CONFLICT AND VIEW SERIALIZABILITY

SERIALIZABILITY

- A schedule is serializable if it is **equivalent to a serial schedule**.
- In **serial schedules**, only **one transaction is allowed to execute at a time** i.e. **no concurrency is allowed**.
- Whereas in **serializable schedules**, **multiple transactions can execute simultaneously** i.e. **concurrency is allowed**.
- Types (forms) of serializability
  - Conflict serializability
  - View serializability

SIMPLIFIED VIEW OF TRANSACTIONS

We ignore operations other than **read** and **write** instructions

We assume that transactions may perform arbitrary computations on data in local buffers in between reads and writes.

Our simplified schedules consist of only **read** and **write** instructions.

CONFLICTING INSTRUCTIONS

Instructions I_i and I_j of transactions T_i and T_j respectively, **conflict** if and only if there exists some item Q accessed by both I_i and I_j, and at least one of these instructions wrote Q.

1. I_i = **read**(Q), I_j = **read**(Q). I_i and I_j don't conflict.
2. I_i = **read**(Q), I_j = **write**(Q). They conflict.
3. I_i = **write**(Q), I_j = **read**(Q). They conflict
4. I_i = **write**(Q), I_j = **write**(Q). They conflict

Intuitively, a conflict between I_i and I_j forces a (logical) temporal order between them.

If I_i and I_j are consecutive in a schedule and they do not conflict, their results would remain the same even if they had been interchanged in the schedule.

CONFLICT SERIALIZABILITY

If a schedule *S* can be transformed into a schedule *S'* by a series of swaps of non-conflicting instructions, we say that *S* and *S'* are **conflict equivalent**.

We say that a schedule *S* is **conflict serializable** if it is conflict equivalent to a serial schedule

CONFLICT SERIALIZABILITY (CONT.)

Schedule 3 can be transformed into Schedule 6, a serial schedule where T_2 follows T_1, by series of swaps of non-conflicting instructions. Therefore Schedule 3 is conflict serializable.

T_1	T_2
read (A)	
write (A)	
	read (A)
	write (A)
read (B)	
write (B)	
	read (B)
	write (B)

Schedule 3

T_1	T_2
read (A)	
write (A)	
read (B)	
write (B)	
	read (A)
	write (A)
	read (B)
	write (B)

Schedule 6

CONFLICT SERIALIZABILITY (CONT.)

Example of a schedule that is not conflict serializable:

T_3	T_4
read (Q)	
	write (Q)
write (Q)	

We are unable to swap instructions in the above schedule to obtain either the serial schedule $< T_3, T_4 >$, or the serial schedule $< T_4, T_3 >$.

TESTING FOR SERIALIZABILITY

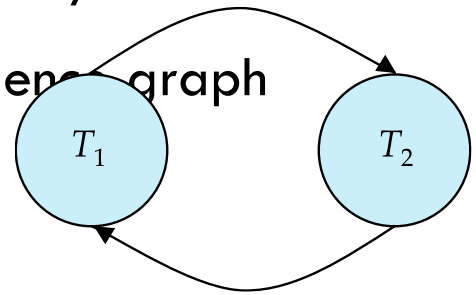
Consider some schedule of a set of transactions T_1, T_2, ..., T_n

Precedence graph — a direct graph where the vertices are the transactions (names).

We draw an arc from T_i to T_j if the two transaction conflict, and T_i accessed the data item on which the conflict arose earlier.

We may label the arc by the item that was accessed.

Example of a precedence graph



TEST FOR CONFLICT SERIALIZABILITY

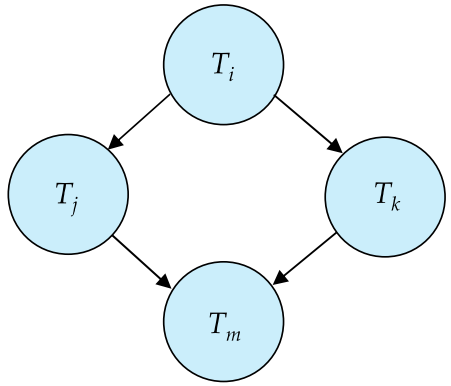
A schedule is conflict serializable if and only if its precedence graph is acyclic.

Cycle-detection algorithms exist which take order n^2 time, where n is the number of vertices in the graph.

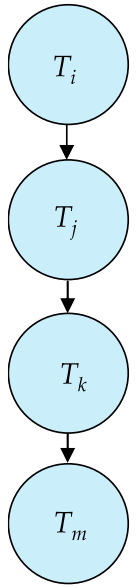
- (Better algorithms take order n + e where e is the number of edges.)

If precedence graph is acyclic, the serializability order can be obtained by a *topological sorting* of the graph.

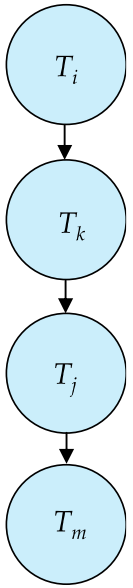
- This is a linear order consistent with the partial order of the graph.
- For example, a serializability order for Schedule A would be
  $T_5 \rightarrow T_1 \rightarrow T_3 \rightarrow T_2 \rightarrow T_4$
  - Are there others?



(a)



(b)



(c)

VIEW SERIALIZABILITY

- Let S1 and S2 be two schedules with the same set of transactions. S1 and S2 are view equivalent if the following three conditions are satisfied, for each data item Q
  - Initial Read
  - Updated Read
  - Final Write
- If a schedule is view equivalent to its serial schedule then the given schedule is said to be view serializable.

INITIAL READ

- If in schedule S1, transaction Ti reads the initial value of Q, then in schedule S2 also transaction Ti must read the initial value of Q.

S1	
T1	T2
Read (A)	
	Write (A)

S3	
T1	T2
	Read (A)
Write (A)	

S2	
T1	T2
	Write (A)
Read (A)	

- Above two schedules S1 and S3 are not view equivalent because initial read operation in S1 is done by T1 and in S3 it is done by T2.
- Above two schedules S1 and S2 are view equivalent because initial read operation in S1 is done by T1 and in S2 it is also done by T1.

UPDATED READ

- If in schedule S1 transaction Ti executes read(Q), and that value was produced by transaction Tj (if any), then in schedule S2 also transaction Ti must read the value of Q that was produced by transaction Tj.

S1		
T1	T2	T3
Write (A)		
	Write (A)	
		Read (A)

S3		
T1	T2	T3
Write (A)		
	Write (A)	
		Read (A)

S2		
T1	T2	T3
Write (A)		
	Write (A)	
		Read (A)

- Above two schedules S1 and S3 are not view equal because, in S1, T3 is reading A that is updated by T2 and in S3, T3 is reading A which is updated by T1.
- Above two schedules S1 and S2 are view equal because, in S1, T3 is reading A that is updated by T2 and in S2 also, T3 is reading A which is updated by T2.

FINAL WRITE

- If Ti performs the final write on the data value in S1, then it also performs the final write on the data value in S2.

S1		
T1	**T2**	**T3**
Write (A)		
	Read (A)	
		Write (A)

S3		
T1	**T2**	**T3**
Write (A)		
	Write (A)	
		Read (A)

S2		
T1	**T2**	**T3**
Write (A)		
	Read (A)	
		Write (A)

- Above two schedules S1 and S3 are not view equal because final write operation in S1 is done by T3 and in S3 final write operation is also done by T1.
- Above two schedules S1 and S2 are view equal because final write operation in S1 is done by T3 and in S2 also the final write operation is also done by T3.

VIEW SERIALIZABLE EXAMPLE

- If a schedule is view equivalent to its serial schedule then the given schedule is said to be view serializable.

Non-Serial Schedule (S1)	
T1	**T2**
Read (A)	
Write (A)	
	Read (A)
	Write (A)
Read (B)	
Write (B)	
	Read (B)
	Write (B)

Serial Schedule (S2)	
T1	**T2**
Read (A)	
Write (A)	
Read (B)	
Write (B)	
	Read (A)
	Write (A)
	Read (B)
	Write (B)

- **S2 is the serial schedule of S1**. If we can **prove that they are view equivalent** then we can says that **given schedule S1 is view serializable**.

VIEW SERIALIZABLE EXAMPLE (INITIAL READ)

Non-Serial Schedule (S1)	
T1	**T2**
Read (A)	
Write (A)	
	Read (A)
	Write (A)
Read (B)	
Write (B)	
	Read (B)
	Write (B)

Serial Schedule (S2)	
T1	**T2**
Read (A)	
Write (A)	
Read (B)	
Write (B)	
	Read (A)
	Write (A)
	Read (B)
	Write (B)

- In schedule S1, transaction T1 first reads the data item X. In S2 also transaction T1 first reads the data item X.
- In schedule S1, transaction T1 first reads the data item Y. In S2 also the first read operation on Y is performed by T1.
- The initial read condition is satisfied for both the schedules

VIEW SERIALIZABLE EXAMPLE (UPDATED READ)

Non-Serial Schedule (S1)	
T1	**T2**
Read (A)	
Write (A)	
	Read (A)
	Write (A)
Read (B)	
Write (B)	
	Read (B)
	Write (B)

Serial Schedule (S2)	
T1	**T2**
Read (A)	
Write (A)	
Read (B)	
Write (B)	
	Read (A)
	Write (A)
	Read (B)
	Write (B)

- In schedule S1, transaction T2 reads the value of X, written by T1. In S2, the same transaction T2 reads the X after it is written by T1.
- In schedule S1, transaction T2 reads the value of Y, written by T1. In S2, the same transaction T2 reads the value of Y after it is updated by T1.
- The updated read condition is also satisfied for both the schedules.

VIEW SERIALIZABLE EXAMPLE (FINAL WRITE)

Non-Serial Schedule (S1)	
T1	**T2**
Read (A)	
Write (A)	
	Read (A)
	Write (A)
Read (B)	
Write (B)	
	Read (B)
	Write (B)

Serial Schedule (S2)	
T1	**T2**
Read (A)	
Write (A)	
Read (B)	
Write (B)	
	Read (A)
	Write (A)
	Read (B)
	Write (B)

- In schedule S1, the final write operation on X is done by transaction T2. In S2 also transaction T2 performs the final write on X.
- In schedule S1, the final write operation on Y is done by transaction T2. In schedule S2, final write on Y is done by T2.
- The final write condition is also satisfied for both the schedules.

VIEW SERIALIZABLE EXAMPLE

Non-Serial Schedule (S1)	
T1	**T2**
Read (A)	
Write (A)	
	Read (A)
	Write (A)
Read (B)	
Write (B)	
	Read (B)
	Write (B)

Serial Schedule (S2)	
T1	**T2**
Read (A)	
Write (A)	
Read (B)	
Write (B)	
	Read (A)
	Write (A)
	Read (B)
	Write (B)

- Since **all the three conditions** that checks whether the two schedules are view equivalent **are satisfied** in this example, which means **S1 and S2 are view equivalent**.
- Also, as we know that the **schedule S2 is the serial schedule of S1**, thus we can say that the **schedule S1 is view serializable schedule**.