OTE: Ohjelmointitekniikka
Programming Techniques

course homepage: `http://www.cs.uku.fi/~mnykanen/OTE/`

Week 03/2009

Exercise 1.

(a) Write the well-known Fibonacci function

$$\mathit{fib}(0) = 0 \tag{1}$$
$$\mathit{fib}(1) = 1 \tag{2}$$
$$\mathit{fib}(m+2) = \mathit{fib}(m+1) + \mathit{fib}(m) \tag{3}$$

as a recursive GCL procedure.

(b) Prove your procedure correct.

Exercise 2. Let a be an array containing both positive and negative numbers. We want to find its indices l and r such that the sum

$$\sum_{i=l}^{r} a[i]$$

of the slice $a[l \ldots r]$ is as large as possible.

(a) Express this pre- and postcondition formally.

(b) Manipulate the postcondition into the initialization, invariant and guard of the search loop.

(c) Write the corresponding GCL program.

Exercise 3. In the *saddleback search* problem, we are given a rectangular matrix a with rows $0, 1, 2, \ldots, M-1$ and columns $0, 1, 2, \ldots, N-1$ and an element x which is guaranteed to be in a. Moreover, we also know that the rows and colums of a are ordered: always $a[p][q] \leq a[p][q+1]$ and $a[p][q] \leq a[p+1][q]$. We must find indices i and j such that $a[i][j] = x$.

(a) Express this pre- and and postcondition formally.

(b) Manipulate the postcondition into the initialization, invariant and guard of the search loop.

(c) Write the corresponding GCL program.

(d) How did the ordering help, compared with the general matrix search in the lectures (Figure 12)?

Exercise 4. Redo the three parts (a)–(c) of Exercise 3, but this time x is not guaranteed to be in a, so that the search may also fail.

Exercise 5. Consider the subroutine

$$
\begin{array}{l}
\{\text{ pre: TRUE} \\
\quad \text{post: } x = y + z \;\} \\
\textbf{proc}\ \ \mathit{sum}(\,\textbf{result}\ \ x : \mathbb{R};\ \ \textbf{value}\ \ y, z : \mathbb{R}\,); \\
\mathcal{B}
\end{array}
$$

Verify that the new value of p after the call $\mathit{sum}(p, p, p)$ is twice its old value before the call.