CSC2125 - Axiomatic Semantics assignment

Jocelyn Simmonds

January 8, 2007

Questions

1. Prove the following Hoare triple:

   $\{x > a \wedge y > b\}$ $t := x;$ $x := x + y;$ $y := t;$ $\{x = a + b \wedge y = a\}$

2. Determine the weakest precondition of the following piece of code:

```
s1: if (x >= 10) then
s2:    x = x/2;
s3: else
s4:    x = x + 5;
```

   Postcondition: $\{x \geq 5 \wedge x \leq 15\}$

3. The following is Euclid's algorithm for computing the greatest common divisor of two positive integers, m and n.

```
s1: x := m;
s2: y := n;
s3: while x != y do
s4:    if x > y then
s5:       x := x - y;
s6:    else
s7:       y := y - x;
s8: g := x
```

   Prove that the following Hoare triple holds:

   Precondition: $\{m > 0 \wedge n > 0\}$
   Postcondition: $\{g = gcd(m, n)\}$

Solutions

1. Application of assignment and sequential composition rules:

$\{x > a \wedge y > b\}$	(Pre)
$t := x;$	(s1)
$x := x + y;$	(s2)
$y := t;$	(s3)
$\{x = a + b \wedge y = a\}$	(Post)

(s3): Post$\{y \leftarrow t\} \;\equiv\; A \;\equiv\; \{x = a + b \wedge t = a\}$
(s2): $A\{x \leftarrow x + y\} \;\equiv\; B \;\equiv\; \{x + y = a + b \wedge t = a\}$
(s1): $B\{t \leftarrow x\} \;\equiv\; C \;\equiv\; \{x + y = a + b \wedge x = a\}$
It is easy to see that $C \;\equiv\;$ Pre

2. Using the conditional rule: $wp(if\ b\ then\ c1\ else\ c2, P) \;=\; (b \wedge wp(c1, P)) \vee (\neg b \wedge wp(c2, P))$
   $wp(s1 - s4, (x \geq \wedge x \leq 15))$
   $\equiv ((x \geq 10 \wedge wp(x := x/2, \text{Post})) \vee (x < 10 \wedge wp(x := x + 5, \text{Post})))$

   $wp(x := x/2, \text{Post}) \equiv x/2 \geq 5 \wedge x/2 \leq 15 \equiv x \geq 10 \wedge x \leq 30$

   $wp(x := x + 5, \text{Post}) \equiv x + 5 \geq 5 \wedge x + 5 \leq 15 \equiv x \geq 0 \wedge x \leq 15$

   So, $wp(s1 - s4, \text{Post}) \equiv (x \geq 10 \wedge x \geq 10 \wedge x \leq 30) \vee (x < 10 \wedge x \geq 0 \wedge x \leq 15)$
   $\equiv (x \geq 10 \wedge x \leq 30) \vee (x < 10 \wedge x \geq 0)$
   $\equiv (x \geq 0 \wedge x \leq 30)$

3. Invariant: $\{x > 0 \wedge y > 0 \wedge gcd(x, y) = gcd(m, n)\}$
   Proof steps:

   (a) Test that invariant is satisfied before the loop begins:
   $\{\text{Pre}\}\ x := m;\ y := n\ \{\text{INV}\}$

   $\{\text{Pre}\}\ x := m\ \{Q\},\ \{Q\}\ y := n\ \{\text{INV}\}$ (Seq. Comp.)

   $\text{INV}\{y \leftarrow n\} \;\equiv\; Q \;\equiv\; \{x > 0 \wedge n > 0 \wedge gcd(x, n) = gcd(m, n)\}$ (Assign.)

   $Q\{x \leftarrow m\} \;\equiv\; \{m > 0 \wedge n > 0 \wedge gcd(m, n) = gcd(m, n)\}$ (Assign.)

   $\{m > 0 \wedge n > 0 \wedge gcd(m, n) = gcd(m, n)\} \;\Rightarrow\; \{m > 0 \wedge n > 0\}$ (Math)

   (b) Now we want to prove that the invariant holds on the loop:
   $\{\text{INV} \wedge x \neq y\}\ s4 - s7\ \{\text{INV}\}$
   But s4-s7 is a conditional, so we need to test both branches.

   - True branch:
     $\{\text{INV} \wedge x \neq y \wedge x > y\}\ x := x - y\ \{INV\}$

     $\text{INV}\{x \leftarrow x - y\} \;\equiv\; \{x - y > 0 \wedge y > 0 \wedge gcd(x - y, y) = gcd(m, n)\}$ (Assign.)

     but $x, y > 0 \wedge x \neq y \wedge gcd(x, y) = gcd(m, n) \wedge x > y \;\Rightarrow\; x - y, y > 0 \wedge gcd(x - y, y) = gcd(m, n)$

     so, by consequence: $\{x, y > 0 \wedge x \neq y \wedge gcd(x, y) = gcd(n, m) \wedge x > y\}\ x := x - y\ \{\text{INV}\}$ (A)
   - False branch:
     $\{\text{INV} \wedge x \neq y \wedge !(x > y)\}\ y := y - x\ \{INV\}$

     $\text{INV}\{y \leftarrow y - x\} \;\equiv\; \{x > 0 \wedge y - x > 0 \wedge gcd(x, y - x) = gcd(m, n)\}$ (Assign.)

but $x, y > 0 \wedge x \neq y \wedge gcd(x, y) = gcd(m, n) \wedge !(x > y) \;\Rightarrow\; x, y - x > 0 \wedge gcd(x, y - x) = gcd(m, n)$

so, by consequence: $\{x, y > 0 \wedge x \neq y \wedge gcd(x, y) = gcd(n, m) \wedge !(x > y)\}$ $y := y - x$ {INV} (B)

By (A), (B) and the rule of conditional:
{INV $\wedge\, x \neq y$} if $x > y$ then $x := x - y$ else $y := y - x${INV}

(c) Proving that INV is strong enough to prove the postcondition:
$\{R\}$ $g := x$ {Post}

Post$\{g \leftarrow x\} \;\equiv\; R \;\equiv\; \{x = gcd(m, n)\}$

After the loop, {INV $\wedge\, x = y$} holds.

By def of gcd: $\{x, y > 0 \wedge x = y \Rightarrow gcd(x, y) = x\}$

So, INV $\wedge\, x = y \;\Rightarrow\; R$

so, by consequence: {INV $\wedge\, x = y$} $g := x$ $\{g = gcd(m, n)\}$

(d) Joining the previous arguments, we have that {Pre} Program {Post}