Since this is the last week of tutorials, the present problem set contains only three Classroom Problems and one Supplementary Problem pertaining to Lecture 12. The course exam, which covers material in Lectures 1–11 and the respective Tutorials, takes place on Monday 11 April, 5–8 p.m. in Otakaari 1 ("Undergraduate Centre") Lecture Halls A (Aalto) and B. More information on the arrangements is provided on MyCourses Announcements. Recall also that you must have passed all your computerised "Astra" assignments before the exam, otherwise your exam will not be graded. If you haven't completed them yet, now is a good time to start working on the outstanding ones.

Instructions

- Classroom Problems C12.1–C12.3 will be discussed and solved onsite at the tutorial sessions in lecture week 12. No credit is given for these problems.
- Supplementary Problem S12.1 is provided for additional illustration and practice. This can also be discussed at the tutorial sessions in week 12, if there is time and interest. No credit is given.

Classroom Problems

C12.1 Verify the claim in Section 12.2 of Lecture 12 that for all $n \geq 0$, $m_0 \in \mathbb{Z}$ the following state-transformation relationship holds:

$$\begin{array}{l} \langle i = n, m = m_0 \rangle \\ [\![\textbf{while } i > \mathit{0} \textbf{ do } m \leftarrow 2 * m; i \leftarrow i - 1 \textbf{ od}]\!] \\ \langle i = 0, m = m_0 * 2^n \rangle \end{array}$$

(*Hint:* Induction on n.)

C12.2 Verify the following loop invariant claims in Section 12.4 of Lecture 12 hold for any $N \in \mathbb{Z}$:

(i)

$$\{m * 2^i = N\} m \leftarrow 2 * m;\ i \leftarrow i - 1 \{m * 2^i = N\}$$

(ii)

$$\begin{array}{l} \{m * 2^i = N\} \\ \textbf{while } i > \mathit{0} \textbf{ do } m \leftarrow 2 * m;\ i \leftarrow i - 1 \textbf{ od} \\ \{m * 2^i = N\} \end{array}$$

You don't need to go back to the detailed semantics of the programming language here; commonsense mathematical reasoning suffices.

C12.3 Prove that the following maximum-finding algorithm is totally correct with respect to precondition P and postcondition Q, where:

> P: *$n \geq 1$ is an integer and $A[1:n]$ is an integer array*[1] *with n elements.*
>
> Q: $a = \max\{A[i] \mid i = 1, \ldots, n\}$.

$k \leftarrow 1$;
$a \leftarrow A[1]$;
while $k < n$ **do**
    $k \leftarrow k + 1$;
    if $a < A[k]$ **then** $a \leftarrow A[k]$
od

(*Hint:* For weak correctness, first find an appropriate loop invariant.)

Supplementary Problems

S12.1 Prove that the following "selection sort" algorithm is totally correct with respect to precondition P and postcondition Q, where:

> P: *$n \geq 1$ is an integer and $A[1:n]$ is an integer array with initial values $A[1:n] = A_0[1:n]$.*
>
> Q: *$A \cong A_0$*[2] *and $A[1] \leq A[2] \leq \cdots \leq A[n]$.*

$k \leftarrow 1$;
while $k \leq n$ **do**
    $kmin \leftarrow k$;
    $j \leftarrow k + 1$;
    while $j \leq n$ **do**
        if $A[j] < A[kmin]$ **then** $kmin \leftarrow j$;
        $j \leftarrow j + 1$
    od;
    $A[k] \leftrightarrow A[kmin]$;
    $k \leftarrow k + 1$
od

[1] "vector", "list",... depending on your preferred terminology.
[2] We use the notation $A \cong B$ here to indicate that arrays A and B contain the same values, but possibly in different order.