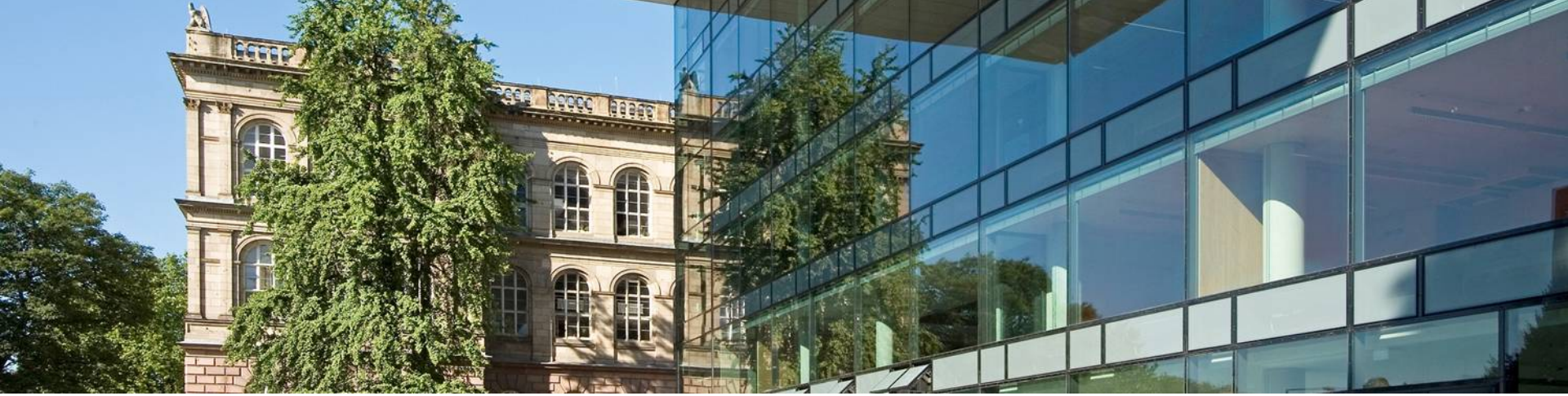


Semantics and Verification of Software

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/

Software Modeling and Verification Chair

RWTH AACHEN UNIVERSITY

Outline of Lecture 11

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

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

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

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

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.

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

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

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

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)

Incompleteness of Hoare Logic II

Corollary

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

Proof.

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

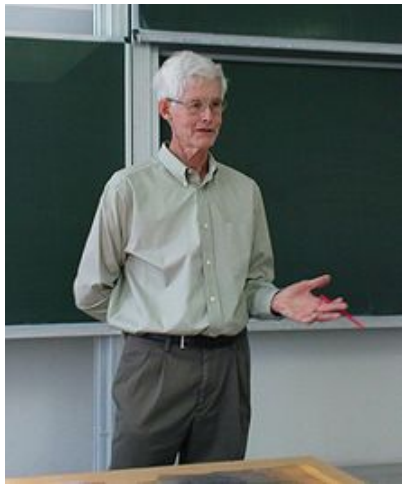
Remark: alternative proof (using computability theory):
$\{\text{true}\}\ c\ \{\text{false}\}$ is valid iff c does not terminate on any input state. But the set of all non-terminating WHILE statements is not enumerable.

Relative Completeness of Hoare Logic II

Theorem (Cook's Completeness Theorem)

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

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



Stephen A. Cook (* 1939)

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

Outline of Lecture 11

Total Correctness

- **Observation:** partial correctness properties only speak about terminating computations of a given program

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)

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

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

Semantics of Total Correctness Properties

Definition 11.1 (Semantics of total correctness properties)

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

- $\{A\}\, c\, \{\Downarrow B\}$ is called valid in $\sigma \in \Sigma$ and $I \in \mathit{Int}$ (notation: $\sigma \models^I \{A\}\, c\, \{\Downarrow B\}$) if $\sigma \models^I A$ implies that $\mathfrak{C}[\![c]\!]\sigma \neq \bot$ and $\mathfrak{C}[\![c]\!]\sigma \models^I B$.

Semantics of Total Correctness Properties

Definition 11.1 (Semantics of total correctness properties)

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

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

Semantics of Total Correctness Properties

Definition 11.1 (Semantics of total correctness properties)

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

- $\{A\}\, c\, \{\Downarrow B\}$ is called valid in $\sigma \in \Sigma$ and $I \in \mathit{Int}$ (notation: $\sigma \models^I \{A\}\, c\, \{\Downarrow B\}$) if $\sigma \models^I A$ implies that $\mathfrak{C}[\![c]\!]\sigma \neq \bot$ and $\mathfrak{C}[\![c]\!]\sigma \models^I B$.
- $\{A\}\, c\, \{\Downarrow B\}$ is called valid in $I \in \mathit{Int}$ (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 $I \in \mathit{Int}$.

Semantics of Total Correctness Properties

Definition 11.1 (Semantics of total correctness properties)

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

- $\{A\}\, c\, \{\Downarrow B\}$ is called valid in $\sigma \in \Sigma$ and $I \in \mathit{Int}$ (notation: $\sigma \models^I \{A\}\, c\, \{\Downarrow B\}$) if $\sigma \models^I A$ implies that $\mathfrak{C}[\![c]\!]\sigma \neq \bot$ and $\mathfrak{C}[\![c]\!]\sigma \models^I B$.
- $\{A\}\, c\, \{\Downarrow B\}$ is called valid in $I \in \mathit{Int}$ (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 $I \in \mathit{Int}$.

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

Corollary 11.2

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

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

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 $i \in LVar$)

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

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

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

$$\text{(cons)}\ \frac{\models (A \Rightarrow A') \quad \{A'\}\ c\ \{\Downarrow B'\} \quad \models (B' \Rightarrow B)}{\{A\}\ c\ \{\Downarrow B\}}$$

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

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

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

Proving Total Correctness II

- In rule

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

  the notation $A(i)$ indicates that assertion A parametrically depends on the value of the logical variable $i \in \mathit{LVar}$.
- Idea: i represents the remaining number of loop iterations

Proving Total Correctness II

- In rule

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

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

Proving Total Correctness II

- In rule

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

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

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

Total Correctness of Factorial Program I

Example 11.4

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

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

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

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

such that (seq) implies

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

Total Correctness of Factorial Program I

Example 11.4

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

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

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

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

such that (seq) implies

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

Now $C(j+1) = (\mathtt{x} > 0 \wedge \mathtt{y*x}! = i! \wedge \mathtt{x} = j + 2)$ and
$C(j)[\mathtt{x} \mapsto \mathtt{x-1}][\mathtt{y} \mapsto \mathtt{y*x}] = (\mathtt{x} - 1 > 0 \wedge \mathtt{y} * \mathtt{x} * (\mathtt{x} - 1)! = i! \wedge \mathtt{x} - 1 = j + 1)$
such that

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

Total Correctness of Factorial Program II

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

Hence (cons) implies

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

Total Correctness of Factorial Program II

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

Hence (cons) implies

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

Moreover we have

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

such that (while) yields

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

Total Correctness of Factorial Program II

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

Hence (cons) implies

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

Moreover we have

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

such that (while) yields

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

For the initializing assignment, (asgn) implies

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

such that (seq) allows to conclude

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

Total Correctness of Factorial Program II

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

Hence (cons) implies

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

Moreover we have

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

such that (while) yields

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

For the initializing assignment, (asgn) implies

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

such that (seq) allows to conclude

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

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

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

such that (cons) yields the desired result:

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

Outline of Lecture 11

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

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

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) □

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 □

Outline of Lecture 11

Operational and Denotational Equivalence

Definition 4.1: $\mathfrak{O}[\![.]\!] : \mathit{Cmd} \to (\Sigma \dashrightarrow \Sigma)$ given by

$$\mathfrak{O}[\![c]\!]\sigma = \sigma' \iff \langle c, \sigma \rangle \to \sigma'$$

Operational and Denotational Equivalence

Definition 4.1: $\mathfrak{O}[\![.]\!] : \mathit{Cmd} \to (\Sigma \dashrightarrow \Sigma)$ given by

$$\mathfrak{O}[\![c]\!]\sigma = \sigma' \iff \langle c, \sigma \rangle \to \sigma'$$

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

$$\mathfrak{O}[\![c_1]\!] = \mathfrak{O}[\![c_2]\!]$$

Operational and Denotational Equivalence

Definition 4.1: $\mathfrak{O}[\![.]\!] : \mathit{Cmd} \to (\Sigma \dashrightarrow \Sigma)$ given by

$$\mathfrak{O}[\![c]\!]\sigma = \sigma' \iff \langle c, \sigma \rangle \to \sigma'$$

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

$$\mathfrak{O}[\![c_1]\!] = \mathfrak{O}[\![c_2]\!]$$

Theorem 8.5: For every $c \in \mathit{Cmd}$,

$$\mathfrak{O}[\![c]\!] = \mathfrak{C}[\![c]\!]$$

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

$$\models \{A\}\, c_1\, \{B\} \quad \Longleftrightarrow \quad \models \{A\}\, c_2\, \{B\}.$$

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

$$\models \{A\}\, c_1\, \{B\} \quad \Longleftrightarrow \quad \models \{A\}\, c_2\, \{B\}.$$

(later: total correctness yields same notion of equivalence)

Axiomatic Equivalence II

Example 11.8

We show that $\texttt{while}\ b\ \texttt{do}\ c\ \texttt{end} \approx \texttt{if}\ b\ \texttt{then}\ c\texttt{;while}\ b\ \texttt{do}\ c\ \texttt{end}\ \texttt{else skip end}$
(cf. Lemma 4.3).

Axiomatic Equivalence II

Example 11.8

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

$$\models \{A\}\ \texttt{while}\ b\ \texttt{do}\ c\ \texttt{end}\ \{B\}$$

Axiomatic Equivalence II

Example 11.8

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

$$\begin{aligned} & \models \{A\}\ \texttt{while}\ b\ \texttt{do}\ c\ \texttt{end}\ \{B\} \\ \iff & \vdash \{A\}\ \texttt{while}\ b\ \texttt{do}\ c\ \texttt{end}\ \{B\} \qquad \text{(Theorem 10.2, 10.5)} \end{aligned}$$

Axiomatic Equivalence II

Example 11.8

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

$$
\begin{aligned}
& \models \{A\}\ \texttt{while}\ b\ \texttt{do}\ c\ \texttt{end}\ \{B\} \\
\iff & \vdash \{A\}\ \texttt{while}\ b\ \texttt{do}\ c\ \texttt{end}\ \{B\} \qquad \text{(Theorem 10.2, 10.5)} \\
\iff & \text{ex. } C \in \mathit{Assn} \text{ such that } \models (A \Rightarrow C), \models (C \wedge \neg b \Rightarrow B), \\
& \vdash \{C\}\ \texttt{while}\ b\ \texttt{do}\ c\ \texttt{end}\ \{C \wedge \neg b\} \qquad \text{(rule (cons))}
\end{aligned}
$$

Axiomatic Equivalence II

Example 11.8

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

$$
\begin{aligned}
& \models \{A\}\ \texttt{while}\ b\ \texttt{do}\ c\ \texttt{end}\ \{B\} \\
\iff & \vdash \{A\}\ \texttt{while}\ b\ \texttt{do}\ c\ \texttt{end}\ \{B\} \qquad \text{(Theorem 10.2, 10.5)} \\
\iff & \text{ex. } C \in \mathit{Assn} \text{ such that } \models (A \Rightarrow C), \models (C \wedge \neg b \Rightarrow B), \\
& \vdash \{C\}\ \texttt{while}\ b\ \texttt{do}\ c\ \texttt{end}\ \{C \wedge \neg b\} \qquad \text{(rule (cons))} \\
\iff & \text{ex. } C \in \mathit{Assn} \text{ such that } \models (A \Rightarrow C), \models (C \wedge \neg b \Rightarrow B), \\
& \vdash \{C \wedge b\}\ c\ \{C\} \qquad \text{(rule (while))}
\end{aligned}
$$

Axiomatic Equivalence II

Example 11.8

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

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

Axiomatic Equivalence II

Example 11.8

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

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

Axiomatic Equivalence II

Example 11.8

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

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

Axiomatic Equivalence II

Example 11.8

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

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