Consensus Algorithm for 2 Processes

Suppose that 2 processes run the following algorithm:

1: function propose(v)
2:   est ← v
3:   r ← 1
4:   while true do
5:     view ← IS[r].write-snapshot(s)
6:     if |view| = 2 then
7:         est ← v_j such that (j, v_j) ∈ view and j ≠ id
8:         r ← r + 1
9:     else
10:        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 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.

Solution sketch  (a) The two processes might be set-linearized together forever, thus not satisfying termination. (b) Termination is satisfied because we assume the two processes are not set-linearized together forever, so the check in line 6 will eventually fail and propose will return in the else branch. Validity is satisfied because est is equal to a proposed value at all times. Agreement is satisfied: the second process to decide will choose the same value as the first process to decide. This because the second process will have seen the first process and adopted its value (line 7). (c) See separate figure on course website.

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.

Solution sketch  Use n registers, one per process. Each process waits for at least n − k + 1 registers to be filled with processes’ proposed values, then returns the minimum.