ is valid if 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.

In particular, $\{\text{true}\}\ c\ \{\text{false}\}$ is valid for $c = \texttt{while true do skip end}$

Total correctness properties

$\{A\}\ c\ \{\Downarrow B\}$ is valid if for all states $\sigma \in \Sigma$ which satisfy A:
the execution of c in σ terminates and yields a state which satisfies B.

Recap: Axiomatic Semantics of WHILE

Syntax of the Assertion Language

Definition (Syntax of assertions)

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

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

- Thus: $\mathit{AExp} \subsetneq \mathit{LExp}$, $\mathit{BExp} \subsetneq \mathit{Assn}$
- The following (and other) abbreviations will be employed:

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

Recap: Axiomatic Semantics of WHILE

Semantics of Assertions I

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

Definition (Semantics of assertions)

Let $A \in Assn$, $\sigma \in \Sigma$, and $I \in Int$. The relation "σ satisfies A in I" (notation: $\sigma \models^I A$) is inductively defined by:

$$
\begin{array}{rll}
\sigma \models^I & \text{true} & \\
\sigma \models^I & a_1{=}a_2 & \text{if } \mathfrak{L}[\![a_1]\!] I\sigma = \mathfrak{L}[\![a_2]\!] I\sigma \\
\sigma \models^I & a_1{>}a_2 & \text{if } \mathfrak{L}[\![a_1]\!] I\sigma > \mathfrak{L}[\![a_2]\!] I\sigma \\
\sigma \models^I & \neg A & \text{if not } \sigma \models^I A \\
\sigma \models^I & A_1 \wedge A_2 & \text{if } \sigma \models^I A_1 \text{ and } \sigma \models^I A_2 \\
\sigma \models^I & A_1 \vee A_2 & \text{if } \sigma \models^I A_1 \text{ or } \sigma \models^I A_2 \\
\sigma \models^I & \forall i.A & \text{if } \sigma \models^{I[i \mapsto z]} A \text{ for every } z \in \mathbb{Z} \\
(\textit{not } \bot \models^I & A) &
\end{array}
$$

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

Semantics of Assertions II

Definition (Extension)

Let $A \in \mathit{Assn}$ and $I \in \mathit{Int}$. The extension of A with respect to I is given by

$$A^I := \{\sigma \in \Sigma \mid \sigma \models^I A\}.$$

Example

For $A := (\exists i.\, i*i = x)$ and every $I \in \mathit{Int}$,

$$A^I = \{\sigma \in \Sigma \mid \sigma(x) \in \{0, 1, 4, 9, \ldots\}\}$$

Partial Correctness Properties

Definition (Partial correctness properties)

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

- An expression of the form $\{A\}\, c\, \{B\}$ is called a partial correctness property (PCP) with precondition A and postcondition B.
- Given $\sigma \in \Sigma$ and $I \in \mathit{Int}$, we let

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

  if ($\sigma \models^I A$ and $\mathfrak{C}[\![c]\!]\sigma \neq \bot$) implies $\mathfrak{C}[\![c]\!]\sigma \models^I B$
  (or equivalently: $\sigma \in A^I \Rightarrow \mathfrak{C}[\![c]\!]\sigma \in B^I \cup \{\bot\}$).
- $\{A\}\, c\, \{B\}$ is called valid in I (notation: $\models^I \{A\}\, c\, \{B\}$) if $\sigma \models^I \{A\}\, c\, \{B\}$ for every $\sigma \in \Sigma$
  (or equivalently: $\mathfrak{C}[\![c]\!]A^I \subseteq B^I \cup \{\bot\}$).
- $\{A\}\, c\, \{B\}$ is called valid (notation: $\models \{A\}\, c\, \{B\}$) if $\models^I \{A\}\, c\, \{B\}$ for every $I \in \mathit{Int}$.

Recap: Axiomatic Semantics of WHILE

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

$$\text{(skip)}\ \frac{}{\{A\}\ \texttt{skip}\ \{A\}} \qquad \text{(asgn)}\ \frac{}{\{A[x \mapsto a]\}\ x := a\ \{A\}}$$

$$\text{(seq)}\ \frac{\{A\}\ c_1\ \{C\} \quad \{C\}\ c_2\ \{B\}}{\{A\}\ c_1 ; c_2\ \{B\}} \qquad \text{(if)}\ \frac{\{A \wedge b\}\ c_1\ \{B\} \quad \{A \wedge \neg b\}\ c_2\ \{B\}}{\{A\}\ \texttt{if}\ b\ \texttt{then}\ c_1\ \texttt{else}\ c_2\ \texttt{end}\ \{B\}}$$

$$\text{(while)}\ \frac{\{A \wedge b\}\ c\ \{A\}}{\{A\}\ \texttt{while}\ b\ \texttt{do}\ c\ \texttt{end}\ \{A \wedge \neg b\}} \qquad \text{(cons)}\ \frac{\models (A \Rightarrow A') \quad \{A'\}\ c\ \{B'\} \quad \models (B' \Rightarrow B)}{\{A\}\ c\ \{B\}}$$

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.

Outline of Lecture 10

Applying Hoare Logic I

Example 10.1 (Factorial program)

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

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

(on the board)

An Example

Applying Hoare Logic I

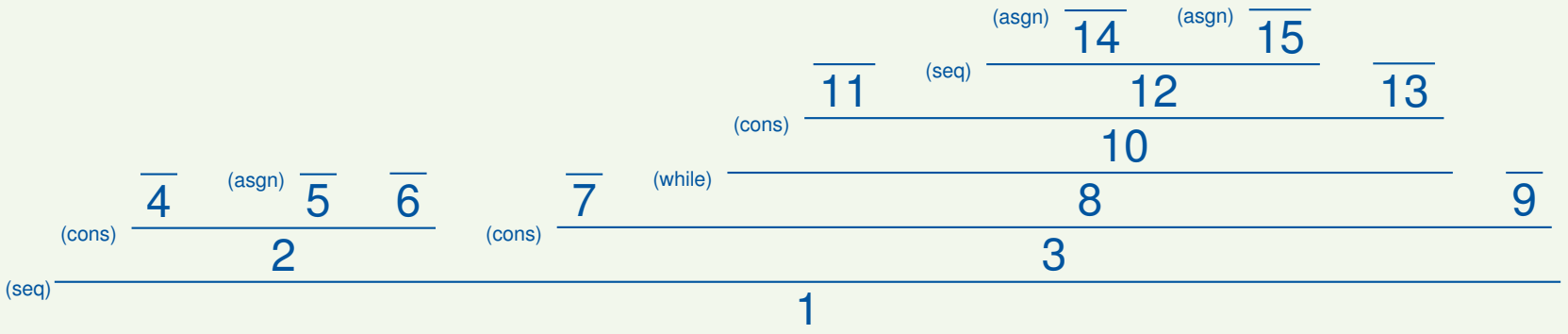
Example 10.1 (Factorial program)

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

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

(on the board)

Structure of the proof:



Applying Hoare Logic II

Example 10.1 (continued)

Here the respective propositions are given by (where $C := (\mathtt{x} > 0 \wedge \mathtt{y} * \mathtt{x}! = i!)$):

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

Outline of Lecture 10

Discovering Invariants

Goal

Prove PCP $\{A\}\ \texttt{while}\ b\ \texttt{do}\ c\ \texttt{end}\ \{B\}$ by identifying invariant C:

$$\text{(while)}\ \frac{\{C \wedge b\}\, c\, \{C\}}{\{C\}\ \texttt{while}\ b\ \texttt{do}\ c\ \texttt{end}\ \{C \wedge \neg b\}}$$

Discovering Invariants

Goal

Prove PCP $\{A\}\ \texttt{while}\ b\ \texttt{do}\ c\ \texttt{end}\ \{B\}$ by identifying invariant C:

$$\text{(while)}\ \frac{\{C \wedge b\}\, c\, \{C\}}{\{C\}\ \texttt{while}\ b\ \texttt{do}\ c\ \texttt{end}\ \{C \wedge \neg b\}}$$

This may require some ingenuity, but there are a few hints on how to do that:

- In general, there are several invariants but most of them are useless (for example, true is always an invariant)
- A suitable invariant has to be
  - weak enough to be implied by the precondition: $\models (A \Rightarrow C)$
  - strong enough to imply the postcondition: $\models (C \wedge \neg b \Rightarrow B)$
- In general, looking at the logical structure of the postcondition will help
- Often a suitable invariant is found by generalising the postcondition, replacing a constant by a variable that is changed in the body of the loop
- It can be helpful to "trace" the loop and inspect the values of the variables at every iteration

More on Invariants

What is the Invariant?

Example 10.2

1. $\{\mathtt{y} \geq 0 \wedge \mathtt{y} = i\}$ `z := 1; while ¬(y=0) do y := y-1; z := z*x end` $\{\mathtt{z} = \mathtt{x}^i\}$
   - Invariant: $C = ?$
   - Precondition: $\mathtt{y} \geq 0 \wedge \mathtt{y} = i \wedge \mathtt{z} = 1 \Rightarrow C$
   - Postcondition: $C \wedge \mathtt{y} = 0 \Rightarrow \mathtt{z} = \mathtt{x}^i$

More on Invariants

What is the Invariant?

Example 10.2

1. $\{\mathtt{y} \geq 0 \wedge \mathtt{y} = i\}$ `z := 1; while ¬(y=0) do y := y-1; z := z*x end` $\{\mathtt{z} = \mathtt{x}^i\}$
   - Invariant: $C = (\mathtt{z} = \mathtt{x}^{i-\mathtt{y}})$
   - Precondition: $\mathtt{y} \geq 0 \wedge \mathtt{y} = i \wedge \mathtt{z} = 1 \Rightarrow C$ ✓
   - Postcondition: $C \wedge \mathtt{y} = 0 \Rightarrow \mathtt{z} = \mathtt{x}^i$ ✓

More on Invariants

What is the Invariant?

Example 10.2

1. $\{\mathtt{y} \geq 0 \wedge \mathtt{y} = i\}$ `z := 1; while ¬(y=0) do y := y-1; z := z*x end` $\{\mathtt{z} = \mathtt{x}^i\}$
   - Invariant: $C = (\mathtt{z} = \mathtt{x}^{i-\mathtt{y}})$
   - Precondition: $\mathtt{y} \geq 0 \wedge \mathtt{y} = i \wedge \mathtt{z} = 1 \Rightarrow C$ ✓
   - Postcondition: $C \wedge \mathtt{y} = 0 \Rightarrow \mathtt{z} = \mathtt{x}^i$ ✓
2. $\{\mathtt{x} \geq 0 \wedge \mathtt{y} > 0 \wedge \mathtt{x} = i\}$
   `z := 0; while y<=x do x := x-y; z := z+1 end`
   $\{i = \mathtt{z} * \mathtt{y} + \mathtt{x}\}$
   - Invariant: $C = ?$
   - Precondition: $\mathtt{x} \geq 0 \wedge \mathtt{y} > 0 \wedge \mathtt{x} = i \wedge \mathtt{z} = 0 \Rightarrow C$
   - Postcondition: $C \wedge \mathtt{y} > \mathtt{x} \Rightarrow i = \mathtt{z} * \mathtt{y} + x$

More on Invariants

What is the Invariant?

Example 10.2

1. $\{\mathtt{y} \geq 0 \wedge \mathtt{y} = i\}$ `z := 1; while ¬(y=0) do y := y-1; z := z*x end` $\{\mathtt{z} = \mathtt{x}^i\}$
   - Invariant: $C = (\mathtt{z} = \mathtt{x}^{i-\mathtt{y}})$
   - Precondition: $\mathtt{y} \geq 0 \wedge \mathtt{y} = i \wedge \mathtt{z} = 1 \Rightarrow C$ ✓
   - Postcondition: $C \wedge \mathtt{y} = 0 \Rightarrow \mathtt{z} = \mathtt{x}^i$ ✓
2. $\{\mathtt{x} \geq 0 \wedge \mathtt{y} > 0 \wedge \mathtt{x} = i\}$
   `z := 0; while y<=x do x := x-y; z := z+1 end`
   $\{i = \mathtt{z} * \mathtt{y} + \mathtt{x}\}$
   - Invariant: $C = (i = \mathtt{z} * \mathtt{y} + \mathtt{x})$
   - Precondition: $\mathtt{x} \geq 0 \wedge \mathtt{y} > 0 \wedge \mathtt{x} = i \wedge \mathtt{z} = 0 \Rightarrow C$ ✓
   - Postcondition: $C \wedge \mathtt{y} > \mathtt{x} \Rightarrow i = \mathtt{z} * \mathtt{y} + x$ ✓

Outline of Lecture 10

Soundness of Hoare Logic I

Soundness: no wrong propositions can be derived, i.e., every (syntactically) provable partial correctness property is also (semantically) valid

Soundness of Hoare Logic I

Soundness: no wrong propositions can be derived, i.e., every (syntactically) provable partial correctness property is also (semantically) valid

For the corresponding proof we use:

Lemma 10.3 (Substitution lemma)

For every $A \in \mathit{Assn}$, $x \in \mathit{Var}$, $a \in \mathit{AExp}$, $\sigma \in \Sigma$, *and* $I \in \mathit{Int}$:

$$\sigma \models^I A[x \mapsto a] \iff \sigma[x \mapsto \mathfrak{A}[\![a]\!]\sigma] \models^I A.$$

Soundness of Hoare Logic I

Soundness: no wrong propositions can be derived, i.e., every (syntactically) provable partial correctness property is also (semantically) valid

For the corresponding proof we use:

Lemma 10.3 (Substitution lemma)

For every $A \in Assn$, $x \in Var$, $a \in AExp$, $\sigma \in \Sigma$, and $I \in Int$:

$$\sigma \models^I A[x \mapsto a] \iff \sigma[x \mapsto \mathfrak{A}[\![a]\!]\sigma] \models^I A.$$

Proof.

by induction over $A \in Assn$ (omitted) □

Soundness of Hoare Logic II

Theorem 10.4 (Soundness of Hoare Logic)

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

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

Soundness of Hoare Logic II

Theorem 10.4 (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 $I \in \mathit{Int}$ such that $\sigma \models^I A$, $\mathfrak{C}[\![c]\!]\sigma = \bot$ or $\mathfrak{C}[\![c]\!]\sigma \models^I B$ (on the board). □