CS 4110

Programming Languages & Logics

Lecture 11
More Hoare Logic

Overview

Last time

- Hoare Logic

Today

- “Decorated” programs
- Weakest Preconditions

Review: Hoare Logic

$$\frac{}{\vdash \{P\}\ \mathbf{skip}\ \{P\}}\ \textsc{Skip} \qquad \frac{}{\vdash \{P[a/x]\}\ x := a\ \{P\}}\ \textsc{Assign}$$

$$\frac{\vdash \{P\}\ c_1\ \{R\} \qquad \vdash \{R\}\ c_2\ \{Q\}}{\vdash \{P\}\ c_1; c_2\ \{Q\}}\ \textsc{Seq}$$

$$\frac{\vdash \{P \land b\}\ c_1\ \{Q\} \qquad \vdash \{P \land \neg b\}\ c_2\ \{Q\}}{\vdash \{P\}\ \mathbf{if}\ b\ \mathbf{then}\ c_1\ \mathbf{else}\ c_2\ \{Q\}}\ \textsc{If}$$

$$\frac{\vdash \{P \land b\}\ c\ \{P\}}{\vdash \{P\}\ \mathbf{while}\ b\ \mathbf{do}\ c\ \{P \land \neg b\}}\ \textsc{While}$$

$$\frac{\models P \Rightarrow P' \qquad \vdash \{P'\}\ c\ \{Q'\} \qquad \models Q' \Rightarrow Q}{\vdash \{P\}\ c\ \{Q\}}\ \textsc{Consequence}$$

Decorated Programs

Observation: Once we've identified loop invariants and uses of consequence, the structure of a Hoare logic is determined!

Notation: Can write proofs by "decorating" programs with:

- A precondition ($\{P\}$)
- A postcondition ($\{Q\}$)
- Invariants ($\{I\}$**while** b **do** c)
- Uses of consequence $\{R\} \Rightarrow \{S\}$
- Assertions between sequences $c_1; \{T\} c_2$

A decorated program describes a valid Hoare logic proof if the rest of the proof tree's structure is implied. (Caveats: Invariants are constrained, etc.)

Example: Decorated Factorial

$\{\mathsf{x} = n \wedge n > 0\}$

$\mathsf{y} := 1;$
$\textbf{while}\ \mathsf{x} > 0\ \textbf{do}\ \{$
$\quad \mathsf{y} := \mathsf{y} * \mathsf{x};$
$\quad \mathsf{x} := \mathsf{x} - 1$
$\}$

$\{\mathsf{y} = n!\}$

Example: Decorated Factorial

$$
\begin{array}{l}
\{\mathsf{x} = n \wedge n > 0\} \Rightarrow \\
\{1 = 1 \wedge \mathsf{x} = n \wedge n > 0\} \\
\mathsf{y} := 1; \\
\{\mathsf{y} = 1 \wedge \mathsf{x} = n \wedge n > 0\} \Rightarrow \\
\{\mathsf{y} * \mathsf{x}! = n! \wedge x \geq 0\} \\
\textbf{while } \mathsf{x} > 0 \textbf{ do } \{ \\
\qquad \{\mathsf{y} * \mathsf{x}! = n! \wedge x > 0 \wedge x \geq 0\} \Rightarrow \\
\qquad \{\mathsf{y} * \mathsf{x} * (\mathsf{x} - 1)! = n! \wedge (x - 1) \geq 0\} \\
\qquad \mathsf{y} := \mathsf{y} * \mathsf{x}; \\
\qquad \{\mathsf{y} * (\mathsf{x} - 1)! = n! \wedge (x - 1) \geq 0\} \\
\qquad \mathsf{x} := \mathsf{x} - 1 \\
\qquad \{\mathsf{y} * \mathsf{x}! = n! \wedge x \geq 0\} \\
\} \\
\{\mathsf{y} * \mathsf{x}! = n! \wedge (x \geq 0) \wedge \neg(x > 0)\} \Rightarrow \\
\{\mathsf{y} = n!\}
\end{array}
$$

Informal Rules for Decoration

Check whether a decorated program represents a valid proof using local consistency checks.

Informal Rules for Decoration

Check whether a decorated program represents a valid proof using local consistency checks.

For **skip**, the precondition and postcondition should be the same:

$$\{P\}$$
$$\textbf{skip}$$
$$\{P\}$$

Informal Rules for Decoration

For sequences, $\{P\}\ c_1\ \{R\}$ and $\{R\}\ c_2\ \{Q\}$ must be (recursively) locally consistent:

$$
\begin{array}{l}
\{P\} \\
c_1; \\
\{R\} \\
c_2 \\
\{Q\}
\end{array}
$$

Informal Rules for Decoration

Assignment should use the substitution from the rule:

$$
\begin{array}{l}
\{P[a/x]\} \\
x := a \\
\{P\}
\end{array}
$$

Informal Rules for Decoration

An **if** is locally consistent when both branches are locally consistent after adding the branch condition to each:

$\{P\}$ $x \neq 5$

if $\cancel{b}$ **then** $x = 5$

  $\{P \wedge b\}$ $x \neq 5 \wedge x = 5$

  c_1

  $\{Q\}$

else

  $\{P \wedge \neg b\}$ $x \neq 5 \wedge x \neq 5$

  c_2

  $\{Q\}$

$\{Q\}$

Informal Rules for Decoration

Decorate a **while** with the loop invariant:

$$
\begin{array}{l}
\{P\} \\
\textbf{while } b \textbf{ do} \\
\quad \{P \wedge b\} \\
\quad c \\
\quad \{P\} \\
\{P \wedge \neg b\}
\end{array}
$$

Informal Rules for Decoration

To capture the CONSEQUENCE rule, you can always write a (valid) implication:

$$\{P\} \Rightarrow \{Q\}$$

$\{P\}$

C_1 ;

$\{Q\}$

$\Rightarrow \{P'\}$

C_2

$\{Q'\}$

Example

$\{ \qquad\qquad\qquad\qquad\qquad\qquad \}$

$\textbf{while}\ (0 < \mathsf{y})\ \textbf{do}\ ($
$\quad \mathsf{x} := \mathsf{x} + 1;$
$\quad \mathsf{y} := \mathsf{y} - 1$
$)$

$\{ \qquad\qquad \}$

Example

$\{\mathsf{x} = m \ \wedge \ \mathsf{y} = n \wedge 0 \leq n\}$

while $(0 < \mathsf{y})$ **do** (

$\quad \mathsf{x} := \mathsf{x} + 1;$

$\quad \mathsf{y} := \mathsf{y} - 1$

)

$\{\mathsf{x} = m + n\}$

Example

$$
\begin{array}{l}
\{\mathrm{x} = m \ \wedge \ \mathrm{y} = n \ \wedge \ 0 \leq n\} \Rightarrow \\
\{I\} \\
\textbf{while } (0 < \mathrm{y}) \textbf{ do } (\\
\quad \{I \wedge 0 < y\} \Rightarrow \\
\quad \{I[\mathrm{y} - 1/\mathrm{y}][\mathrm{x} + 1/\mathrm{x}]\} \\
\quad \mathrm{x} := \mathrm{x} + 1; \\
\quad \{I[\mathrm{y} - 1/\mathrm{y}]\} \\
\quad \mathrm{y} := \mathrm{y} - 1 \\
\quad \{I\} \\
) \\
\{I \wedge 0 \not< \mathrm{y}\} \Rightarrow \\
\{\mathrm{x} = m + n\}
\end{array}
$$

Where I is $(\mathrm{x} = m + n - \mathrm{y}) \wedge 0 \leq \mathrm{y}$.

Example

$\{ \quad \}$

$\textbf{while}\ (\mathsf{x} \neq 0)\ \textbf{do}\ ($
$\quad \mathsf{x} := \mathsf{x} - 1$
$)$

$\{ \quad \}$

Example

$\{\textbf{true}\}$

$\textbf{while}\ (x \neq 0)\ \textbf{do}\ ($
$\quad x := x - 1$
$)$

$\{x = 0\}$

Example

$\{ \qquad\qquad\qquad\qquad \}$

$\mathsf{y} := 1$
$\textbf{while}\ (0 < \mathsf{x})\ \textbf{do}\ ($
$\quad \mathsf{x} := \mathsf{x} - 1;$
$\quad \mathsf{y} := \mathsf{y} * 2$
$)$

$\{ \qquad\qquad \}$

Example

$\{x = n \land 0 \leq n\}$

$y := 1$
while $(0 < x)$ **do** (
    $x := x - 1;$
    $y := y * 2$
)

$\{y = 2^n\}$