Programming Languages: Imperative Program Construction
Practicals 7: Loop Constuction III

Shin-Cheng Mu

Autumn Term, 2024

1. Solve:

   $$\begin{array}{l}\textbf{con}\ A, B : \mathit{Int}\{A \geqslant 0\ \wedge B \geqslant 0\};\\ \textbf{var}\ r : \mathit{Int};\\ S\\ \{r = A \times B\}\ ,\end{array}$$

   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

   $$\begin{array}{l}sd\ 0 = 0\\ sd\ x = x\ \%\ 10 + sd\ (x\ /\ 10)\ \ , for|x > 0|,\end{array}$$

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

   $$\begin{array}{l}con\ N : Int\ \{0 \leqslant N\}\\ var\ r : Int\\ ?\\ \{r = sd\ N\}\end{array}$$

3. Given integral number N, derive a program that computes the number of factors 3 of N. For example, when $N = 945 = 3^3 \times 5 \times 7$, the program should store the value 3 in variable r. You are allowed to use integral division and $(\%)$ (the operator for remainders).

   $$\begin{array}{l}con\ N : Int\ \{N..\}\quad \text{-- what should the constraint on } N \text{ be to make the problem easier?}\\ var\ r : Int\\ ?\\ \{...r = ...\text{how do you write the post condition?}\}\end{array}$$

4. Solve:

   $$\begin{array}{l}\textbf{con}\ N, X : Int\ \{0 \leqslant N\}\\ \textbf{con}\ f : \textbf{array}\ [0..N)\ \textbf{of}\ Int\\ \textbf{var}\ r : Int\\ ?\\ \{r = \langle \Sigma i : 0 \leqslant i < N : f[i] \times X^i \rangle\}\end{array}$$

   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.)

   (a) Define $g\ n = \langle \Sigma i : n \leqslant i < N : f[i] \times X^{i-n} \rangle$ for $0 \leqslant n \leqslant N$, derive a recursive definition of g.

(b) 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:

$$\begin{array}{lcl} \mathit{fusc}\ 0 & = & 0 \\ \mathit{fusc}\ 1 & = & 1 \\ \mathit{fusc}\ (2 \times n) & = & \mathit{fusc}\ n \\ \mathit{fusc}\ (2 \times n + 1) & = & \mathit{fusc}\ n + \mathit{fusc}\ (n + 1). \end{array}$$

Derive a program computing $\mathit{fusc}\ N$ for $N \geqslant 0$. Hint: try $\mathit{fusc}\ 78$.

6. Solve:

$$\begin{array}{l} \textbf{con}\ N : \mathit{Int}\ \{0 \leqslant N\} \\ \textbf{con}\ f : \textbf{array}\ [0..N)\ \textbf{of}\ \mathit{Int} \\ \textbf{var}\ r : \mathit{Bool} \\ ? \\ \{r = \langle \exists i : 0 \leqslant i < N : f[i] = 0 \rangle\} \end{array}$$

(a) Define, for $0 \leqslant n \leqslant N$, $g\ n = \langle \exists i : n \leqslant i < N : f[i] = 0 \rangle$. 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?