Immediate Snapshot Algorithm

(a) Prove that, in the Algorithm of Slide 24, if at most $x$ processes invoke \textsc{rec\_write-snapshot}(x, -), then at most $x-1$ invoke \textsc{rec\_write-snapshot}(x-1,-) at Line 12 and at least one returns a view at Line 10.

(b) Prove that the Algorithm of Slide 24 fulfills the termination, self-inclusion, containment and immediacy properties.

(c) Compute the complexity of the Algorithm of Slide 24 in terms of number of shared memory accesses.

Consensus Algorithm for 2 Processes

Suppose that 2 processes run the following algorithm:

1: \textbf{function} propose(v)
2: \hspace{1cm} est $\leftarrow$ v
3: \hspace{1cm} $r \leftarrow$ 1
4: \hspace{1cm} \textbf{while} true \textbf{do}
5: \hspace{1.5cm} \textit{view} $\leftarrow$ \textit{IS}[r].\textsc{write-snapshot}(s)
6: \hspace{1.5cm} \textbf{if} $|\textit{view}| = 2$ \textbf{then}
7: \hspace{2cm} est $\leftarrow v_j$ such that $(j, v_j) \in \textit{view}$ and $j \neq id$
8: \hspace{1.5cm} $r \leftarrow r + 1$
9: \hspace{1.5cm} \textbf{else}
10: \hspace{2cm} \textbf{return} est

(a) Show that this algorithm doesn’t solve consensus.

(b) Suppose that in any execution, there is a round such that the invocations of \textsc{write-snapshot} by the two processes are not set-linearized together. Prove that this algorithm then solves consensus between the two processes.

(c) Draw the subdivided segment representing the possible states of the system after a few rounds. Then, tag the possible state of the processes with their decision values.

$k$-Set Agreement with Less than $k$ crashes

Find and prove an algorithm solving the $k$-set agreement among $n > k$ processes in presence of at most $k - 1$ crashes.