EECS 484 HW 5 Chien-Wei Lin

Q1

(A)

No, T1 R(X) after T2 W(X). However T1 commit before T2 commit

(B)

No, reason is same as (A) T1 R(X) before T2 commit.

(C)

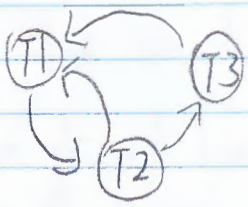
Yes, there's no dependency

(D)

It's not ACA ∴ It can't be generate by S2PL.

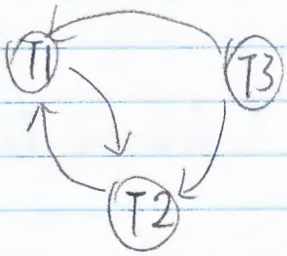
Q2

Schedule A



It's not serializable nor conflict serializable.

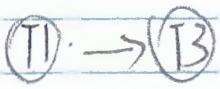
schedule B.



It's not serializable nor conflict serializable

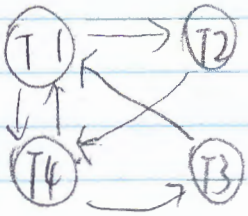
Schedule C

Since T2 is aborted



It's conflict serializable and serializable.

Schedule D.



It's not conflict serializable, but it's serializable

Q3

schedule B:

It doesn't have deadlock.

Schedule C:

It doesn't have deadlock.

schedule D:

There's a deadlock between T2 & T4
T2, T3, T4 have to wait until T1 finishes.
=> T4 grab (B). T2 grab (C) => deadlock.

Q4

(A)

Write a transaction log of timestamp 1 and Page 1 to the disk.

(B)

It violates the atomicity. Since Page 1 has been evict from the memory and no log has record, transaction 1 made some change on disk. However, the system doesn't have the ability to recover it. We have a incomplete transaction here.

(C)

It violates the durability. No pages are written back to disk, even transaction 1 has commited.

(D)

We can force a committed transaction to write pages on disk. However, if there is any transaction wanting to modify these pages, it will increase the disk I/O.

Q5

(A)

Page ID	rec LSN
P1	1
P2	2
P3	3

(B)

Transaction ID	Last LSN	State
T2	5	U
T3	11	C

(C)

T2

(D)

There's no CLR during the redo phase, because no transaction was aborted before the crash.

(E)

There are 2 CLR records are added in in the UNDO phase. Including LSN:2 and 5