HW2

Xinhao Luo

Sunday 16$^{\text{th}}$ February, 2020

1 Problem 1

a) $m^n = x^y \times result$ && $y \geq 0$

b) Initial Condition: $result = 1$, $x = m$, $y = n$, $n \geq 0$

$$\begin{aligned} m^n &= x^y \times result \\ m^n &= m^n \times 1 \\ m^n &= m^n \end{aligned} \tag{1}$$

The base case is true

c) Assume LI is true in loop k, identify if it is true for loop k+1
There are two kinds of condition

When y is even
- $x_{new} = x_{old} \times x_{old}$
- $y_{new} = y_{old}/2$
-

$$\begin{aligned} {x_{new}}^{y_{new}} \times result &= m^n \\ {x_{old}^2}^{y_{old}/2} \times result &= m^n \\ {x_{old}}^{y_{old}} \times result &= m^n \end{aligned} \tag{2}$$

When y is not even
- $result_{new} = result_{old} \times x$
- $y_{new} = y_{old} - 1$
-

$$\begin{aligned} x^{y_{new}} \times result_{new} &= m^n \\ x^{y_{old}-1} \times result_{old} \times x &= m^n \\ x^{y_{old}} \times result_{old} &= m^n \end{aligned} \tag{3}$$

The LI holds true for both conditions.

d) When the loop terminate, we know that: $y == 0$

$$\begin{aligned} !(y! = 0)\&\&(m^n = x^y \times result)\&\&(y \geq 0)) \\ y == 0\&\&(m^n = x^y \times result)\&\&(y \geq 0)) \\ y == 0\&\&(m^n = x^y \times result) \\ result = m^n \end{aligned} \tag{4}$$

Here we have the same condition as the post condition listed.

e) Here we define function $D = y$

- D decrease each step. If D is even, it decrease by D = D/2, else D = D - 1
- D will reach 0 at last When $D == 0$, loop exited. At the same time, $y == 0$

2 Problem 2

1. ```
   i = 0
   j = N - 1
   while (true) {
       while (arr[i] is Red && i < N) {
           i++
       }
       while (arr[j] is Blue && j > -1) {
           j--
       }

       if (i > j) break
       swap(arr, i, j)
   }
   ```

2. arr[0]...arr[k - 1] are red && arr[k]...arr[N - 1] are blue && $0 \leq j < i \leq (N-1)$

3. 1) element(s) before arr[i] are red && element(s) after arr[j] are blue

   2) element(s) before arr[i] are red

   3) element(s) after arr[j] are blue
   ```

3 Problem 3

1.

```
function Factorial(n: int): int
  requires n >= 0
{
  if n == 0 then 1 else n * Factorial(n-1)
}

method LoopyFactorial(n: int) returns (u: int)
  requires n >= 0
  ensures u == Factorial(n)
  {
    u := 1;
    var r := 0;
    while (r < n)
      invariant 0 <= r <= n
      invariant u == Factorial(r)
    {
      var v := u;
      var s := 1;
      while (s <= r)
        invariant 1 <= s <= r + 1
        invariant u == v * s
      {
        u := u + v;
        s := s + 1;
      }
      r := r + 1;
      assert u == v * r;
    }
  }
```

2. For the inner loop, we have:

Base Case We have: $u = v$; $s = 1$.
From outer loop, we have: $0 <= r <= n$.
So we have:

. $1 <= s <= 1$, while $s = 1$
. $u == v * 1$, while $u == v$

They hold true when entering the loop

Induction Step Assume LI holds true for loop k, try $k + 1$. Here we have:

- $u_{new} = u_{old} + v$
- $s_{new} = s_{old} + 1$

– $s <= r$, or the loop will not exist

We can get:

(a)

$$
\begin{aligned}
u &== v * s \\
u_{old} + v &== v * (s_{old} + 1) \\
u_{new} &== v * s_{new}
\end{aligned}
\tag{5}
$$

(b)

$$
\begin{aligned}
1 \leq s &\leq r + 1 \\
1 \leq s_{old} + 1 &\leq r + 1 \\
s_{new} &\leq r
\end{aligned}
\tag{6}
$$

Proved that invariant holds true

3. For the outer loop, we have:

Base Case $u = 1$, $r = 0$.
$Factorial(0) == 1$, $0 \leq 0$
Invariant holds true for base case

Induction Step Assume LI holds true for loop k, try $k + 1$
Assume the inner loop invariant: $u == v * r$

– $u_{new} = u_{old} \times r_{old}$
– $r_{new} = r_{old} + 1$
– $r < n$, or loop will not exist

$$
\begin{aligned}
u &== Factorial(r) \\
u_{old} \times r_{old} &== Factorial(r_{old} + 1) \\
u_{old} \times r_{old} &== (r_{old} + 1) * Factorial(r_{old}) \\
u_{new} &== r_{new} * Factorial(r_{old}) \\
u_{new} &== Factorial(r_{new})
\end{aligned}
\tag{7}
$$

Since $r < n$; $r_{new} = r_{old} + 1$, $r_{new} \leq n$ hold true Invariant hold true