Slide 1

Administrivia

- None.

Slide 2

Minute Essay From Last Lecture

- Did you recognize the last problem on the quiz? (It was one of the “not to turn in” homework problems, with predicate names changed.)
- If we want to have $\{\ R\ \}\ \ x := x * 2\ \ \{\ x < 16\ \}$, what should R be?

  Notice: As with logic proofs, one point of this is to replace maybe-difficult thinking with allegedly-more-reliable symbol manipulation.

Slide 3

Specifications and Correctness (Review)

- If we have a program P, and a specification consisting of precondition Q and postcondition R, we write

  $$\{\ Q\ \}\ \ P\ \ \{\ R\ \}$$

  to mean that if we start in a state where Q is true and run P, we end in a state where R is true. (Also, P terminates — no “infinite” loops.)

Slide 4

Sequential Composition (Review)

- For two programs P_1 and P_2
  If we have $\{\ Q\ \}\ \ P_1\ \ \{\ R_1\ \}$
  and $\{\ R_1\ \}\ \ P_2\ \ \{\ R\ \}$
  then we can derive $\{\ Q\ \}\ \ P_1; P_2\ \ \{\ R\ \}$

Slide 5

Assignment (Review)

- We can derive $\{\ R_1\ \}\ \ x := e\ \ \{\ R_2\ \}$
  where R_1 is R_2 with all occurrences of x replaced by e.

Slide 6

Strengthening Preconditions, Weakening Postconditions (Review)

- If we have $\{\ Q\ \}\ \ P\ \ \{\ R\ \}$
  then for "stronger" precondition Q_1 (i.e., $Q_1\ \rightarrow\ Q$)
  we can derive $\{\ Q_1\ \}\ \ P\ \ \{\ R\ \}$
  and for "weaker" postcondition R_1 $(i.e., R\ \rightarrow\ R_1)$
  we can derive $\{\ Q\ \}\ \ P\ \ \{\ R_1\ \}$

Slide 7

Conditionals (Review)

- If we have program S of the form

  if B **then**

  P_1

  else

  P_2

  end if

  and we have $\{ \ (Q \ \wedge \ B) \ \} \ \ P_1 \ \ \{ \ R \ \}$

  and $\{ \ (Q \ \wedge \ B') \ \} \ \ P_2 \ \ \{ \ R \ \}$

  then we can derive $\{ \ Q \ \} \ \ S \ \ \{ \ R \ \}$

Slide 8

Example

- Try an example — silly program to compute absolute value (call it S):

  if $x \geq 0$ **then**

  $y := x$

  else

  $y := -x$

  end if

  We want to show that $\{ \ true \ \} \ \ S \ \ \{ \ y := |x| \ \}$

- We can do this using rules for conditionals and assignment ...

Examples of Less Formal Use

Slide 9

- Rule for sequential composition leads to "programming with assertions" — at "interesting" points in the program, use to document/check what you know to be true at that point. Example: Program that first sorts an array, then repeatedly performs binary search. Could use assertion to document that array is sorted.
- Rule for conditionals can also be used informally: Code for "if" branch only has to work if condition is true; code for "else" branch only has to work if condition is false. Example: Function to compute root(s) of quadratic equation.

Semi-Intermezzo: A Puzzle

Slide 10

- Suppose you have a jar containing white marbles and black marbles, plus an unlimited supply of extra black marbles, and you do the following:
  1. Select two marbles.
  2. If they're the same color, discard them both and put a black marble in the jar. If they're different colors, discard the black one and put the white one back in the jar.
  3. If there are at least two marbles in the jar, repeat.
- Does this end? If it does, what if anything can you say about the marble(s) in the jar when it ends?
- (Similar ideas behind "metric" for loop termination and "invariant" for loop correctness.)

Slide 11

Program Correctness and Loops

- We'll write loops in this form

  while B **do**

  P

  end while

  After the loop terminates (assuming it does), what do we know about B? True or false?

- We also need the notion of a "loop invariant" — a predicate that, if true before we execute the loop body is true again after. More formally, Q is an invariant for the above loop if

$$\{\, Q \;\wedge\; B \,\}\;\; P \;\; \{\, Q \,\}$$

- Our rule for loops is: If we have such a Q, and P_1 is the "program" above, we can derive

Slide 12

$$\{\, Q \,\}\;\; P_1 \;\; \{\, Q \;\wedge\; B' \,\}$$

- We could prove this using induction (on the number of trips through the loop).

Trivial Example

Slide 13

- Suppose we have

  while $x > 0$ **do**

  $x := x - 1$

  end while

  with x an integer variable.

- Show that after the loop $x = 0$.

Correctness of Loops, Continued

Slide 14

- The textbook isn't very explicit about this, but strictly speaking we have something else to prove — that the loop terminates!
- Can do this with a "metric" (think "measure") — integer function of program variables that decreases every time through the loop, and when it's less than or equal to zero the loop stops.
- In the silly example, we could use what?

Slide 15

Minute Essay

- Given program P as follows:

  if $x \geq 0$ **then**

  $x := x * 2$

  else

  $x := -x$

  end if

  We can show that $\{ \; x \neq 0 \; \} \;\; P \;\; \{ \; x \neq 0 \; \}$

  by showing that two other Hoare triples are true — what are they? (No need to say why they're true, just what they are.)

- Reminder: Homework 3 was due Monday. Turn in today if you didn't drop it off Monday.