CS 4110

Programming Languages & Logics

Lecture 11
More Hoare Logic

22 September 2014

Announcements

- Today: Foster office hours 4-5pm
- Friday 9/26: Guest lecture by Michael Clarkson
- Wednesday 10/1: CS 50 and Gates Dedication! No lecture
- Monday 10/6: Preliminary Exam I
  - In-class, 50-minute exam
  - Scope: through Hoare Logic
  - Practice exam out Friday

Overview

Last time

- Hoare Logic

Today

- “Decorated” programs
- Soundness
- Completeness
- Weakest Preconditions

Review: Hoare Logic

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

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

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

$$\frac{\vdash \{P \land b\}\ c\ \{P\}}{\vdash \{P\}\ \textbf{while}\ b\ \textbf{do}\ c\ \{P \land \neg b\}}\ \text{While}$$

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

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 $\{R\} \Rightarrow \{S\}$
- (Optionally) assertions between commands $c_1; \{T\} c_2$

Convention: A decorated program describes a valid Hoare logic proof if one can fill in the rest of the structure from these annotations. This places constraints on the invariants, the assertions in force before and after the conditional rule etc.

Example: Decorated Programs

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

  y := 1;
  while x > 0 do {
          y := y * x;
          x := x - 1
  }

{$y = n!$}
```

Example: Decorated Programs

```
$\{\mathrm{x} = n \wedge n > 0\} \Rightarrow$
$\{1 = 1 \wedge \mathrm{x} = n \wedge n > 0\}$
$\mathrm{y} := 1;$
$\{\mathrm{y} = 1 \wedge \mathrm{x} = n \wedge n > 0\} \Rightarrow$
$\{\mathrm{y} * \mathrm{x}! = n! \wedge x \geq 0\}$
while $\mathrm{x} > 0$ do {
        $\{\mathrm{y} * \mathrm{x}! = n! \wedge x > 0 \wedge x \geq 0\} \Rightarrow$
        $\{\mathrm{y} * \mathrm{x} * (\mathrm{x} - 1)! = n! \wedge (x - 1) \geq 0\}$
        $\mathrm{y} := \mathrm{y} * \mathrm{x};$
        $\{\mathrm{y} * (\mathrm{x} - 1)! = n! \wedge (x - 1) \geq 0\}$
        $\mathrm{x} := \mathrm{x} - 1$
        $\{\mathrm{y} * \mathrm{x}! = n! \wedge x \geq 0\}$
}
$\{\mathrm{y} * \mathrm{x}! = n! \wedge (x \geq 0) \wedge \neg(x > 0)\} \Rightarrow$
$\{\mathrm{y} = n!\}$
```

Example

```
{                                   }
while (0 < y) do {
    x := x + 1;
    y := y − 1
}
{                }
```

Example

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

```
while $(0 < y)$ do {
    $x := x + 1;$
    $y := y - 1$
}
```

$\{x = m + n\}$

Example

```
{      }
while (x != 0) do {
    x := x − 1
}
{      }
```

Example

```
{true}
while (x != 0) do {
    x := x − 1
}
{x = 0}
```

Example

```
{                    }

y := 1
while (0 < x) do {
    x := x − 1
    y := y ∗ 2
}

{         }
```

Example

```
$\{x = n \land 0 \le n\}$

y := 1
while ($0 < x$) do {
    x := x − 1
    y := y ∗ 2
}

$\{y = 2^n\}$
```