Assignment 1 CSSE3100/7100 Reasoning about Programs

Personal feedback ***Jacob Freeman***

Proof of GCD1

Pre and postcondition (1 mark): *Correct*

Your mark: *1*

Termination metric (0.5 marks): *Correct*

Your mark: *0.5*

Weakest precondition proof (4 marks) :

Cannot invoke gcd rule 4 without a >= 0 in third branch. a % b is only defined when b != 0, so you need b > 0 as support (no marks deducted)

Your mark: *3.5*

A sample solution is provided below. Each red asterisk (*) represents 0.5 marks. Additionally, 0.5 marks are taken off for each wrong simplification, or unjustified non-trivial simplification.

```
method GCD1(a: int, b: int) returns (r: int)
        requires a > 0 && b > 0 *
        ensures r == gcd(a, b) *
        decreases b *                                          // note that a % b < b
{
        { b > 0 && a > 0 && (a % b == 0 ==> b == gcd(a, b)) &&
          (a % b != 0 ==> a % b > 0 && gcd(b, a % b) == gcd(a, b)) }      strengthening
        { (a < b ==> b > 0 && a > 0) &&
         (a >= b ==> (a % b == 0 ==> b == gcd(a, b)) &&
                  (a % b != 0 ==> b > 0 && a % b > 0 && gcd(b, a % b) == gcd(a, b)) } *
        if a < b {
                { b > 0 && a > 0 } *                                       rule (iii)
                { b > 0 && a > 0 && gcd(b, a) == gcd(a, b) }           one-point rule
                { b > 0 && a > 0 && forall r' :: r' == gcd(b, a) ==> r' == gcd(a, b) }   *
                r := GCD1(b, a);
                { r == gcd(a, b) }
        } else
        { (a % b == 0 ==> b == gcd(a, b)) &&
          (a % b != 0 ==> b > 0 && a % b > 0 && gcd(b, a % b) == gcd(a, b)) } *
        if (a % b == 0) {
```

```
            { b == gcd(a, b) } *
            r := b;
            { r == gcd(a, b) }
        } else {
            { b > 0 && a % b > 0 && gcd(b, a % b) == gcd(a, b) }          one-point rule
            { b > 0 && a % b > 0 && forall r' :: r' == gcd(b, a % b) ==> r' == gcd(a, b) } *
            r := GCD1(b, a % b);
            { r == gcd(a, b) }
        }
        { r == gcd(a, b) }
}
```

Since a > 0 and b > 0, a % b == 0 implies b == gcd(a,b) *, and a % b != 0 implies both a % b > 0 and gcd(b, a % b) == gcd(a, b) by rule (iv) *, the stated precondition of the method a > 0 && b > 0 implies the calculated precondition b > 0 && a > 0 && (a % b == 0 ==> b == gcd(a,b)) && (a % b != 0 ==> a % b > 0 && gcd(b, a % b) == gcd(a, b)). Therefore, Andy is correct.

Proof of GCD2

Pre and postcondition (1 mark): *Correct*

Your mark: *1*

Termination metric (0.5 marks): *Correct*

Your mark: *0.5*

Weakest precondition proof (3 marks) :

Cannot invoke gcd rule 4 without a >= 0 in second branch. Cannot invoke rule 1 without a >= 0 after if statement.

Your mark: *2*

A sample solution is provided below. Each red asterisk (*) represents 0.5 marks. Additionally, 0.5 marks are taken off for each wrong simplification, or unjustified non-trivial simplification.

```
method GCD2(a: int, b: int) returns (r: int)
        requires a >= 0 && b >= 0 *
        ensures r == gcd(a, b) *
        decreases b *
{
        { (b == 0 ==> a == gcd(a, b)) &&
```

```
    (b != 0 ==> b >= 0 && a % b >= 0 && gcd(b, a % b) == gcd(a, b)) } *
  if b == 0 {
      { a == gcd(a, b) } *
      r := a;
      { r == gcd(a, b) }
  } else {
      { b >= 0 && a % b >= 0 && gcd(b, a % b) == gcd(a, b) }        one-point rule
      { b >= 0 && a % b >= 0 && forall r' :: r' == gcd(b, a % b) ==> r' == gcd(a, b) } *
      r := GCD2(b, a % b);
      { r == gcd(a, b) }
  }
  { r == gcd(a, b) }
}
```

Since a >= 0 and b >= 0 together with b == 0 implies a == gcd(a,b) by rule (i) *, and together with b != 0 implies a % b >= 0 *, and also implies gcd(b, a % b) == gcd(a, b) by rule (iv) *, the stated precondition of the method a >= 0 && b >= 0 implies the calculated precondition (b == 0 ==> a = gcd(a,b)) && (b! = 0 ==> b >= 0 && a % b >= 0 && gcd(b, a % b) == gcd(a, b)). Therefore, Candy is also correct.

Total mark: | *8.5* |