Correctness Statements Proved in Class

Stefan D Bruda

11 March 2026

1. In-place swap:

```
ASSERT(x==x0 && y==y0)
x = x - y;
y = x + y;
x = y - x;
ASSERT(x==y0 && y==x0)
```

   Proof:

```
ASSERT(x==x0 && y==y0)
ASSERT(y==y0 && x==x0)       // 5 - math
ASSERT(y==y0 && x-y+y==x0)   // 4 - assign
x = x - y;
ASSERT(y==y0 && x+y==x0)     // 3 - math
ASSERT(x+y-x==y0 && x+y==x0) // 2 - assign
y = x + y;
ASSERT(y-x==y0 && y==x0)     // 1 - assign
x = y - x;
ASSERT(x==y0 && y==x0)
```

2. Some piece of code with identical pre- and post-conditions:

```
ASSERT(i >= 0 && y == power(x,i))
y = y * x;
i++;  // i = i+1;
ASSERT(i >= 0 && y == power(x,i))
```

   Proof:

```
ASSERT(i >= 0 && y == power(x,i))        // 5 strengthen
ASSERT(i+1 >= 0 && y == power(x,i))      // 4 - math
FACT(power(x,i+1) == x * power(x,i))
ASSERT(i+1 >= 0 && y*x == x*power(x,i))  // 3 - math
ASSERT(i+1 >= 0 && y*x == power(x,i+1))  // 2 - assign
y = y * x;
```

```
ASSERT(i+1 >= 0 && y == power(x,i+1))    // 1 - assign
i++;  // i = i+1;
ASSERT(i >= 0 && y == power(x,i))
```

3. Typical code for finding the maximum of two values:

```
ASSERT(true)
if (x>y)
    m = x;
else
    m = y;
ASSERT(m >= x && m >= y && (m == x || m == y))
```

Proof:

```
ASSERT(true)                                          // 13 - if, qed
if (x>y)
    ASSERT(true && x > y)                             // 9 - strengthen
    ASSERT(true && x >= y)                            // 8 - strengthen
    ASSERT(x >= y)                                    // 6 - math
    ASSERT(x >= x && x >= y && (x == x || x == y))    // 5 - assign
    m = x;
    ASSERT(m >= x && m >= y && (m == x || m == y))    // 1 - if
else
    ASSERT(true && ! x>y )                           // 12 - math
    ASSERT(true && ! y<x )                           // 11 - math
    ASSERT(true && ! !(y >= x))                      // 10 - math
    ASSERT(true && y >= x)                           // 7 - strengthen
    ASSERT(y >= x)                                   // 4 - math
    ASSERT(y >= x && true && (y == x || true))       // 3 - math
    ASSERT(y >= x && y >= y && (y == x || y == y))   // 2 - assign
    m = y;
    ASSERT(m >= x && m >= y && (m == x || m == y))   // 1 - if
ASSERT(m >= x && m >= y && (m == x || m == y))
```

4. Alternative code for finding the maximum of two variables:

```
ASSERT(true)
m = x;
if (y > x)
    m = y;
ASSERT(m >= x && m >= y && (m == x || m == y))
```

Proof:

```
ASSERT(true)                                                                  // 14
ASSERT(x == x)                                                                // 13
m = x;
ASSERT(m == x)                                                                // 12
if (y > x)
    ASSERT(m == x && y > x)                                                   // 11
    ASSERT(y > x)                                                             // 5
    ASSERT(y >= x && y > x)                                                   // 4
    ASSERT(y >= x)                                                            // 3
    ASSERT(y >= x && y >= y && (y == x || y == y))                            // 2
    m = y;
    ASSERT(m >= x && m >= y && (m == x || m == y))                            // 1
else  // added to comply with the tableau for conditionals
    ASSERT(m == x && ! (y > x))                                               // 10
    ASSERT(m == x && x >= y && ! (y > x))                                     // 9
    ASSERT(m == x && x >= x && x >= y && (x == x || x == y) && ! (y > x))     // 8
    ASSERT(m == x && m >= x && m >= y && (m == x || m == y) && ! (y > x))     // 7
    ASSERT(m >= x && m >= y && (m == x || m == y) && ! (y > x))               // 6
    ASSERT(m >= x && m >= y && (m == x || m == y))                            // 1
ASSERT(m >= x && m >= y && (m == x || m == y))
```