Semantics and Verification of Software

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

Thomas Noll

Lehrstuhl für Informatik 2
(Software Modeling and Verification)

RWTH AACHEN UNIVERSITY

noll@cs.rwth-aachen.de

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

Winter Semester 2011/12

Outline

1. Repetition: The Axiomatic Approach

2. Proof Rules for Partial Correctness

Partial Correctness Properties

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

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:

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

Semantics of *LExp*

The semantics now additionally depends on values of logical variables:

Definition (Semantics of *LExp*)

An interpretation is an element of the set

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

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

$$\mathfrak{L}[\![.]\!] : LExp \to (Int \to (\Sigma \to \mathbb{Z}))$$

where

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

Semantics of Assertions

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

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

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 $\sigma \in \Sigma$.

Partial Correctness Properties

Definition (Partial correctness properties)

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

- 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 $I \in Int$, 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 $I \in Int$.

Outline

Goal: syntactic derivation of valid partial correctness properties

Definition 10.1 (Hoare Logic)

The Hoare rules are given by

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

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

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

$$\text{(cons)}\ \frac{\models (A \implies A') \quad \{A'\}\,c\,\{B'\} \quad \models (B' \implies B)}{\{A\}\,c\,\{B\}}$$

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.

Hoare Logic II

Example 10.2

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

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

(on the board)

Structure of the proof:

$$(\text{seq})\,\dfrac{(\text{cons})\,\dfrac{\overline{4}\quad(\text{asgn})\,\overline{5}\quad\overline{6}}{2}\qquad(\text{cons})\,\dfrac{\overline{7}\quad(\text{while})\,\dfrac{(\text{cons})\,\dfrac{\overline{11}\quad(\text{seq})\,\dfrac{(\text{asgn})\,\overline{14}\quad(\text{asgn})\,\overline{15}}{12}\quad\overline{13}}{10}}{8}\quad\overline{9}}{3}}{1}$$

Example 10.2 (continued)

Here the respective propositions are given by:

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