Concurrent Algorithms November 4, 2019

Exercise 6

Problem 1. A *k-set-agreement* object is a generalization of a consensus object in which processes could decide up to k different values. Formally, k-set-agreement is defined as follows. It has an operation *propose*(v) that returns (or we say *decides*) a value, which satisfies the following properties:

1. *Validity:* Decided values are proposed values.
2. *Agreement:* At most k different values could be decided.
3. *Termination:* Every correct process eventually decides a value.

A *k-simultaneous-consensus* object is another generalization of a consensus object in which processes could decide k values simultaneously. Formally, k-simultaneous consensus is defined as follows. It has an operation *propose*($v_1, \ldots, v_k$) that returns (or we say *decides*) a pair $(index, value)$ with $index \in \{1, \ldots, k\}$, which satisfies the following properties:

1. *Validity:* If a process decides (i, v), then some process proposed $(v_1, \ldots, v_k)$ with $v_i = v$.
2. *Agreement:* If two processes decide (i, v) and (i', v') with $i = i'$, then $v = v'$.
3. *Termination:* Every correct process eventually decides a value.

Your task is to show that k-set-agreement and k-simultaneous-consensus are equivalent. That is, you have to show that one implements the other.

Hint: When implementing k-consensus using k-set-agreement, an algorithm that solves the problem is the following:

1: **function** KSC.PROPOSE($v_1, \ldots, v_k$)
2: $V_i \leftarrow [v_1, \ldots, v_k]$
3: $dV_i \leftarrow kSA.\text{PROPOSE}(V_i)$
4: $REG[i] \leftarrow dV_i$
5: $snap_i \leftarrow REG.\mathsf{snapshot}()$
6: $c_i \leftarrow$ number of distinct (non-$\perp$) vectors in $snap_i$
7: $d_i \leftarrow$ minimum (non-$\perp$) vector in $snap_i$
8: $\mathsf{return}\langle c_i, d_i[c_i] \rangle$
9: **end function**

Where $REG[0, \ldots, n-1]$ is an array of single-writer multi-readers atomic registers initialized at $\perp$. Processes write atomically a *vector of values* in their register (Line 4). $REG.\mathsf{snapshot}()$ returns an atomic snapshot of this array of registers. Consequently, $snap_i[0, \ldots, n-1]$ is an array of vectors, possibly containing $\perp$ values for some indices. We suppose that there is an order on the set of values that can be proposed, and we use the induced *lexicographic order* on vectors at Line 7.

Your task is then to (1) prove that the algorithm above implements a k-simultaneous consensus from k-set agreement objects and atomic registers; and (2) find an algorithm that implements a k-set agreement object using k-simultaneous consensus objects and atomic registers.