# STiDC'06: Exercise 3

21st November 2006

## 1 Problem 1

Write an algorithm that implements consensus for 2 processes using queues that are *initially empty* and atomic registers. Note that a queue is defined by the following operations:

```
macro ENQ(Q, element)
begin
  Q := Append(Q, element);
end macro

macro DEQ(Q, result)
begin
  result := Head(Q);
  Q := Tail(Q);
end macro
```

where each queue variable $Q$ is initialized to $\langle\,\rangle$ (an empty sequence).

## 2 Problem 2

Assume we have a shared object $Q$ that implements, among others, an operation $init(s)$ that atomically changes the state of $Q$ to $s$. Let $A$ be an algorithm that implements $n$-process consensus using object $Q$ initialized to some state $q \neq \perp$. Find an algorithm $B$ that implements $n$-process consensus using algorithm $A$, a number of instances of object $Q$ initialized to $\perp$, and atomic registers, or prove that such an algorithm does not exist.

Note: for this problem no $^+$CAL code is needed. The solution should be included in the report and described. However, the $^+$CAL notation should be used for an algorithm (if any).