Database Management Systems
Practice Problem Set: Transaction and Concurrency Control

---

1. Consider the following two transactions:

```
T1 : read(A);
     read(B);
     if A = 0 then B := B + 1;
     write(B).
T2 : read(B);
     read(A);
     if B = 0 then A := A + 1;
     write(A).
```

Let the consistency requirement be $A = 0 \lor B = 0$, with $A = B = 0$ the initial values.

a. Show that every serial execution involving these two transactions preserves the consistency of the database.
b. Show a concurrent execution of T1 and T2 that produces a nonserializable schedule.
c. Is there a concurrent execution of T1 and T2 that produces a serializable schedule?

2. Fnd all view-equivalent serial orders for the following schedule:
r1 (X) r3 (T) w1 (Y) r2 (Y) w3 (Y) r4 (Y) w2 (Z) r5 (Z) w4 (S) r5 (S) w5 (Y)

3. Consider the following two transactions:
T1 = w1 (Z) r1 (X) w1 (X) r1 (Y) w1 (Y)
T2 = r2 (Y) w2 (Y) r2 (X) w2 (X)
Let us assume that our scheduler performs exclusive locking only (i.e., no shared locks). Consider the following instances of transactions T1 and T2 annotated with lock, unlock, and (sometimes) commit actions:

(a) T1 = L1 (Z) w1 (Z) L1 (X) r1 (X) w1 (X) U1 (X) L1 (Y) r1 (Y) w1 (Y) U1 (Z) U1 (Y)
T2 = L2 (Y) r2 (Y) w2 (Y) U2 (Y) L2 (X) r2 (X) w2 (X) U2 (X)
(b) T1 = L1 (Z) w1 (Z) L1 (X) r1 (X) w1 (X) L1 (Y) r1 (Y) w1 (Y) C1 U1 (X) U1 (Z) U1 (Y)
T2 = L2 (Y) r2 (Y) w2 (Y) L2 (X) r2 (X) w2 (X) C2 U2 (X) U2 (Y)
(c) T1 = L1 (Z) L1 (X) w1 (Z) r1 (X) w1 (X) L1 (Y) r1 (Y) w1 (Y) C1 U1 (X) U1 (Z) U1 (Y)
T2 = L2 (Y) r2 (Y) w2 (Y) L2 (X) r2 (X) w2 (X) C2 U2 (X) U2 (Y)
(d) T1 = L1 (Y) L1 (Z) w1 (Z) L1 (X) r1 (X) w1 (X) r1 (Y) w1 (Y) C1 U1 (X) U1 (Z) U1 (Y)
T2 = L2 (Y) r2 (Y) w2 (Y) L2 (X) r2 (X) w2 (X) C2 U2 (X) U2 (Y)

For each instance, indicate whether the annotated transactions satisfy the following conditions:

(1) Obey two-phase locking.
(2) Obey strict two-phase locking.
(3) Will necessarily result in a conflict serializable schedule (if no deadlock occurs).
(4) Will necessarily result in a recoverable schedule (if no deadlock occurs).
(5) Will necessarily result in a schedule that avoids cascading rollbacks (if no deadlock occurs).
(6) Will necessarily result in a strict schedule (if no deadlock occurs).
(7) Will necessarily result in a serial schedule (if no deadlock occurs).
(8) May result in a deadlock.

4. Show that there are schedules that are possible under the two-phase locking protocol, but are not possible under the timestamp protocol, and vice versa.

5. Consider a variant of the tree protocol called the forest protocol. The database is organized as a forest of rooted trees. Each transaction Ti must follow the following rules:

- The first lock in each tree may be on any data item.
- The second, and all subsequent, locks in a tree may be requested only i the parent of the requested node is currently locked.
- Data items may be unlocked at any time.
- A data item may not be relocked by Ti after it has been unlocked by Ti.

Show that the forest protocol does not ensure serializability.