

Programming Languages: Imperative Program Construction

Practicals 7: Loop Constuction III

Shin-Cheng Mu

Autumn Term, 2021

1. Solve:

```

con  $A, B : \text{Int} \{A \geq 0 \wedge B \geq 0\};$ 
var  $r : \text{Int};$ 
 $S$ 
 $\{r = A \times B\}$  ,
    
```

using only ($/2$) (integral division by two), ($\times 2$), *even*, *odd*, addition, and subtraction.

2. The sum of all digits of a natural number can be computed by

```

 $sd\ 0 = 0$ 
 $sd\ x = x \% 10 + sd\ (x / 10)$  ,
    
```

where ($/$) is integral division and $a \% b$ computes the remainder of a / b . Solve

```

con  $N : \text{Int} \{0 \leq N\}$ 
var  $r : \text{Int}$ 
 $?$ 
 $\{r = sd\ N\}$ 
    
```

3. Given a natural number N , derive a program that computes the number of factors 3 of N . For example, when $N = 945 = 3^3 \times 5 \times 7$ we output 3.

```

con  $N : \text{Int} \{0 \leq N\}$ 
var  $r : \text{Int}$ 
 $?$ 
 $\{r = \text{how do you write the post condition?}\}$ 
    
```

4. Solve:

```

con  $N, X : \text{Int} \{0 \leq N\}$ 
con  $f : \text{array}[0..N) \text{ of } \text{Int}$ 
var  $r : \text{Int}$ 
 $?$ 
 $\{r = \langle \sum i : 0 \leq i < N : f[i] \times X^i \rangle\}$ 
    
```

We have seen this problem before but let us do it slightly differently this time. (This problem is not that much about associativity, but a practice constructing and using recursive function definition.)

- Define $g\ n = \langle \sum i : n \leq i < N : f[i] \times X^{i-n} \rangle$ for $0 \leq n \leq N$, derive a recursive definition of g .
- Use $r = g\ n$ as the main invariant, construct a program that solves the problem.

5. The function *fusc* is defined on natural numbers by:

$$\begin{aligned} \text{fusc } 0 &= 0 \\ \text{fusc } 1 &= 1 \\ \text{fusc } (2 \times n) &= \text{fusc } n \\ \text{fusc } (2 \times n + 1) &= \text{fusc } n + \text{fusc } (n + 1). \end{aligned}$$

Derive a program computing *fusc* *N* for $N \geq 0$. Hint: try *fusc* 78.

6. Solve:

```
con N : Int {0 ≤ N}  
con f : array [0..N] of Int  
var r : Bool  
?  
{r =  $\langle \exists i : 0 \leq i < N : f[i] = 0 \rangle$ }
```

- (a) Define, for $0 \leq n \leq N$, $g\ n = \langle \exists i : n \leq i < N : f[i] = 0 \rangle$. Come up with a recursive definition of *g*.
- (b) Try come up with a program that, as soon as a zero is found in the array, terminates without having to scan the entire list. What invariant would you choose?