

Part 1: Schedules and Anomalies

Consider a database with objects X, Y, and Z and assume that there are two transactions T1 and T2 that attempt the following operations.

T1: R(X), R(Y), W(X)

T2: R(X), R(Y), W(Y), R(X), R(Y), W(X), R(Z), W(Z)

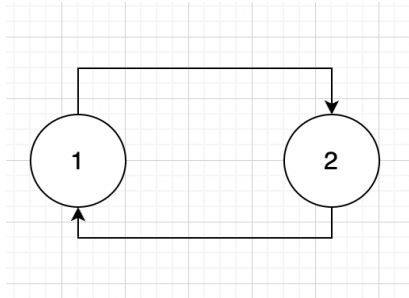
A) Write an example schedule that interleaves operations between T1 and T2, that is NOT conflict serializable.

T1	T2
R(X)	
	R(X)
R(Y)	
	R(Y)
	W(Y)
	R(X)
	R(Y)
W(X)	
	W(X)
	R(Z)
	W(Z)

As above, I have the schedule as below:

$R_1(X); R_2(X); R_1(Y); R_2(Y); W_2(Y); R_2(X); R_2(Y); W_1(X); W_2(X); R_2(Z); W_2(Z);$

We can find the a cycle of edge in the graph as below:



Since there is a cycle, this is not a conflict serializable schedule.

B) If T1 is instead just “R(X)”, this corresponds to T1 just being a single query like

```
SELECT * FROM Flights WHERE id=1024;
```

Should the database treat a single SQL statement like this as a transaction? Why or why not?

Yes, the database should treat this single SQL statement as a transaction. According to the property and definition of transaction, the sequence of element reads and/or writes can be defined as a transaction. This case is a single statement to read data, and it satisfies the “ACID” semantics.