

Hoare Logic II

COMP2600 — Formal Methods for Software Engineering

Ranald Clouston

Australian National University
Semester 2, 2011

The Assignment Axiom (Rule 3/6)

Assignments *change the state* so we expect Hoare triples for assignments to reflect that change.

Suppose $Q(x)$ is a predicate involving a variable x , and that $Q(e)$ indicates the same formula with all occurrences of x replaced by the expression e .

The assignment axiom of Hoare Logic:

$$\{Q(e)\} x := e \{Q(x)\}$$

Intuition:

If we want x to have some property Q *after* the assignment, then that property must hold for the value (e) being assigned to x *before* the assignment is executed.

The Axiom is Not Backwards

One might think that the assignment axiom

$$\{Q(e)\} x := e \{Q(x)\}$$

is somehow backwards, and we should go

$$\{Q(x)\} x := e \{Q(e)\}$$

But this is **wrong**: if we tried to apply this 'axiom' to the precondition $x = 0$ and code fragment $x := 1$ we'd get

$$\{x = 0\} x := 1 \{1 = 0\} !$$

So we're right to work 'backwards' from postcondition to precondition.

This is the usual practice in Hoare Logic.

Example

Consider the statement $x := 2$ and suppose that the desired postcondition is $(y = x)$.

Our precondition is found by copying the postcondition $y = x$ over with one exception: replace every instance of x with the expression 2.

Formally:

$$\{y = 2\} x := 2 \{y = x\}$$

is an instance of the assignment axiom.

Another Example

How might we try to prove

$$\{y > 0\} x := y + 3 \{x > 3\} ?$$

Start with the postcondition $x > 3$ and apply the axiom:

$$\{y + 3 > 3\} x := y + 3 \{x > 3\}$$

Then simplify $y + 3 > 3$ to $y > 0$ to get our result.

An Example with Precondition Strengthening

Let's try to prove

$$\{y = 3\} x := y \{x > 0\}$$

Applying the assignment axiom:

$$\{y > 0\} x := y \{x > 0\}$$

It's a rather obvious fact about arithmetic that

$$(y = 3) \rightarrow (y > 0)$$

So we can **strengthen the precondition** to finish our proof.

Intuition Once Again:

The assignment axiom of Hoare Logic is:

$$\{Q(e)\} x := e \{Q(x)\}$$

Why is it so?

- Let v be the value assigned — i.e. v is the value of expression e in the initial state.
- If $Q(e)$ is true initially, then so is $Q(v)$.
- Since the variable x has value v after the assignment (*and nothing else is changed in the state*), $Q(x)$ must be true after that assignment.

The Assignment Axiom is Optimal

The Hoare triple in the assignment axiom **is as strong as possible**.

$$\{Q(e)\} x := e \{Q(x)\}$$

That is, $Q(e)$ is the weakest precondition which guarantees the postcondition.

Why?

- Suppose $Q(x)$ is true after the assignment.
- If v is the value assigned, $Q(v)$ is true after the assignment.
- Since it is only the value of x that is changed, and the predicate $Q(v)$ does not involve x , $Q(v)$ must also be true before the assignment.
- Since v was the value of e before the assignment, $Q(e)$ is true initially.

Proof rule for Sequencing (Rule 4/6)

Imperative programs consist of a sequence of statements, affecting the state one after the other.

$$\frac{\{P\} S_1 \{Q\} \quad \{Q\} S_2 \{R\}}{\{P\} S_1; S_2 \{R\}}$$

Example Instance:

$$\frac{\{x > 2\} \ x := x + 1 \ \{x > 3\} \quad \{x > 3\} \ x := x + 2 \ \{x > 5\}}{\{x > 2\} \ x := x + 1; \ x := x + 2 \ \{x > 5\}}$$

Validity of Rule for Sequences

Suppose the hypotheses $\{P\}S_1\{Q\}$ and $\{Q\}S_2\{R\}$ are true and let σ_0 be an arbitrary state that satisfies P .

From the rule premises, we know that:

1. Executing S_1 on σ_0 must produce a state σ_1 that satisfies Q .
2. Executing S_2 on σ_1 must produce a state σ_2 that satisfies R .

But $S_1; S_2$ just means execute S_1 and then execute S_2 .

So, when $S_1; S_2$ executes on σ_0 , the resulting state will be state σ_2 which we know must satisfy R .

What about non-termination?

Another Example

What is a suitable precondition for the code

```
z := x;
x := y;
y := z
```

to establish the postcondition $\{x = 5 \wedge y = 3\}$?

1. $\{x = 5 \wedge z = 3\} \ y := z \ \{x = 5 \wedge y = 3\}$ (Assignment)
2. $\{y = 5 \wedge z = 3\} \ x := y \ \{x = 5 \wedge z = 3\}$ (Assignment)
3. $\{y = 5 \wedge z = 3\} \ x := y; \ y := z \ \{x = 5 \wedge y = 3\}$ (Sequencing 1,2)
4. $\{y = 5 \wedge x = 3\} \ z := x \ \{y = 5 \wedge z = 3\}$ (Assignment)
5. $\{y = 5 \wedge x = 3\} \ z := x; \ x := y; \ y := z \ \{x = 5 \wedge y = 3\}$ (Seq. 3,4)

Proof Rule for Conditionals (Rule 5/6)

$$\frac{\{P \wedge b\} S_1 \{Q\} \quad \{P \wedge \neg b\} S_2 \{Q\}}{\{P\} \text{ if } b \text{ then } S_1 \text{ else } S_2 \{Q\}}$$

- When a conditional is executed, either S_1 or S_2 is executed.
- Therefore, if the *conditional* is to establish Q , **both** S_1 and S_2 must establish Q .
- Similarly, if the precondition for the *conditional* is P , then it must also be a precondition for the two branches S_1 and S_2 .
- The choice between S_1 and S_2 depends on evaluating b *in the initial state*, so we can also assume b to be a precondition for S_1 and $\neg b$ to be a precondition for S_2 .

Example of Conditional Rule

Suppose we wish to prove:

$$\{x > 2\} \text{ if } x > 2 \text{ then } y := 1 \text{ else } y := -1 \{y > 0\}$$

The proof rule for conditionals suggests we prove:

$$\{x > 2 \wedge x > 2\} y := 1 \{y > 0\}$$

$$\{x > 2 \wedge \neg(x > 2)\} y := -1 \{y > 0\}$$

Simplifying the preconditions:

(1) $\{x > 2\} y := 1 \{y > 0\}$

(2) $\{False\} y := -1 \{y > 0\}$

Example ctd

- For subgoal (1) the assignment axiom tells us that

$$\{1 > 0\} y := 1 \{y > 0\}$$

Which simplifies to

$$\{True\} y := 1 \{y > 0\}$$

Now since $(x > 2) \rightarrow True$ we can strengthen the precondition to $(x > 2)$ as required.

- For subgoal (2) the assignment axiom tells us that

$$\{-1 > 0\} y := -1 \{y > 0\}$$

and $(-1 > 0)$ is obviously equal to *False*, as required.

QED.

A Simpler Proof Rule for Conditionals?

Consider this rule:

$$\frac{\{P\} S_1 \{Q\} \quad \{P\} S_2 \{Q\}}{\{P\} \text{ if } b \text{ then } S_1 \text{ else } S_2 \{Q\}}$$

- Is it valid?
- It looks simpler, so why isn't it the standard rule?
- Because *subgoals* may not be true, even if the conclusion is true.
- That is, the *precondition* P may be *too weak* to establish Q as a postcondition of S_1 and S_2 .

The Problem Illustrated:

We've just proved

$$\{x > 2\} \text{ if } x > 2 \text{ then } y := 1 \text{ else } y := -1 \{y > 0\}$$

The subgoals suggested by the simpler rule are:

$$\{x > 2\} y := 1 \{y > 0\}$$

$$\{x > 2\} y := -1 \{y > 0\}$$

The first statement is true and provable in Hoare Logic, but the second statement most definitely is not!

So our 'simpler' rule is *not* what we want.