Concurrent Algorithms October 17, 2017

Exercise 4

Problem 1. Given the algorithm that implements immediate snapshot in the lecture, your task is to show that if at most x processes invoke rec_update_snapshot(x, -), then

(a) At most $(x - 1)$ processes invoke rec_update_snapshot($x - 1$,-) and

(b) At least one process returns during its invocation rec_update_snapshot(x,-).

Problem 2. In the lecture, we saw that in Epoch-based Reclamation, a deleting process takes a snapshot of all process epochs and records it with the deleted node in the limbo list. Does this snapshot need to be atomic? Why or why not?

Problem 3. Write pseudocode for the SCAN method of Hazard Pointers. Recall that the SCAN method goes through the limbo list of a process and checks for every node in the list if it protected by some process' hazard pointer. Can you implement SCAN such that it runs in $O(m + n)$ time, where m is the number of nodes in the limbo list and n is the number of processes?