Tianyu Zhang
CS143
HW6

A.

1. Since T2 → T3 on C and T3 → T2 on A. There is a directed cycle T3 ← →T2 so the schedule is not confict-serializable.

2.(i) complete:

1. T2 sets a lockX on C just before write C, and it must keep until read A. So when T1 sets lockS on C it stops wait-for T1 → T2. Similarly for T3 wait-for T3 → T2. There is no cycle. First T2 will complete and T3 and T1 will complete.

3.

no deadlock, T1 will wait for previous transaction to release C. and T3 is younger than T2; so when it requests C it will die. So it will be T2-T1-T3.

4.

we have T2 → T3 on C and T3 → T2 on A. It is not confict-serializable.

5.

With no deadlock stragety,

1. T2 sets a lockX for C just before write C
2. T1 does lockS for C it wait-for T1 → T2.
3. T3 does lockX for A, and then by requesting lock-2 for C it wait-for arc T3 → T2.
4. T2 does a lockS for A and waitfor T2 → T3.

Then there will be a deadlock

6.

(i) complete

T1	T2	T3
LockX for D		
write D		
	LockX for C	
	write C	
	roll back T2	
lockS for c		
read c		
		lockX for a
		write a
		wait for c
		roll back T3
lockx for a		
write a		
unlock D		
unlock C		
unlock A		

B.

1.

No, there is a intersection between different read write object.

2.

yes. An equivalent schedule is: w3(A)c3r1(A)w1(B)c1r2(B)w2(C)c2r4(B)c4

3.

yes. w3(a) is before r1(a) and c3 is before c1. w1(b) is before r2(b) and c1 is before c2

4.

No. move c3 to second position.