Problem 1. A \textit{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 \(\text{propose}(v)\) that returns (or we say \textit{decides}) a value, which satisfies the following properties:

1. \textbf{Validity}: Decided values are proposed values.
2. \textbf{Agreement}: At most \(k\) different values could be decided.
3. \textbf{Termination}: Every correct process eventually decides a value.

A \textit{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 \(\text{propose}(v_1, \ldots, v_k)\) that returns (or we say \textit{decides}) a pair \((\text{index}, \text{value})\) with \(\text{index} \in \{1, \ldots, k\}\), which satisfies the following properties:

1. \textbf{Validity}: If a process decides \((i, v)\), then some process proposed \((v_1, \ldots, v_k)\) with \(v_i = v\).
2. \textbf{Agreement}: If two processes decide \((i, v)\) and \((i', v')\) with \(i = i'\), then \(v = v'\).
3. \textbf{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.

\textbf{Hint}: When implementing \(k\)-consensus using \(k\)-set-agreement, an algorithm that solves the problem is the following:

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

Where \(\text{REG}[0, \ldots, n - 1]\) is an array of single-writer multi-readers atomic registers initialized at \(\perp\). Processes write atomically a \textit{vector of values} in their register (Line 4). \(\text{REG.snapshot}()\) returns an atomic snapshot of this array of registers. Consequently, \(\text{snap}_i[0, \ldots, n - 1]\) 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 \textit{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.