SPTCC Summer School on Concurrent Computing

June 3-7, 2017

Simple exercises related to Michel Raynal's lectures

1 Exercise 1: on the kFK universal construction

Slides 50-60 presented a very simple universal construction suited to the computation model $\mathcal{CARW}$[LL/SC]. This construction uses an internal procedure apply(), which is based on a "repeat twice" loop statement. Let us redefine this operation where "repeat twice" is replaced by a "repeat until" statement as follows:

internal procedure apply() **is**
  repeat
    $ls \leftarrow STATE$.LL();
    $pairs \leftarrow BOARD$.collect();
    for $\ell \in \{1, 2, \cdots, n\}$ **do**
      if $(pairs[\ell].sn = ls.sn[\ell] + 1$ **then**
        $\langle ls.value, r\rangle \leftarrow \delta(ls.value, pairs[\ell].op)$;
        $ls.res[\ell] \leftarrow r$;
        $ls.sn[\ell] \leftarrow pairs[\ell].sn$
      end if
    end for
  until $STATE$.SC(ls) **end repeat**.

Is this modification correct? If it is correct provide a proof of it. If it is incorrect, provide a counter-example.

2 Exercise 2: on operations on memory locations

Let us consider the model $\mathcal{CARW}[\emptyset]$ enriched with the following atomic hardware-provided operations. Hence (as the read and write operations) these operations can access any memory location, or the very same location. Let X denote a memory location, and α an integer greater than 1.

- X.multiply(α) multiplies by α the value in X. (Hence if $X = x$ when X.multiply(α) is invoked we have $X = \alpha \times x$ when it returns.)
- X.decrement() decrements by 1 the value in X. (Hence if $X = x$ when X.decrement() is invoked we have $X = x - 1$ when it returns.)

Show that in the system $\mathcal{CARW}[\emptyset]$ enriched with multiply(α) and decrement() (in addition to read()), consensus can be solved for ANY number of processes. To this end design (and prove correct) in this computing model, a binary consensus algorithm which works for any number of processes.

If time permits (... much more difficult ...) try to show that binary consensus for 2 processes is impossible in $\mathcal{CARW}[\emptyset]$ enriched with only one of the operations decrement() or multiply(α)). (Actually, $\mathcal{CARW}[\emptyset]$ enriched with only one of these operations has consensus number 1.)