Part 2: Conflict Serializability (20 points)

Consider the following three transactions and schedule (time goes from top to bottom). Is this schedule conflict-serializable? Show why or why not.

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

The above transactions and schedule can be written in “in-line” form as follows:

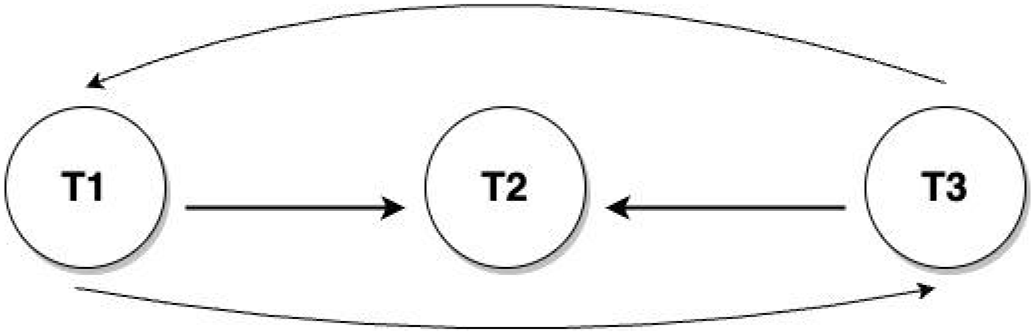
R1(A), W1(A), R3(A), W3(A), R2(A), R1(B), R3(B), W1(B), W3(B), R2(B), C2, C1, C3

The following list contains all conflicts that this schedule creates:

- Conflict 1: R1(A) to W3(A),
- Conflict 2: W1(A) to R3(A),
- Conflict 3: W1(A) to W3(A),
- Conflict 4: W1(A) to R2(A),

- Conflict 5: W3(A) to R2(A),
- Conflict 6: R1(B) to W3(B),
- Conflict 7: R3(B) to W1(B),
- Conflict 8: W1(B) to W3(B),
- Conflict 9: W1(B) to R2(B),
- Conflict 10: W3(B) to R2(B)

As seen above, there are conflicts from Transaction 1 to Transaction 2, Transaction 1 to Transaction 3, Translation 3 to Transaction 2, and Transaction 3 to Transaction 1. Hence the following precedence graph is shown below.



As seen above, since there is a cycle in the precedence graph, this schedule is not conflict serializable.