

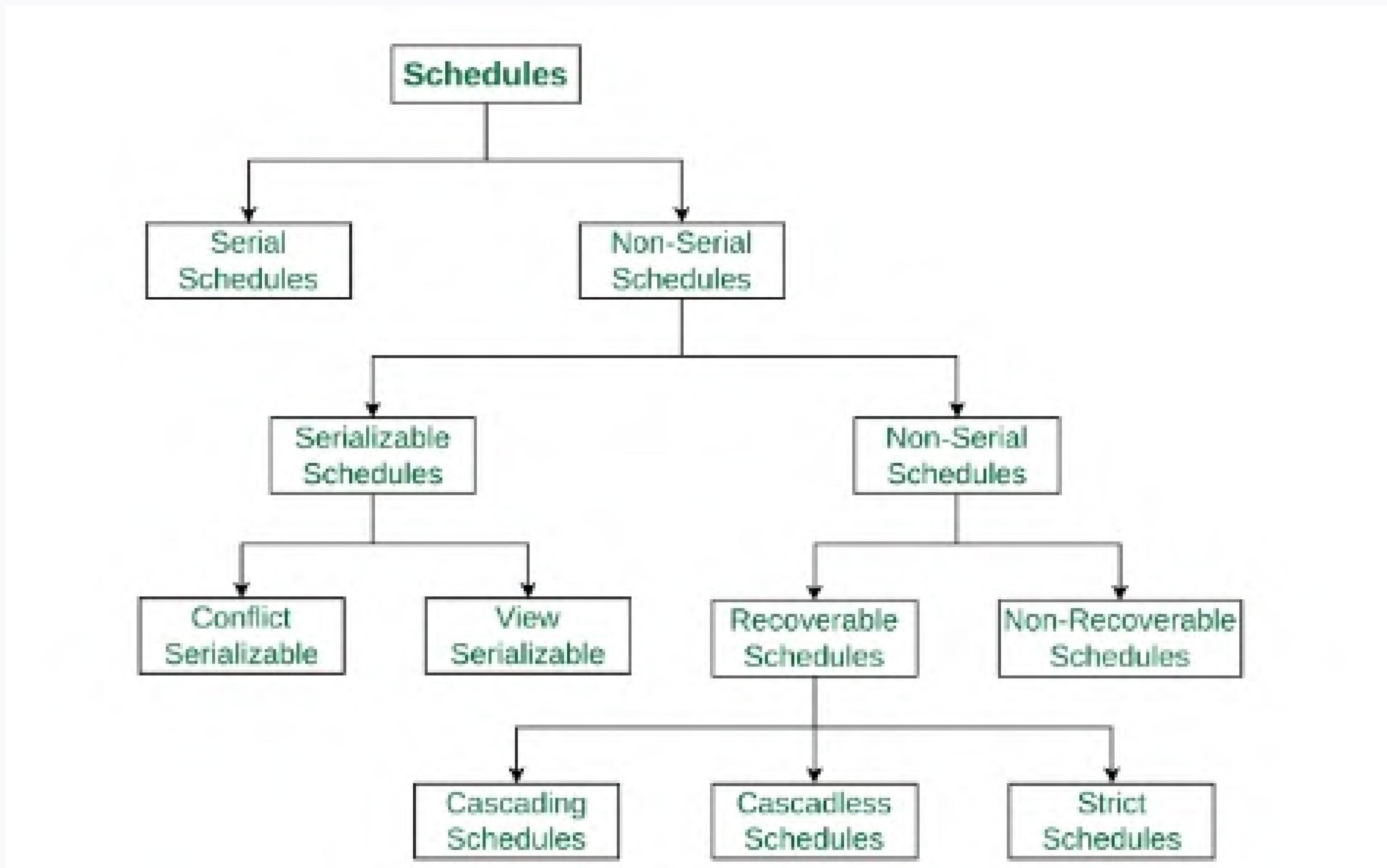
SERIALIZATION

BY ATTIF KHAN

What is Schedule?

A series of operation from one transaction to another transaction is known as schedule

Types of Schedules



Serial Schedule :-

The serial schedule is a type of schedule where one transaction is executed completely before starting another transaction. In the serial schedule, when the first transaction completes its cycle, then the next transaction is executed.

Non Serial Schedule :-

This is a type of Scheduling where the operations of multiple transactions are interleaved.

- **Serializable Schedules**
 1. Conflict Serializable
 2. View Serializable

Serializable Schedule :-

This is used to maintain the consistency of the database. It is mainly used in the Non-Serial scheduling to verify whether the scheduling will lead to any inconsistency or not.

1. Conflict serializable :-

A schedule is called conflict serializable if it can be transformed into a serial schedule

2. View serializable:-

A Schedule is called view serializable if its view is equal to a serial schedule (no overlapping transactions).

Conflict Serializable:-

Conflict serializable can be done in many ways but only two methods are important as per the Gate point of view.

- 1. Testing Method (Precedence graph)**
- 2. Conflict Equivalence**

What are conflict & non-conflict instructions?

Conflict Instruction:

Generally, in concurrent execution, for the same data item, if the I/O operations belong to two different transactions such that the following combinations of instructions are adjacent from different transactions like

$R(A)$	$W(A)$
$W(A)$	$R(A)$
$W(A)$	$W(A)$

Non-Conflict Instruction:

A non conflict instruction is one in which read and write operation is valid in adjacent transactions when if and only if both have different data or both the operations are same like

$R(A)$	$W(B)$
$R(A)$	$R(A)$

Conflict Serializability:-

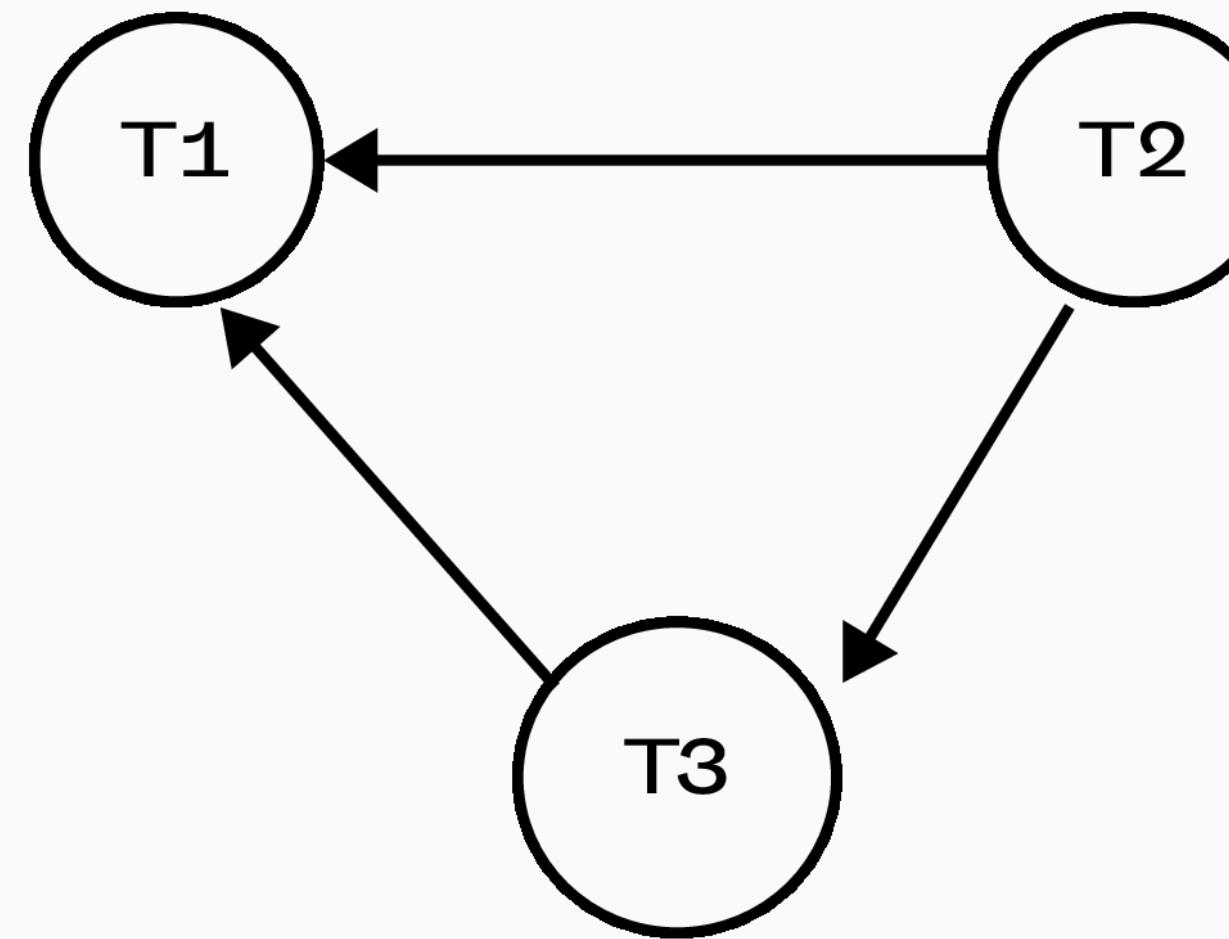
1. Testing method (Precedence graph)

T1	T2	T3
R(X)		
		R(Y)
		R(X)
	R(Y)	
	R(Z)	
		W(Y)
		W(Z)
R(Z)		
W(X)		
W(Z)		

- Take 3 Transactions states to make a graph.
- Check conflict pairs in other transactions and draw edges.

Precedence Graph:

- The edges in the precedence graph are directed edges.



- Now check the precedence graph whether it have any cycle/loop if any then it is not conflict serializable but if it do not contain cycle/loop then it is conflict serializable and we can proceed to Topological sorting.
- In serial schedule the number of possibilities of execution depend on the number of transactions

$$n \text{ transactions} = n !$$

We have 3 transactions so $3 ! = 3 \times 2 \times 1 = 6$ possibilities

$T_1 - T_2 - T_3$

$T_1 - T_3 - T_2$

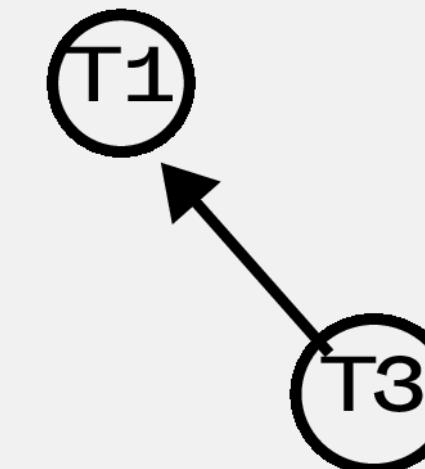
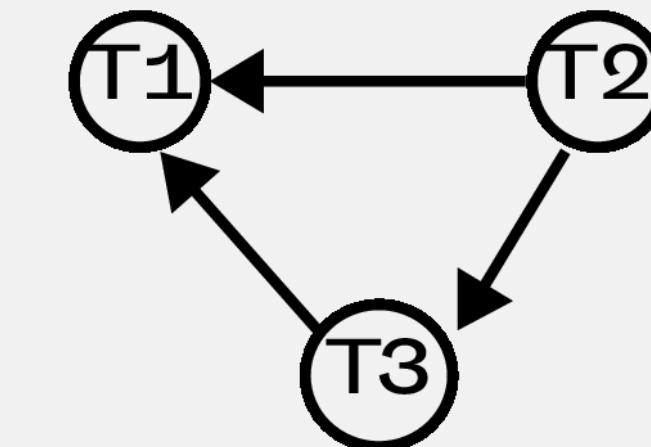
$T_2 - T_1 - T_3$

$T_2 - T_3 - T_1$

$T_3 - T_1 - T_2$

$T_3 - T_2 - T_1$

- Now by using topological sorting we will find the series of execution of the transactions
- Remove the state who have Indegree = 0
- T2 has Indegree 0 so remove T2 along with its edges.
- Now, T3 has Indegree 0 so remove it.
- Now after doing Topological sorting we will get the series of execution of the transactions
- T2 – T3 – T1



T1

Conflict Equivalence:

Two schedules are said to be conflict equivalent, if all conflicting operation in both the schedule must be executing in the same order.

EX1:

$S_1 : R_1(X) W_1(X) R_2(X) W_2(X) R_1(Y)$

$S_2 : R_1(X) W_1(X) R_1(X) R_2(Y) W_2(Y)$

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

conflicting pairs : $W(Y) - R(Y)$

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

conflicting pairs : $R(Y) - W(Y)$

- Both this schedules are not conflict equivalent because the conflicting pairs in both the schedules is different.

EX2:

S1 : R1(A) W1(A) R2(A) W2(A) R1(B) W1(B)

S2 : R1(A) W1(A) R2(A) R1(B) W2(A) W1(B)

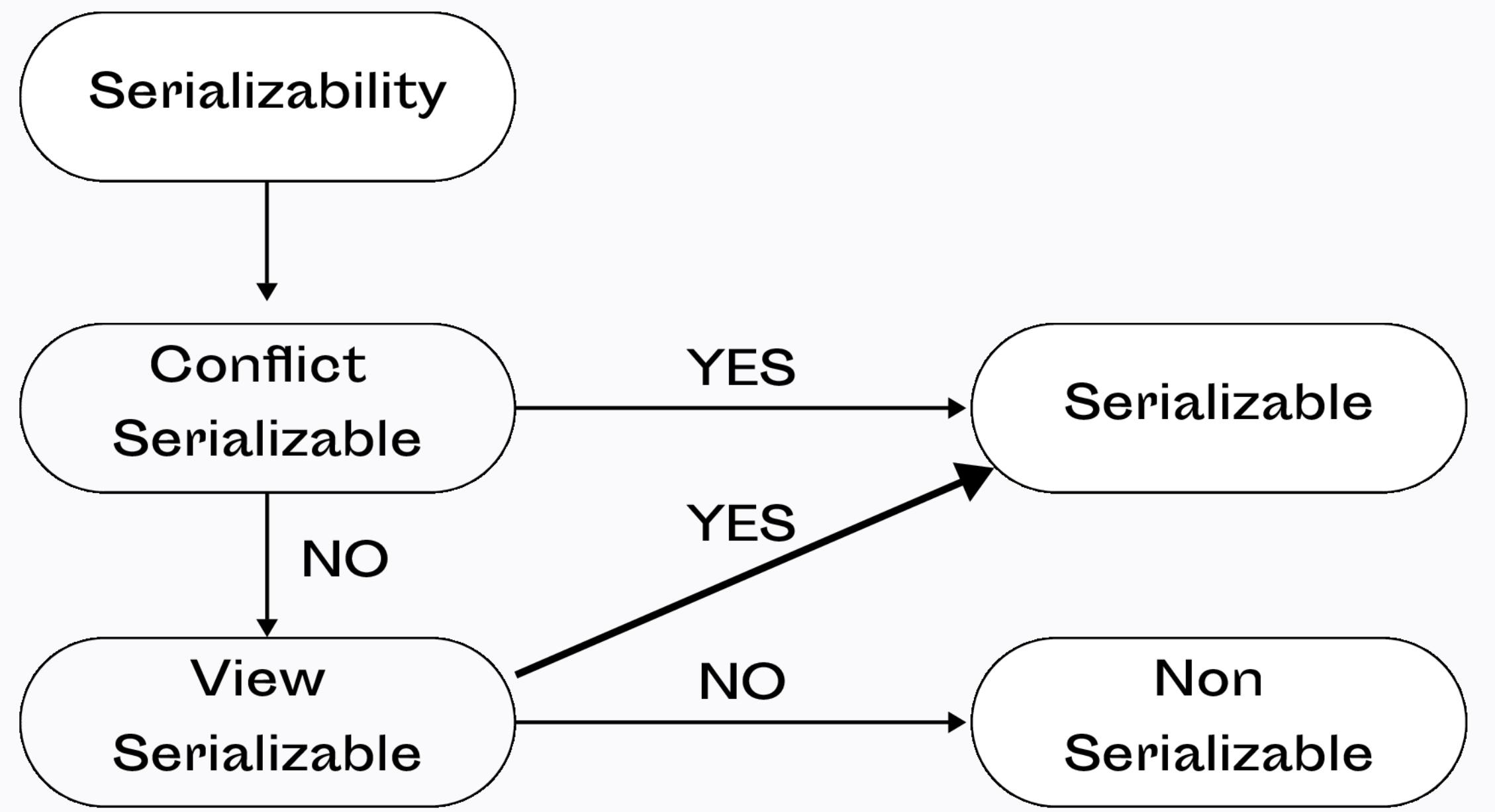
T1	T2
R(A)	
W(A)	
	R(A)
	W(A)
R(B)	
W(B)	

Conflicting pairs: R(A) – W(A)
W(A) – R(A)
W(A) – W(A)

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

Conflicting pairs: R(A) – W(A)
W(A) – R(A)
W(A) – W(A)

- The given two schedules are Conflict Equivalent, because both have same conflicting pairs.



View Serializable:

A schedule is said to be view serializable, if it has an equivalent 'view equivalent schedule'

View Equivalent Schedule:

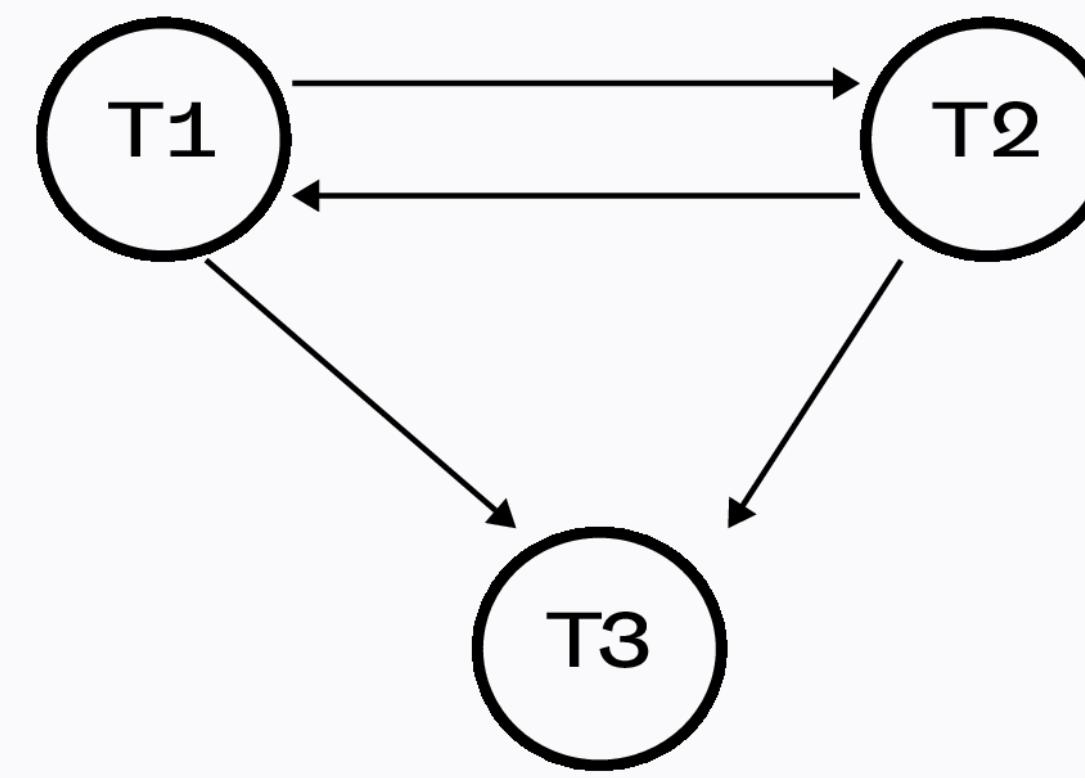
A Schedule S' is said to be a view equivalent schedule to S if it satisfies the following 3 conditions:

1. Initial Read
2. Final Write
3. Updated Read

Example:

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

Precedence Graph:



- This graph contains loop/cycle hence it is not conflict serializable

- Now we should check for view serializable by using those 3 conditions.
- We have 3 Transactions so the number of possibilities are $3! = 6$ possibilities
- Now we have to check all 6 possibilities / schedule.
- To avoid checking all the 6 and saving the time we have a technique called Dummy Technique.
- Dummy technique: In this technique we check which transaction is performing the initial read and which transaction is performing final update

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

$\times \{ T_1 - T_2 - T_3$
 $T_1 - T_3 - T_2$
 $T_2 - T_1 - T_3 \checkmark$
 $T_2 - T_3 - T_1$
 $\times \{ T_3 - T_1 - T_2$
 $T_3 - T_2 - T_1$

Dummy Assumption: T2 – T1 – T3

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

1. Initial Read : A : T2
B : T2
2. Final Write: A : T3
B : T1
3. Updated Read: T1 – T3

Given Schedule:

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

1. Initial Read : A : T2
B : T2
2. Final Write: A : T3
B : T1
3. Updated Read: T1 – T3

Consider the following schedules involving two transactions.
Which one of the following statements is TRUE?

S_1 : $r_1(X); r_1(Y); r_2(X); r_2(Y); w_2(Y); w_1(X)$

S_2 : $r_1(X); r_2(X); r_2(Y); W_2(Y); r_1(Y); w_1(X)$

[2007: 2 Marks]

Both S_1 and S_2 are conflict serializable

S_1 is conflict serializable and S_2 is not conflict serializable

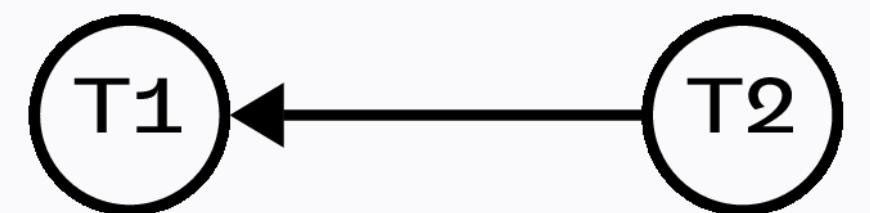
S_1 is not conflict serializable and S_2 is conflict serializable

Both S_1 and S_2 are not conflict serializable

Solution:

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

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



Option C: S1 is not conflict serializable and S2 is conflict serializable

**THANK
YOU!**

“NOW OR NEVER....!!”