Database Systems (CPH) – Fall 2025
Teacher: Tiantian Liu
Exercise Sheet: Lecture 10: Transactions and Concurrency Control

1. Review the following terms.

- Transaction
- ACID properties
  - Atomicity
  - Consistency
  - Isolation
  - Durability
- Serial schedules
- Conflict serializable
- Conflict/Precedence graph
- Locking/Unlocking
- Two-Phase locking
- Shared lock
- Exclusive lock
- Wait-for graph
- Deadlock

Answer:

Review the slides or the textbook (Chapters 17 and 18).

2. Consider the following schedules:

1)

T_1:	W(A);				W(C);		commit.	
T_2:		W(B);				R(C);		commit.
T_3:			R(B);	commit.				

2)

T_1:	R(A);		W(A);	R(C);		commit.						
T_2:		R(C);						W(A);		W(B);	commit.	
T_3:					R(A);		W(C);		W(B);			commit.

3)

T_1:	R(A);	W(A);					W(B);	W(C);					commit.
T_2:				R(A);					R(B);	W(B);	W(C);	commit.	
T_3:			R(C);		W(C);	commit.							

For each of these schedules, please create a precedence graph and decide if the schedule is conflict serializable. If it is conflict serializable, give one example of a conflict equivalent serial schedule?

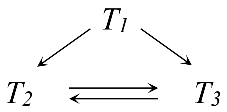
Answers:

1) The precedence graph is:

$T_1 \rightarrow T_2 \rightarrow T_3$

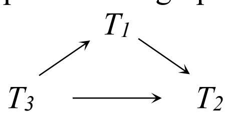
The graph has no cycle. The schedule is conflict serializable.
Equivalent serial schedule: T_1, T_2, T_3.

2) The precedence graph is:



There is a cycle between T_2 and T_3. The schedule is therefore not conflict serializable.

3) The precedence graph is:



The graph has no cycle. The schedule is conflict serializable.
Equivalent serial schedule: T_3, T_1, T_2.

3. Consider the following two transactions:

```
T1: R(A);
    R(B);
    if $A = 0$ then $B := B + 1$;
    W(B).

T2: R(B);
    R(A);
    if $B = 0$ then $A := A + 1$;
    W(A).
```

Add lock and unlock instructions to transactions T_1 and T_2 so that they observe the two-phase locking protocol. Can the execution of these transactions result in a deadlock?

Answers:

Lock and unlock instructions:

```
T1: Lock-S(A);
    R(A);
    Lock-X(B)
    R(B);
    if $A = 0$ then $B := B + 1$;
    W(B);
    Unlock(A);
    Unlock(B).

T2: Lock-S(B);
    R(B);
    Lock-X(A);
    R(A)
    if $B = 0$ then $A := A + 1$;
    W(A);
    Unlock(B);
    Unlock(A).
```

Exeution of these transations may result in deadlock. For example, consider the following partial schedule:

T_1:	Lock-S(A);		Lock-X(B);	
T_2:		Lock-S(B);		Lock-X(A);