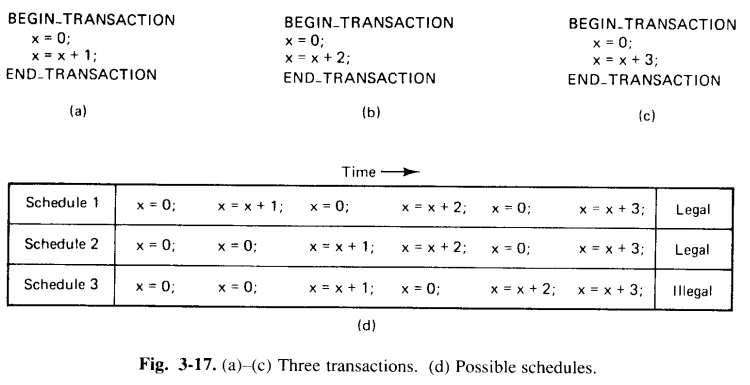
Q.1: A distributed system may have multiple, independent critical regions. Imagine that process 0 wants to enter critical region A and process 1 wants to enter critical reason B. Can Ricart and Agrawal’s algorithm (Distributed algorithm for mutual exclusion) lead to deadlock? Explain your answer.

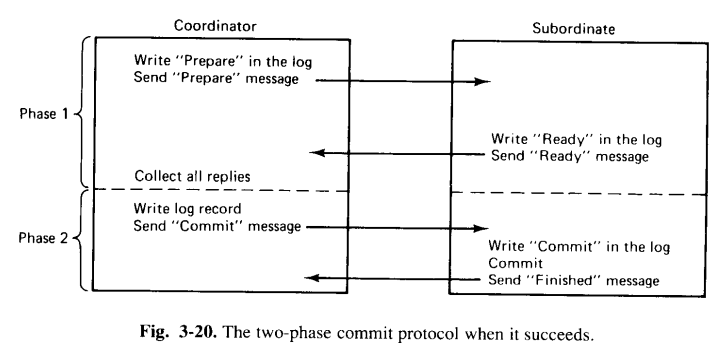
Q.2: Suppose that two processes detect the demise of the coordinator simultaneously and both decide to hold an election using the bully algorithm. what happens?

Q.3: Consider the following:



For the same transactions, give a complete list of all values that x might have at the end, and state which are legal and which are illegal.

Q.4: Consider the following:



In the above figure, at what instant I the point-of-no-return reached? That is, when is the atomic commit actually performed?

Q.5: Systems that use locking for concurrency control usually distinguish read locks from write locks. What should happen if a process has already acquired a read lock and now wants to change it into a write lock? What about changing a write lock into a read lock?

Q.6: Is optimistic concurrency control more or less restrictive than using time stamps? Why?

Q.7: Does using timestamp for concurrency control ensure serializability. Explain.

Q.8: The centralized deadlock detection algorithm initially gave a false deadlock, but was later patched up using global time. Suppose that it has been decided not to maintain the global time (too expensive). Device and alternative way to fix the bug in the algorithm.

Q.9: A process with transaction timestamp 50 needs a resource held by a process with transaction timestamp 100. what happens in:

 A Wait-die?

 B wound-wait?