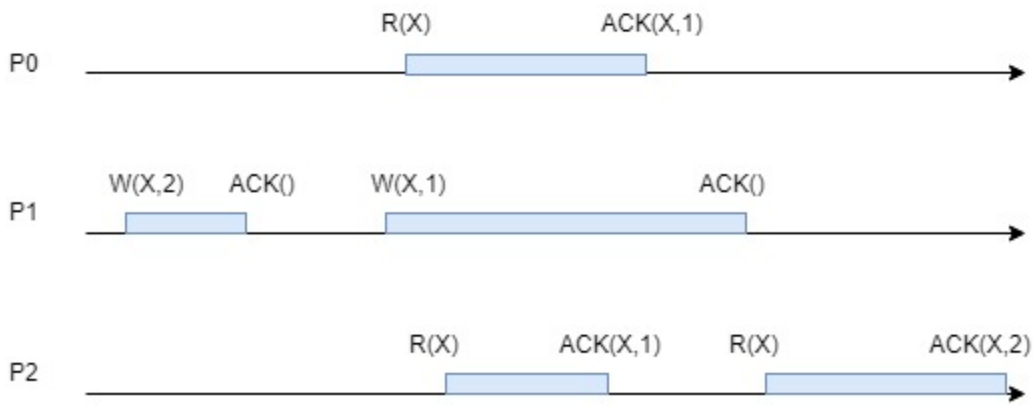
CS 425/ECE 428 Distributed Systems
Homework 4 **Total points: 30**
Due by 5 p.m. on February 22, 2018.
Submit electronically via Compass2g. **PDF format preferred.**

(1) (10 points) In each part of this question, if you answer NO, **briefly explain why** the execution does not satisfy the specified property.

Assume that all variables are initialized to 0

a. Does the execution below satisfy **causal** consistency? (5 points)



No, consider reads in P2 and all the writes: $W_1(X,2)$, $W_1(X,1)$, $R_2(X,1)$, $R_2(X,2)$. There is no permutation of these operations that satisfies both happened-before relation and validity.

Grading policy
-2 for incorrect answer
-3 for incorrect explanation

b. Does the execution below satisfy **causal** consistency? (5 points)

Yes.

Grading policy
-5 for incorrect answer

(2) (10 points) Consider any one of the two algorithms discussed in the class for tolerating f crash failures in a synchronous system. Recall that the algorithm performs f+1 rounds in total.

In this question, assume that f divisible by 3.

Suppose that in a particular system it is guaranteed that, during any given round, either no process crashes, or exactly 3 processes crash. Also, at most f processes may crash during the entire execution.

Under the above assumptions, **determine the minimum number of rounds** that suffice to achieve consensus using the above algorithm (i.e., instead of f+1 rounds in the original algorithm).

Explain your answer briefly.

(f/3)+1 rounds would suffice. For each round, there are either 0 or 3 processes failure. Since at most f processes fail, there is at least one round during which no new process will fail. In this round, all processes that haven't yet failed will receive identical set of values. This ensures that consensus is achieved.

Grading policy
-5 for wrong answer – calculation
-5 for wrong explanation

(3) (10 points) Consider the approximate consensus algorithm discussed in Lecture (see the notes provided for this algorithm). Suppose that f = 2 (i.e., at most 2 process or node may crash). Suppose that the system contains 5 nodes, with inputs 1, 2, 3, 4, and 5 respectively.

After one round of the approximate consensus algorithm, determine the **largest** and the **smallest** value that the local variable y may take at any of these nodes.

Show your work.

In this case n = 5 and f = 2, thus, n-f = 3. Suppose that nodes 1, 2, 3, 4 and 5 have inputs 1, 2, 3, 4 and 5, respectively. The smallest y value is obtained when a node receives values 1, 2, 3 in round 1, respectively, from nodes 1, 2, 3, and computes the new value of y using these three (i.e., n-f) values. Thus, the smallest possible y value after one round is (1+2+3)/3 = 2. Similarly, the largest possible y value after round 1 is (3+4+5)/3 = 4.

Grading policy
-5 for one incorrect answer
-10 for both incorrect answer