Correctness Statements Proved in Class

Stefan D Bruda

13 March 2026

1. Loop that computes x^n:

```
ASSERT(n >= 0)
i = 0;
y = 1;
while (i != n) {
    y = y * x;
    i++;
}
ASSERT(y == power(x,n))
```

Proof:

```
ASSERT(n >= 0)
ASSERT(0 <= n)
ASSERT(1 == 1 && 0 <= n)
ASSERT(1 == 1 && 0 <= 0 <= n)
ASSERT(1 == power(x,0) && 0 <= 0 <= n)
i = 0;
ASSERT(1 == power(x,i) && 0 <= i <= n)
y = 1;
ASSERT(y == power(x,i) && 0 <= i <= n)
while (i != n) {
    ASSERT(y == power(x,i) && 0 <= i <= n && i != n)
    FACT(0 <= i+1 <= n && i != n <=> 0 <= i <= n && i != n)
    ASSERT(y == power(x,i) && 0 <= i+1 <= n && i != n)
    ASSERT(y == power(x,i) && 0 <= i+1 <= n)
    ASSERT(y * x == power(x,i) * x && 0 <= i+1 <= n)
    FACT(power(x,i+1) == power(x,i) * x)
    ASSERT(y * x == power(x,i+1) && 0 <= i+1 <= n)
    y = y * x;
    ASSERT(y == power(x,i+1) && 0 <= i+1 <= n)
    i++; // i = i + 1;
    ASSERT(y == power(x,i) && 0 <= i <= n)
}
```

```
ASSERT(y == power(x,i) && i == n && 0 <= i <= n)
ASSERT(y == power(x,n) && i == n && 0 <= i <= n)
ASSERT(y == power(x,n))
```

Helpful for determining the variant: `i==n <=> n-i == 0`, which suggests the variant `n-i`. Also recall that we started without the range for `i`, which was introduced in a second phase to support the variant.

2. The same loop, but with a slightly modified while condition:

```
ASSERT(true)
i = 0;
y = 1;
while (i < n) {
    y = y * x;
    i++; // i = i + 1;
}
ASSERT(y == power(x,n))
```

Proof:

```
ASSERT(true)
ASSERT(1 == 1)
ASSERT(1 == power(x,0))
i = 0;
ASSERT(1 == power(x,i))
y = 1;
ASSERT(y == power(x,i))
while (i < n) {
    ASSERT(y == power(x,i) && i < n)
    FACT(i+1 <= n <=> i < n)
    ASSERT(y == power(x,i) && i+1 <= n && i < n)
    ASSERT(y == power(x,i) && i+1 <= n)
    ASSERT(y*x == power(x,i) * x && i+1 <= n)
    y = y * x;
    ASSERT(y == power(x,i) * x && i+1 <= n)
    ASSERT(y == power(x,i+1) && i+1 <= n)
    i++; // i = i + 1;
    ASSERT(y == power(x,i) && i <= n)
}
ASSERT(y == power(x,i) && i >= n && i <= n)
ASSERT(y == power(x,n) && i >= n && i <= n) // needs further strengthening
ASSERT(y == power(x,n) && i >= n)
ASSERT(y == power(x,n))
```

If the range for the loop variable is not given then we need to further strengthen the post-condition as shown above. If the range is given then this would be the only strengthening needed (try it and see for yourself).

3. Same thing but slightly more quickly:

```
ASSERT(n >= 0)
i = 0;
y = 1;
while (i != n) {
    y = y * x * x;
    i = i + 2;
}
ASSERT(y == power(x,n))
```

The proof was left as an exercise, but the key takeaway is that pre-condition has to be strengthened to ensure total correctness:

```
ASSERT(n >= 0 && n % 2 == 0)
i = 0;
y = 1;
while (i != n) {
    y = y * x * x;
    i = i + 2;
}
ASSERT(y == power(x,n))
```

The property `i==n <=> n-i == 0` is still useful for the establishment of a variant; this time the variant is `(n-i)/2`.