Transaction Problems

Note: The notion R_{T1} (A) and W_{T1} (A) means T1 reads and writes object A respectively.

1. Draw the precedence graph for the schedule

W_{T1}(A) W_{T2} (A) R_{T3} (B) W_{T3} (B) R_{T1} (B)

T3->T1->T2

2. For each of the following schedules of reads and writes done by transactions, indicate whether the schedule is conflict serializable, or serial:

1) R_{T1} (A) R_{T2} (A) W_{T1} (A) R_{T2} (B)

Conflict serializable

2) R_{T1}(A) W_{T1} (A) R_{T2}(A) W_{T2} (B)

Conflict serializable; Serial

3) W_{T1} (A) W_{T2} (A) W_{T1} (B) W_{T2} (B) W_{T1} (B)

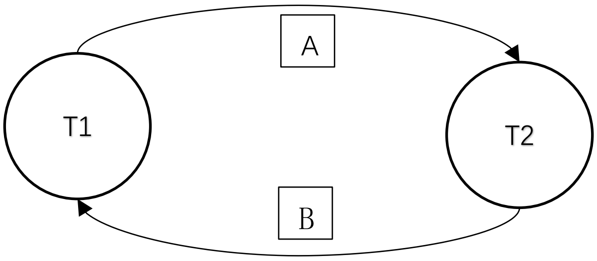
None

3. Consider the following two transactions and schedule

R_{T1}(A) W_{T1}(A) R_{T2}(A) R_{T2}(B) R_{T1}(B) W_{T1}(B)

1) is this schedule conflict-serializable? Explain why or why not.

No, because the precedence graph is as below, which is not acyclic. A schedule is conflict serializable if and only if its precedence graph is acyclic.



2) Show how 2PL can ensure a conflict-serializable schedule for the same transactions above. (use 2PL to re-reschedule the transactions)

Time	Transaction T_1	Transaction T_2
T1	XLOCK A	
T2	Read A	
T3	Write A	SLOCK A
T4	XLOCK B	WAIT
T5	Read B	WAIT
T6	Write B	WAIT
T7	UNLOCK X(A)	Read A
T8	UNLOCK X(B)	SLOCK B
T9	COMMIT	Read B
T10		UNLOCK S(A)
T11		UNLOCK S(B)
T12		COMMIT