

Sample solutions

Question 1.

- a) S1. It is not view serializable, and not recoverable.
- b) S4. It is not view serializable, not avoid cascading abort (cascadeless), and recoverable.
- c) S6. Not cascadeless, Recoverable, Conflict Serializable, View Serializable.
- d) S10. Strict. Not View Serializable. Cascadeless. Recoverable.

Question 2.

w3(A), w1(A) c1, w2(B), c2, w3(B), c3

T1 commits before T2 starts, yet the only serialization order is T2,T3,T1.

Question 3.

(a) There are two possible serial executions. T0 T1 and T1 T0. For the first case, $A = 0$ and $B = 1$, which is consistent. For the second case, $A = 1$, and $B = 0$ eventually. Again, consistency is met.

(b) Any interleaving of T0 and T1 results in a non-serializable schedule. Pick any one.

(c) if you try all cases in (b), that can be your answer. There is no parallel execution resulting in a serializable schedule. From part(a), we know that a serializable schedule results in $A=0$ OR $B=0$. Suppose we start with T0 read(A). Then when the schedule ends, no matter when we run the steps of T1, $B=1$. Now suppose we start executing T1 prior to completion of T0. Then T1 read(B) will give B a value of 0. So, when T1 completes, $A = 1$. Thus, $B=1$ AND $A = 1$. We get similar logic for starting with T1 read(B).

Question 4.

Figure shows the resultant polygraph. Since it is acyclic, S is view serializable. The serial order is T2, T1, T3.

