Assignment 7*

Problem Statement Give an algorithm using only multi-writer registers for solving ϵ-approximate agreement with any $0 < \epsilon < 1$ for inputs in $[0, 1]$ that has $O\left(\log_2\left(\frac{1}{\epsilon}\right)\right)$ step complexity. Prove your algorithm is correct. This is another submission of assignment 7 in light of the above changes. If it is possible to use atomic snapshot objects please refer to the previous submission.

Algorithm. Let the processes be $p_1, ..., p_n$ and let the input to process p_i be $x_i \in [0, 1]$. Our algorithm is based off of the one shown in class for the case where $\epsilon = \frac{1}{2}$. We make use of $r = \lceil \log 1/\epsilon \rceil + 1$ sets of multi-reader multi-writer registers $W^1, ..., W^r$ where each W^k contains 2^k registers $W^k[0], ..., W^k[2^k - 1]$. All registers are initialized to 0.

The high-level description of the algorithm is as follows: the processes will interact with the sets of registers in order. In round k the unit interval is divided into 2^k sub-intervals $I_0 = [0, 1/2^k]$, $I_1 = [1/2^k, 2/2^k]$, ..., $I_{2^k-1} = [(2^k - 1)/2^k, 1]$ with each $W^k[j]$ associated with I_j. When we read or write from I_j we are actually reading and writing to $W^k[j]$.

If $x_i = \frac{j}{2^k}$ for $j = 0, ..., 2^k - 1$, process p_i sets the intervals to the left and right of x_i to 1. Then p_i reads the interval two spaces to the left of x_i. If p_i gets a 1 then p_i updates x_i to $\frac{j-1}{2^k}$. Otherwise, p_i reads the interval two spaces to the right of x_i. If this interval is 1 then p_i updates x_i to $\frac{j+1}{2^k}$. If this value is 0 then x_i remains unchanged. If x_i is only contained in one interval $I_j = [j/2^k, (j+1)/2^k]$ then p_i will write 1 to I_j. Next p_i reads the value of the interval to the left of I_c. If this value is 1 then p_i updates x_i to be $j/2^k$. Otherwise p_i reads the value of the interval to the right of I_c. If its value is 1 then p_i updates x_i to $(j + 1)/2^k$. If not then x_i remains unchanged. (*In the above, we only read and write if the interval exists.*)

Claim 1. *The above algorithm solves ϵ-agreement for any $\epsilon \in (0, 1)$ and inputs in $[0, 1]$. Further the step complexity for each process is $O(\log 1/\epsilon)$.*

Proof. It is easy to see that the algorithm satisfies the termination condition since each process iterates for r rounds and each round consists of constantly many read and write operations. By the same reasoning, each process takes $O(\log 1/\epsilon)$ steps.

We show that the algorithm satisfies the validity condition by induction on the number of rounds. Initially x_i are the inputs so validity is satisfied. Consider process p_i in round $k > 1$. By the induction hypothesis there exists x and x' such that $x \leq x_i \leq x'$ for x_i in round $k - 1$. If x_i did not change in round k, then validity holds. Otherwise, w.l.o.g suppose x_i changed to $\frac{c}{2^k}$ such that x_i was smaller than $\frac{c}{2^k}$. Then there must be some p_j which set I_c to 1. By the induction hypothesis there exists input x'' with $\frac{c}{2^k} \leq x''$. Then $x \leq x_i \leq x''$ in the k^{th} round.

We prove ϵ-agreement by induction on the number of rounds. Namely, at the end of round k, all x_i will be in an interval of length $1/2^k$. This is true initially since the inputs are in $[0, 1]$. Consider x_i and x_j of process p_i and p_j in at the end of round k. By the induction hypothesis, at the beginning of round k, x_i and x_j are in some interval $I_c = [c/2^{k-1}, (c+1)/2^{k-1}]$ of length $1/2^{k-1}$. In round k, I_c is divided into $I_{2c} = [2c/2^k, (2c + 1)/2^k]$ and $I_{2c+1} = [(2c + 1)/2^k, (2c + 2)/2^k]$. If x_i and x_j are in the same interval then we are done. Thus W.l.o.g suppose $x_i \in I_{2c}$ and $x_j \in I_{2c+1}$ and p_j wrote 1 to its intervals first. p_j will its updated value of x_j to some value in I_{2c+1}. p_i will read the value written by p_j and update x_i to $\frac{2c+1}{2^k}$. Thus at the end of round k, all x_i are in the same interval of length $\frac{1}{2^k}$. After r rounds all outputs will be within ϵ of one another. □