CSE 4/560

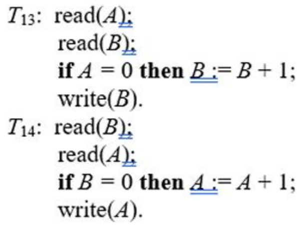
Fall 2024

Databases and Query Languages Homework -3 Bonus

Sahithya Arveti Nagaraju (Person number – 50559752 UBITName - sarvetin)

Problem 1

Consider the following two transactions



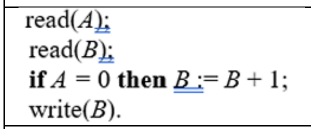
```
$T_{13}$: read($A$);
     read($B$);
     if $A = 0$ then $B := B + 1$;
     write($B$).
$T_{14}$: read($B$);
     read($A$);
     if $B = 0$ then $A := A + 1$;
     write($A$).
```

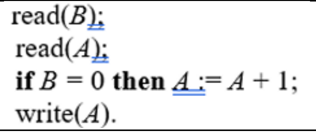
Let the consistency requirement be A = 0 V B = 0, with A = B = 0 as the initial values. Show that every serial execution involving these two transactions preserves the consistency of the database.

Ans: Each transaction modifies only one variable (either A or B) based on the other variable's value. Initially, A=0 and B=0, so both conditions (A=0 and B=0) are true.

T_{13} → T_{14} (T_{13} followed by T_{14}):

T_{13}	T_{14}	A	B
		0	**0**

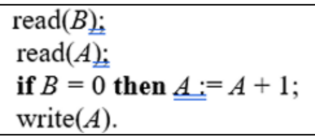
 read(A);
read(B);
if $A = 0$ then $B := B + 1$;
write(B).
			1

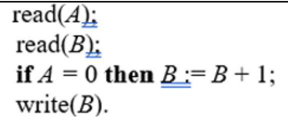
	 read(B);
	read(A);
	if $B = 0$ then $A := A + 1$;
	write(A).
		0	

- T_{13}: Reads A=0, updates B=B+1=1.
- T_{14}: Reads B=1, does not update A (condition B=0 is false).
- Final state: A=0, B=1. The consistency requirement A=0 ∨ B=0 holds.

T_{14} → T_{13} (T_{14} followed by T_{13}):

T_{13}	**T_{14}**	**A**	**B**
		0	**0**

	
	read(B);
	read(A);
	if $B = 0$ **then** $A := A + 1$;
	write(A).
		1	


read(A);
read(B);
if $A = 0$ **then** $B := B + 1$;
write(B).
			0

- T_{14}: Reads B=0, updates A=A+1 =1.
- T_{13}: Reads A=1, does not update B (condition A=0 is false).
- Final state: A=1, B=0. The consistency requirement A=0 ∨ B=0holds.

Every serial execution involving T_{13} and T_{14} preserves the consistency of the database (A=0∨B=0).

Problem 2

For each of the following schedules draw a precedence graph for the schedules. Is the schedule conflict serializable? If so, what are all the equivalent serial schedules:

a) r1(A); r2(A), r3(B),w1(A);r2(C),r2(B),w2(B),w1(C)

b) w3(A),r1(A),w1(B),r2(B),w2(C),r3(C)

Ans:

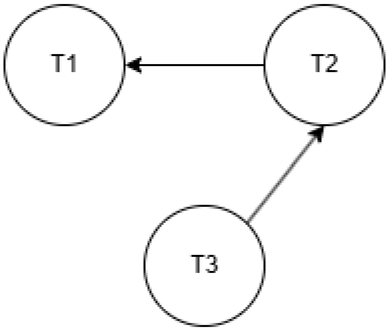
a) r1(A); r2(A), r3(B),w1(A);r2(C),r2(B),w2(B),w1(C)

T1	**T2**	**T3**
r(A)		
	r(A)	
		r(B)
w(A)		
	r(C)	
	r(B)	
	w(B)	
w(C)		

Conflicts:

- r2(A), w1(A): Read- write conflict. Edge from T2 to T1
- r3(B), w2(B): conflicting. Edge from T3 to T2
- r2(C), w1(C): conflicting. Edge from T2 to T1

Precedence graph:



Conflict Serializable: Since the precedence graph is acyclic, the schedule conflict serializable.

Equivalent serial schedule:

- T1 → T2 → T3: r1(A), w1(A), w1(C), r2(A), r2(C), r2(B), w2(B), r3(B)

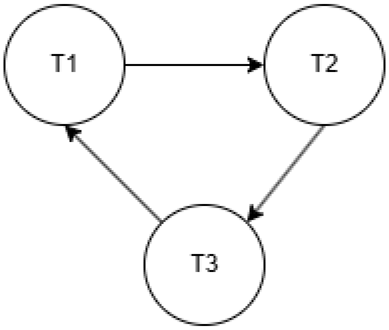
b) w3(A),r1(A),w1(B),r2(B),w2(C),r3(C)

T1	**T2**	**T3**
		w(A)
r(A)		
w(B)		
	r(B)	
	w(C)	
		r(C)

Conflicts:

- w3(A), r1(A): Conflict. Edge from T3 to T1.
- w1(B), r2(B): Conflict. Edge from T1 to T2.
- W2(C), r(C): Conflict. Edge from T2 to T3.

Precedence graph:



Conflict Serializable: Since the precedence graph is not acyclic, the schedule is not conflict serializable.

Equivalent schedules: There are no equivalent serial schedules since it is not conflict serializable.