

Programming Languages: Imperative Program Construction

Practicals 8: Case Studies

Shin-Cheng Mu

Autumn Term, 2021

1. For $r, b : \text{Int}$, verify the following program.

```

{0 ≤ r < b ∧ even b}
b := b / 2
if r < b → skip
  | b ≤ r → r := r - b
fi
{0 ≤ r < b}

```

2. Derive a $O(\log N)$ algorithm for computing square root:

```

con N : Int {0 ≤ N}
var x : Int
?
{x2 ≤ N < (x + 1)2} ,

```

by introducing a variable y and use $P_0 \wedge P_1$ as the invariant, where

$$P_0 \equiv x^2 \leq N < (x + y)^2 ,$$

$$P_1 \equiv 0 \leq k \wedge y = 2^k .$$

3. Derive, again, a $O(\log N)$ algorithm for computing square root:

```

con N : Int {0 ≤ N}
var x : Int
?
{x2 ≤ N < (x + 1)2} .

```

- (a) This time, construct an algorithm using binary search. What Φ will you use? Does your program rely on the fact that x^2 is monotonic on x (that is, $x \geq y \Rightarrow x^2 \geq y^2$)?
- (b) Knowing that $x \geq y \Rightarrow x^2 \geq y^2$, after the loop terminates, what can you conclude in addition to $x^2 \leq N < (x + 1)^2$?