Correctness ("Hoare") Triples, v 1.1

Part 2: Sequencing, Assignment, Strengthening, and Weakening

CS 536: Science of Programming, Fall 2021

A. *Why*

- To specify a program's correctness, we need to know its precondition and postcondition (what should be true before and after executing it).
- The semantics of a verified program combines its program semantics rule with the state-oriented semantics of its specification predicates.
- To connect correctness triples in sequence, we need to weaken and strengthen conditions.

B. *Objectives*

At the end of today you should be able to

- Differentiate between different annotations for the same program.
- Determine whether two correctness triples can be joined and to give the result of joining.
- Reason "backwards" about assignment statements.
- Connect correctness triples in sequence by weakening and strengthening intermediate conditions

C. *Problems*

For all these problems, assume we're working over $\mathbb{Z}$. There may be more than one correct answer; any right answer will do.

1. Find a state σ such that $\sigma \not\models \{T\}\ y := x*x*x\ \{y > 4*x\}$. I.e., give a state in which the triple is unsatisfied — this proves that the triple is invalid.

2. Find the[1] weakest precondition p that makes $\models \{p\}\ y := x*x*x\ \{y > 4*x\}$ valid.

3. Find the strongest postcondition q such that $\{T\}\ y := x;\ \mathit{if}\ x \geq 0\ \mathit{then}\ x := x*x\ \mathit{fi}\ \{q\}$ is valid. (We want q to be satisfied by as many end states as possible.)

4. Fill in the missing code to make $\{T\}\ \mathit{if}\ ???\ \mathit{then}\ y := ???\ \mathit{else}\ y := x*x\ \mathit{fi}\ \{y > 2*x\}$ valid. (There's no unique right answer.)

---

1 Note if p is a weakest precondition, then so is anything logically equivalent to p, so "the" weakest precondition is a bit of a misnomer. The same goes for "the" strongest postcondition.

For Problems 5 and 6, use the backward assignment rule discussed in the notes.

5a. Find the most general precondition p such that $\{p\}\ x := (x+1)*y\ \{x \geq f(y)\}$ is valid.

5b. Using p, now find the most general precondition q such that $\{q\}\ y := y+2\ \{p\}$ is valid. (Note parts (a) and (b) together make $\{q\}\ y := y+2;\ x := (x+1)*y\ \{x \geq f(y)\}$ valid.)

6. Repeat Problem 5 using $\{p\}\ x := x*x\ \{x > 15\}$ and $\{q\}\ x := x+1\ \{p\}$.

Solution to Practice 9 (Hoare Triples, pt. 2)

1. For σ to not satisfy $\{p\}\ y := x*x*x\ \{y > 4*x\}$, we need $\sigma(x*x*x \le 4*x)$. This happens when $\sigma(x) = 0, 1$, or 2 or $\sigma(x) \le -2$.

2. The weakest precondition p for $\models \{p\}\ y := x*x*x\ \{y > 4*x\}$ is $x*x*x > 4*x$.

3. The strongest postcondition q for $\{T\}\ y := x;$ *if* $x \ge 0$ *then* $x := x*x$ *fi* $\{q\}$ valid is $q \equiv y \ge 0 \rightarrow x = y^2$

4. If x = 0, 1, or 2, then $x*x \le 2*x$, so in that case we need to set y to something > $2*x$; the code is $\{T\}$ *if* $0 \le x \wedge x \le 2$ *then* $y := 2*x+1$ *else* $y := x*x$ *fi* $\{y > 2*x\}$.

5a. The weakest p that makes $\{p\}\ x := (x+1)*y\ \{x \ge f(y)\}$ valid is $(x+1)*y \ge f(y)$.
5b. The weakest q that makes $\{q\}\ y := y+2\ \{p\}$ valid is $(x+1)*(y+2) \ge f(y+2)$.

6a. To make $\{p\}\ x := x*x\ \{x > 15\}$ valid, the weakest p is $x*x > 15$.
6b. To make $\{q\}\ x := x+1\ \{p\}$ valid, the weakest q is $(x+1)*(x+1) > 15$.

