Introduction to Database Systems 2021/2022

SÜLEYMAN GÖLBOL [ERASMUS STUDENT]

Homework 3

Task 1 (45 %) - Transaction conflicts

Given are the following schedules:

Schedule 1 (S1)

Time stamp (TS)	*0*	*1*	*2*	*3*	*4*	*5*	*6*
T1	R(A)	R(B)	W(B)				
T2				R(B)	R(A)	W(B)	W(A)

Schedule 2 (S2)

Time stamp (TS)	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*
T1	R(B)					R(A)		R(B)	W(B)
T2		R(A)	W(B)	W(A)					
T3					R(B)		W(A)		

a) (10 %) For EACH given schedule (S1 and S2) determine all conflict pairs.

A pair of actions with Write-Write conflict between (Tx and Ty) over object Z would be noted down as ($W_x(Z)$, $W_y(Z)$).

S1 $(W_1(B), W_2(B))$ $(R_1(A), W_2(A))$ $(W_1(B), R_2(B))$ $(R_1(B), W_2(B))$

S2 $(R_1(B), W_2(B))$ $(W_2(B), R_3(B))$ $(W_2(A), R_1(A))$ $(W_2(A), W_3(A))$ $(W_2(B), R_1(B))$ $(W_2(B), W_1(B))$, $(R_1(A), W_3(A))$, $(R_2(A), W_3(A))$ $(R_3(B), W_1(B))$

b) (5 %) Do we have any blind writes in the second schedule (S2)? If yes, at what time stamp(s)?

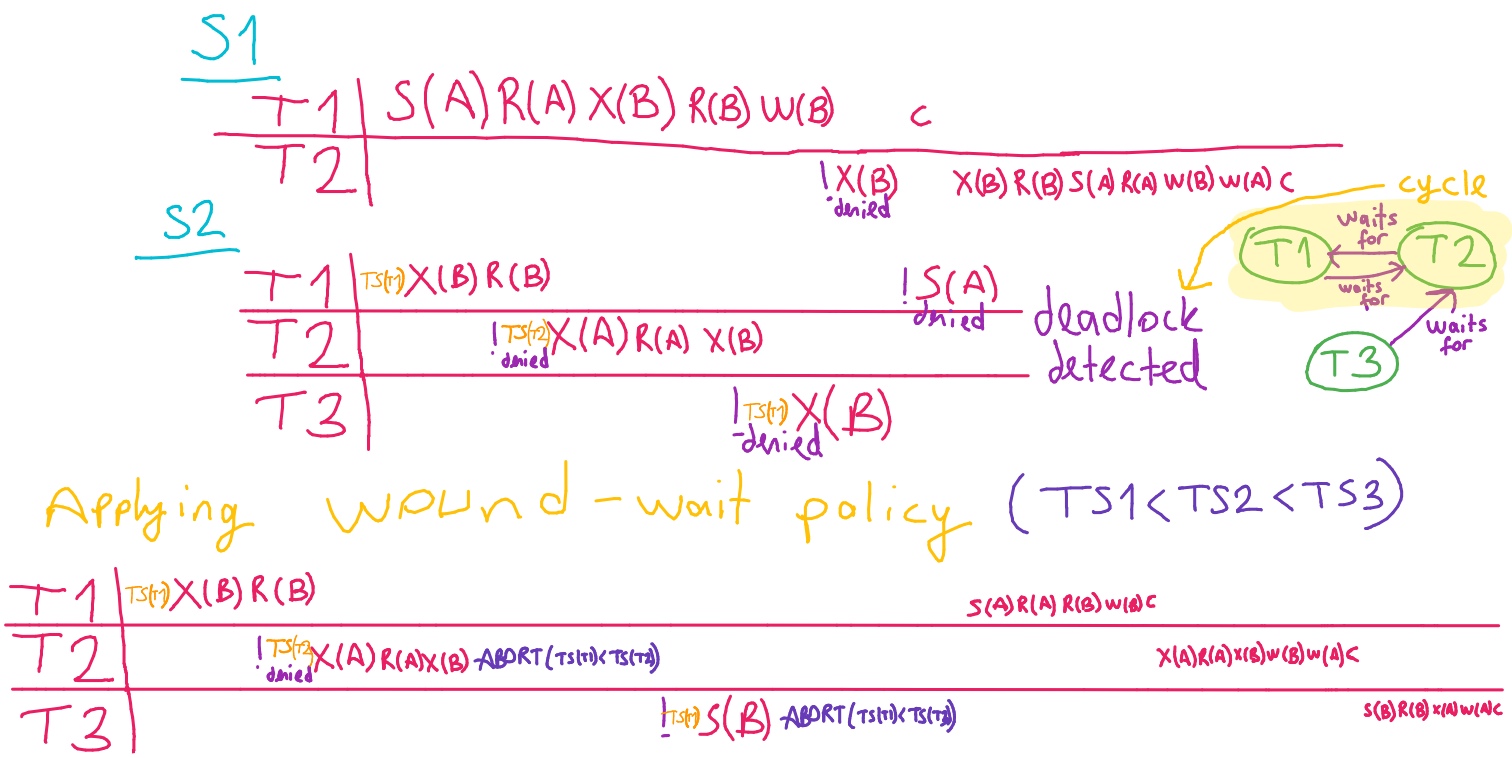
T2 → $W_2(B)$ column number 2

T3 → $W_3(A)$ column number 6

c) (30 %) For EACH schedule (S1 and S2) demonstrate how would you avoid conflicts with the use of Strict Two-Phase Locking (Strict 2PL).

- Apply share/exclusive locks reasonably, assume we foresee what operations we are going to execute.
- Include wait-for graphs for deadlock detection
- In the case of a deadlock situation apply Wound-Wait policy. Priorities are determined according to the time stamps (ex. TS1 < TS2 < TS3 < … < TSX); where transaction with TS1 has the highest priority, meanwhile transaction with TSX is the least prioritized).

Note down all observations.



Task 2 (15 %) – Database consistency

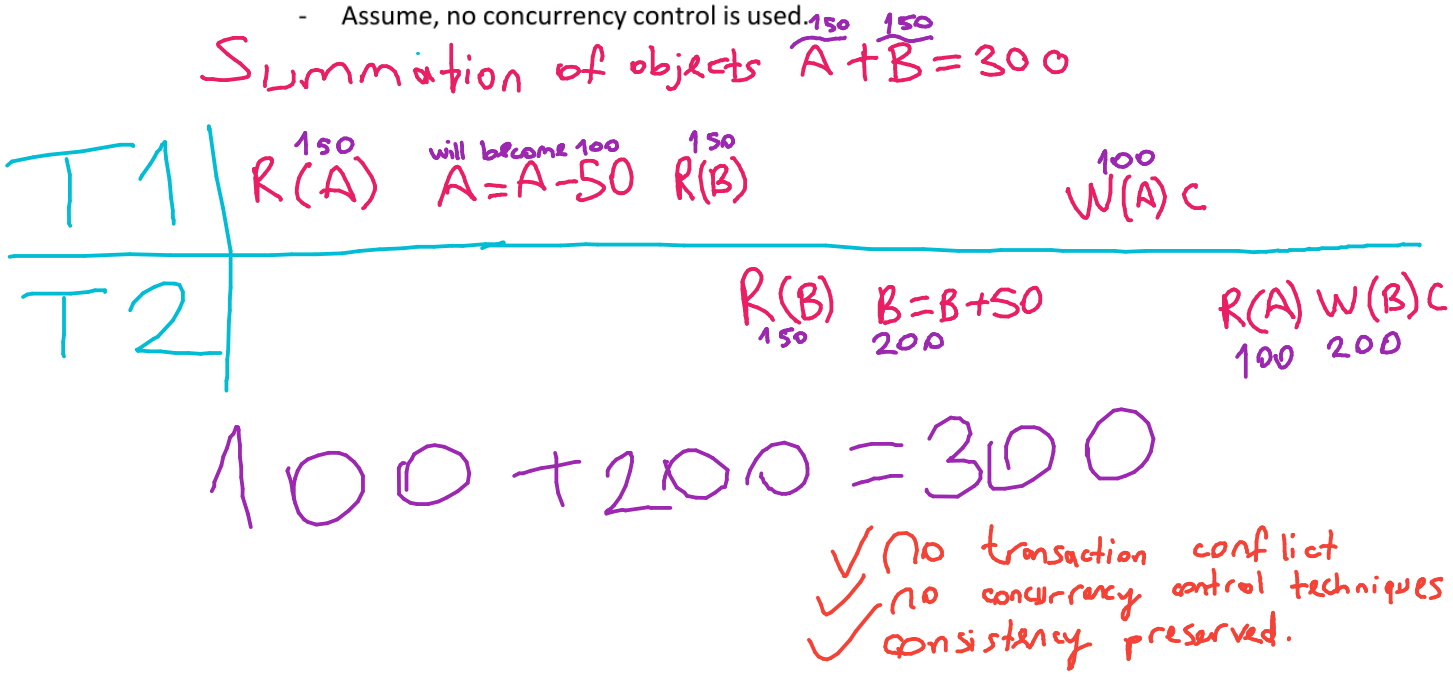
Objects A and B have an initial value of 150. Database consistency is preserved when the summation of objects is the same before and after both transactions are committed.

Transaction 1 (T1)
R(A)
"A = A – 50"
R(B)
W(A)

Transaction 2 (T2)
R(B)
"B = B + 50"
R(A)
W(B)

Find a schedule with concurrent execution of transactions that will preserve database consistency.

- Avoid transaction conflicts.
- Assume, no concurrency control is used.



Task 3 (40 %) – Serial schedule

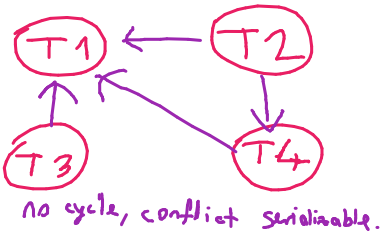
Figure out whether any of the schedules (S1, S2, S3) is conflict serializable.
Provide a proof with including precedence graphs. (Conflict graph) (Serialization graph)

In the graph that we draw, if no cycle then it means it's conflict serializable.

Schedule S is conflict serializable if S is conflict equivalent to some serial schedule.

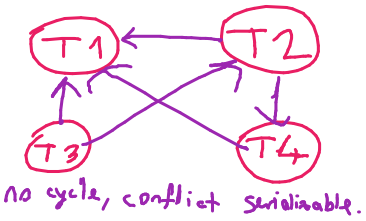
a) (10 %) Schedule 1 (S1)

T1		R(A)			W(A)		W(B)
T2				W(B)			
T3	R(A)						
T4			R(A)			R(B)	



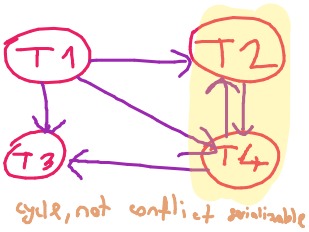
b) (10 %) Schedule 2 (S2)

T1						R(A)	R(B)
T2		W(A)					
T3	R(A)			W(B)	R(C)		
T4			W(A)				



c) (10 %) Schedule 3 (S3)

T1	W(A)					W(C)			
T2				W(B)					R(A)
T3					R(A)			W(A)	
T4		R(A)	W(A)				R(B)		



d) (10 %) For EACH conflict serializable schedule write down ALL corresponding serial schedules. a and b are conflict serializables.

to the one who comes most arrow

d.a) T_3, T_2, T_4, T_1

T_2, T_3, T_4, T_1

T_2, T_4, T_3, T_1

d.b) T_3, T_2, T_4, T_1