RCC 2013/2014: Quiz 3

Exercise 1: atomic registers

Is it possible to devise an implementation of a one-writer N-reader ($1WNR$) atomic register from any number of one-writer one-reader ($1W1R$) atomic registers in which the readers *do not* write? Justify your answer.

Exercise 2: immediate snapshot

Recall the IS construction using AS. Would the algorithm be still correct if instead of $A_r.update_i(v_i)$ and $S := A_r.snapshot()$ we use

$$U_r[i].write(v_i)$$

and

$$S := scan(U_r[1], \ldots, U_r[N])$$

respectively, where U_r is a shared vector of N one-writer-N-reader atomic registers? What if the one-writer-N-reader registers are only regular? Justify your judgement.

Exercise 3: shellability

Let $\mathcal{S}$ be a simplicial complex. An order $\phi_0, \phi_1, \ldots, \phi_t$ on the facets of $\mathcal{S}$ is *shelling* if and only if, for every i, j such that $0 \leq i < j \leq t$, there exists ϕ_k satisfying: (i) $0 \leq k < j$, (ii) $\phi_i \cap \phi_j \subseteq \phi_k \cap \phi_j$, and (iii) $|\phi_j \setminus \phi_k| = 1$.

A complex is *shellable* if there is a shelling order on its facets.

The k-skeleton of $\mathcal{S}$, $skel^k(\mathcal{S})$ is the complex formed by all the simplices of $\mathcal{S}$ of dimension k or less.

Prove that for each simplex Δ, for each $k \in [0, dim(\Delta)]$, $skel^k(\Delta)$ is shellable.

Hint: Order the vertices Δ according to any order, and then order the simplices of Δ lexicographically.

Exercise 4: critical run

Consider any algorithm A that solves binary consensus in the wait-free manner using any collection of atomic objects. We assume that the objects are deterministic and, thus, any run of A is unambiguously determined by its initial state and the sequence of process identifiers specifying the order in which the processes take steps.

A finite run is called *v-valent* ($v \in \{0, 1\}$) if v is the only value decided in extensions of R. A run is *bivalent* if it has an extension in which 0 is decided and an extension in which 1 is decided.

Show that A has a *critical* run, i.e., a bivalent run R such that for all p_i, $R.i$ (extension of R with one step of p_i) is *univalent* (0-valent or 1-valent).

Exercise 5: three-process consensus with queues

Show that queues and registers are not strong enough to solve consensus among three or more processes.
Hint: Consider a critical run R such that the decision value in any extension of it is defined by the state of some queue object: e.g., $R.0$ is 0-valent, $R.1$ is 1-valent, and both p_0 and p_1 are *about* to access a queue object Q at the end of R.