S_1 & S_2 are Conflict equal iff

(1) each transaction of S_1 must be exactly same transacting in S_2

(2) Every conflict pair precedance of S_1 must be exactly same precedence in S_2.

Conflict serializable schedule :-

Schedule(S) is conflict serializable iff at least one serial schedule is conflict equal to (S).

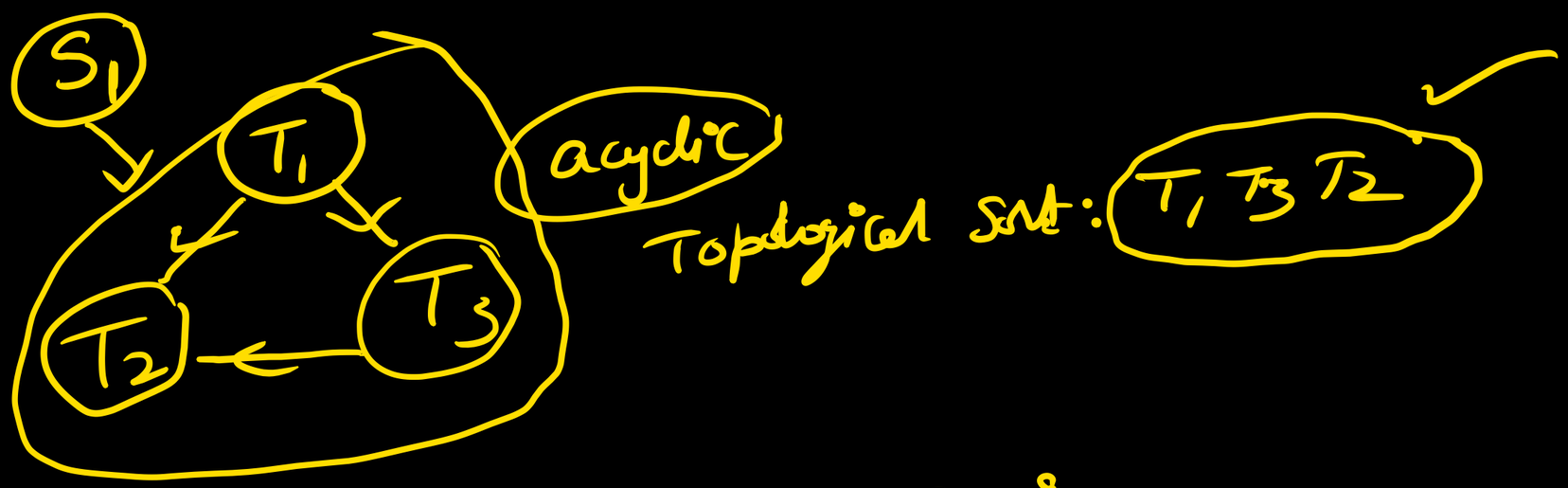
$T_1\ T_2\ T_3$ $\Longrightarrow$? Serial possible $\Rightarrow 3! = 6$.

$$\left\{\begin{array}{l} T_1\ T_2\ T_3 \\ T_1\ T_3\ T_2 \\ T_2\ T_1\ T_3 \\ T_2\ T_3\ T_1 \\ \vdots \end{array}\right\} \Longrightarrow$$

How to know a given schedule is conflict ~~serial~~ serializable

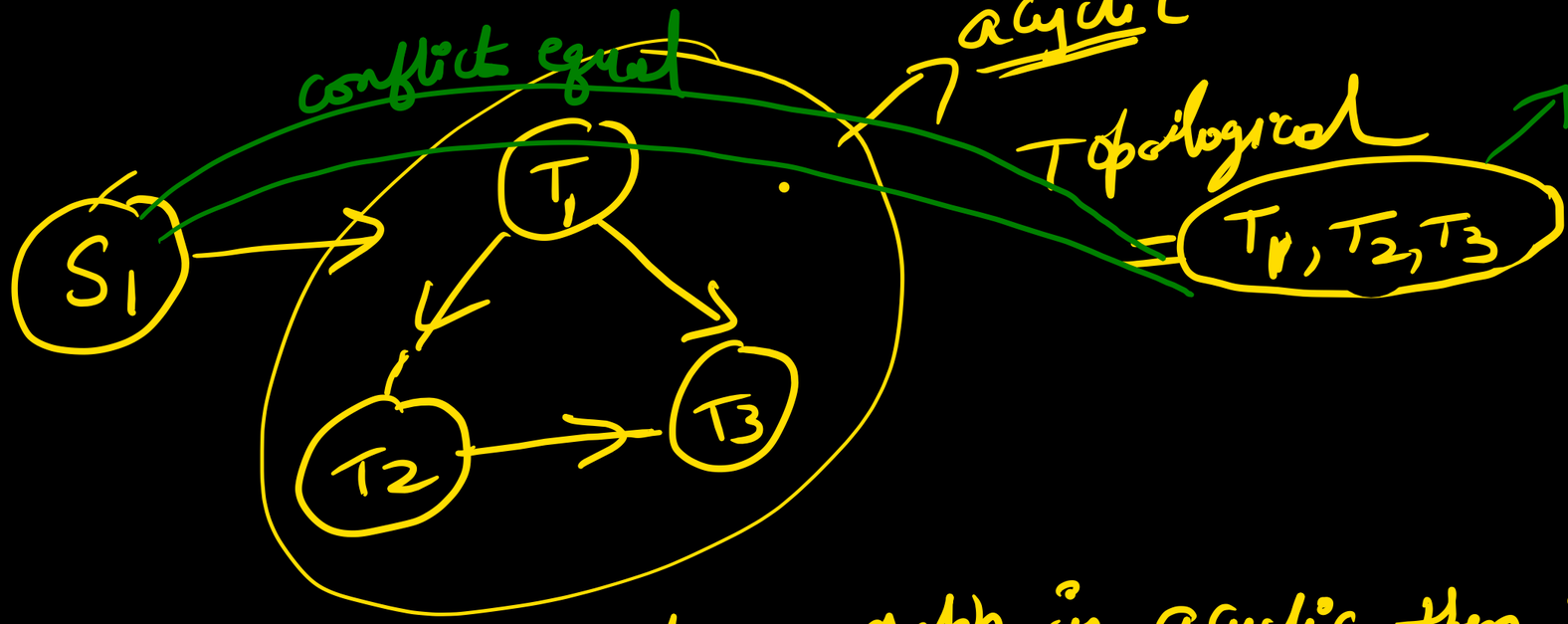
→ Serial schedules precedence graph is always acyclic.

→ If a schedule has acyclic precedence graph, then it is Conflict equal to some serial schedule and so Conflict serializable.



S_1 is Conflict serializable and it is Conflict equal to $T_1 T_3 T_2$

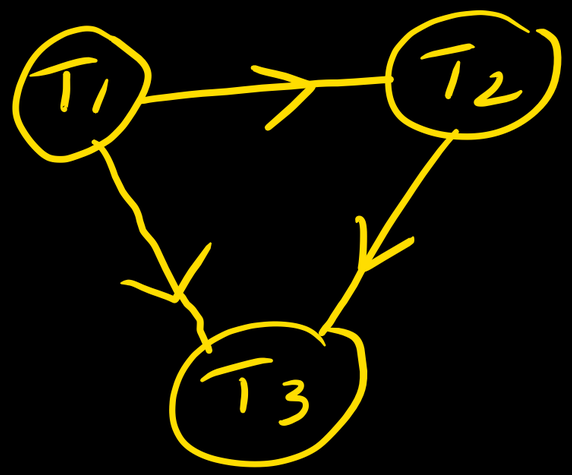
If a schedule has acyclic precedence graph, then it is serializable and it is conflict equal to the topological order of ~~Schedule~~ schedule's precedence graph.



If precedence graph is acyclic then it is conflict serializable

Conflict serializable or not:

(S): $r_2(A), r_1(B), w_2(A), w_3(A), w_2(B), w_1(C), w_3(C)$



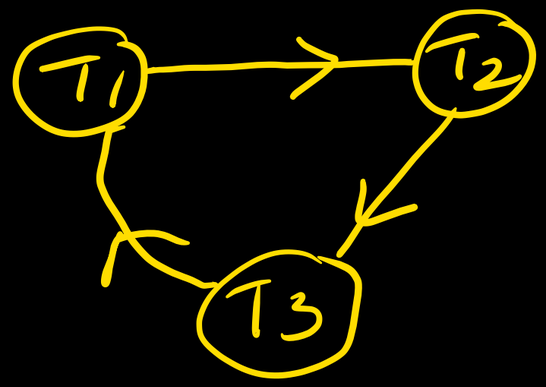
It is acyclic. so Conflict serializable

Topological sort: $T_1\ T_2\ T_3$

(S) is Conflict equal to serial schedule $T_1 T_2 T_3$
and so it is Conflict serializable

S: $r_1(A)$ $w_2(A)$ $w_2(B)$ $r_3(B)$ $w_3(C)$ $c_1(C)$

Conflict serializable ? not ?



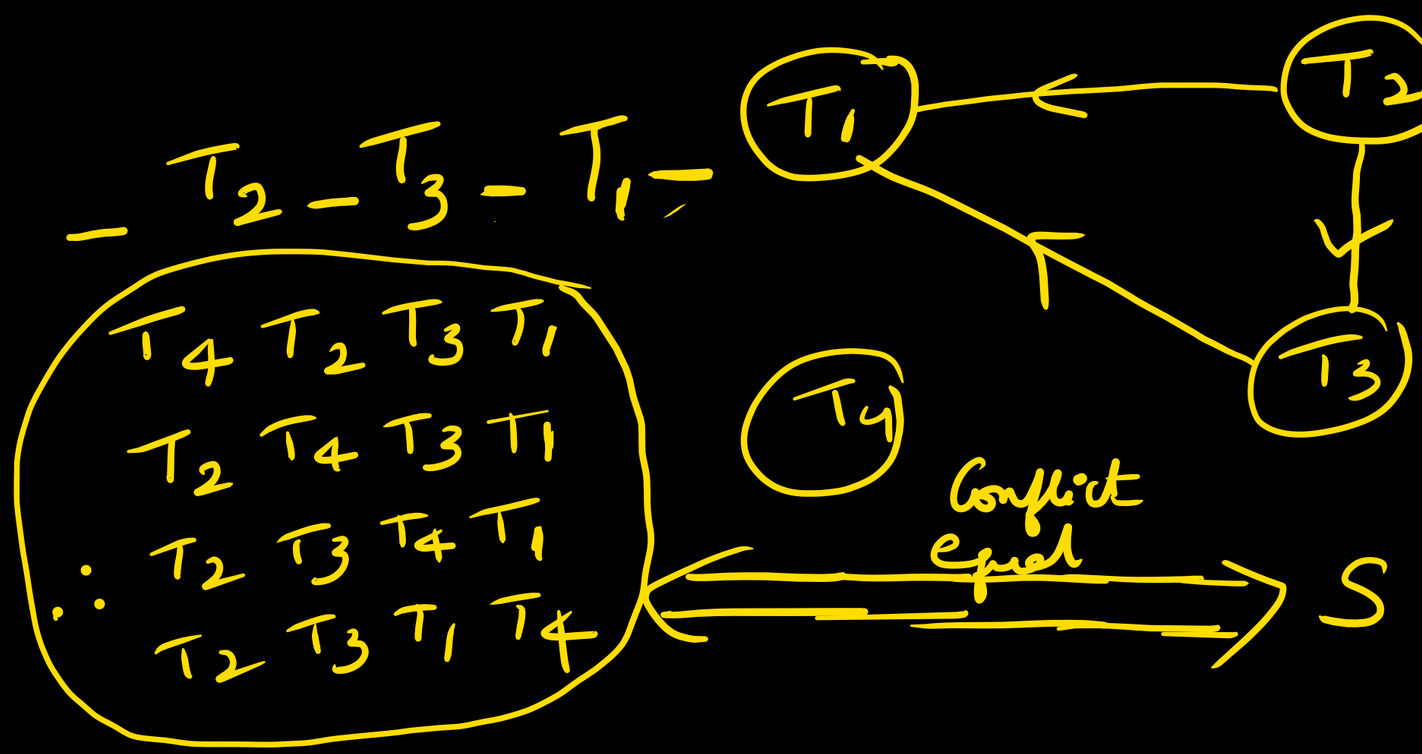
cyclic.

so not conflict serializable.

S: $r_1(A)\ r_2(A)\ r_3(A)\ r_4(A)\ w_2(B)\ w_3(B)\ w_1(B)$.

How many serial schedules are conflict equal to schedule S?

No write on (A). Therefore no of conflict on A.



View Serializability:

Schedule (S) is view serializable iff some serial schedule S' is view equal to schedule S.

Conflict Serializability: → Precedence graph → acyclic → Topological sort

Schedule (S) is CS iff there is some serial schedule that S' is conflict equal to schedule S.

n transactions → n! serial

View equal schedules:

Two schedules are **view equal** if the following are equal:

1) Initial reads
2) updated reads
3) Final writes

$S \overset{VE}{=} S'$

happens to be serial schedule

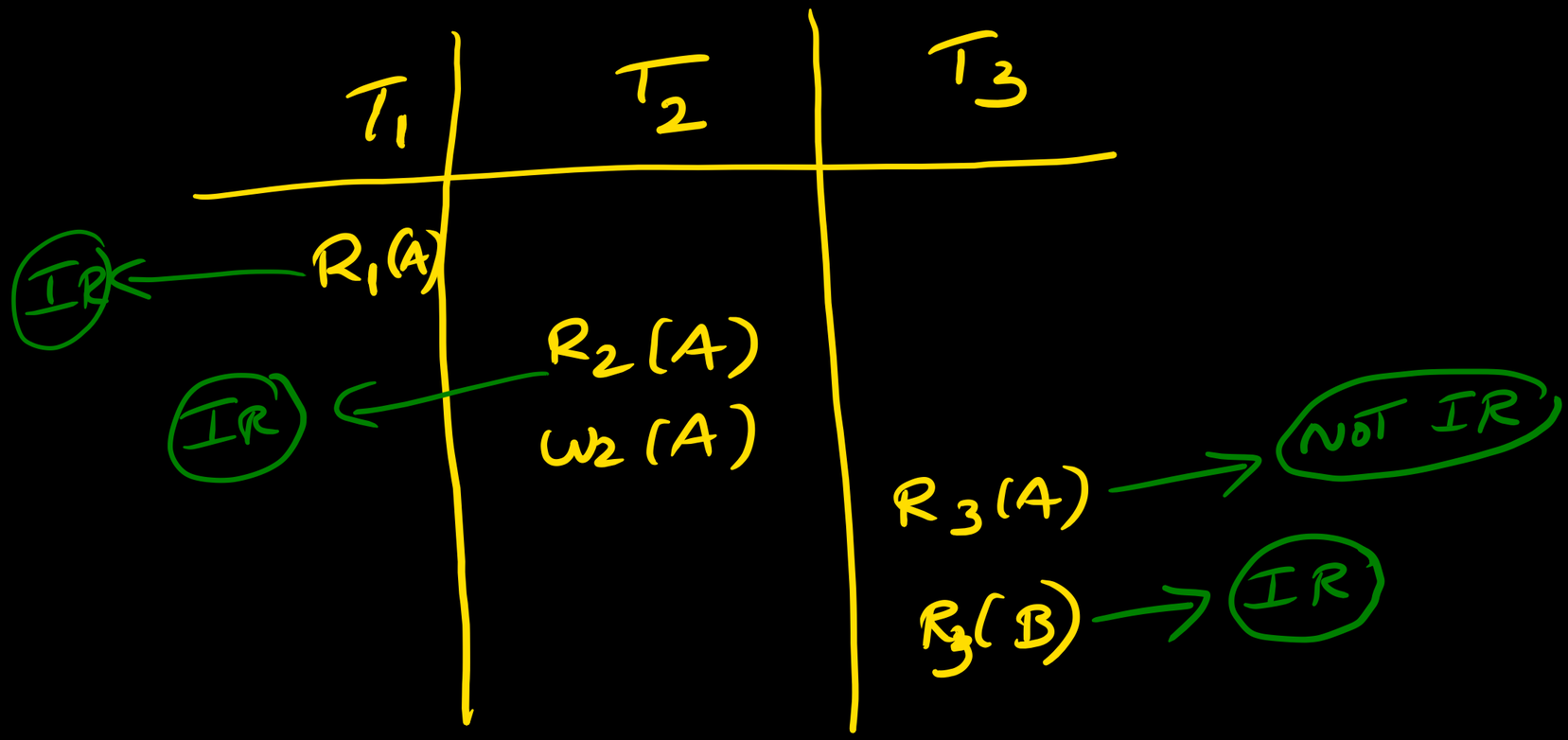
S is view serializable.

I View equal $(S_1\ S_2)$

II View serializability

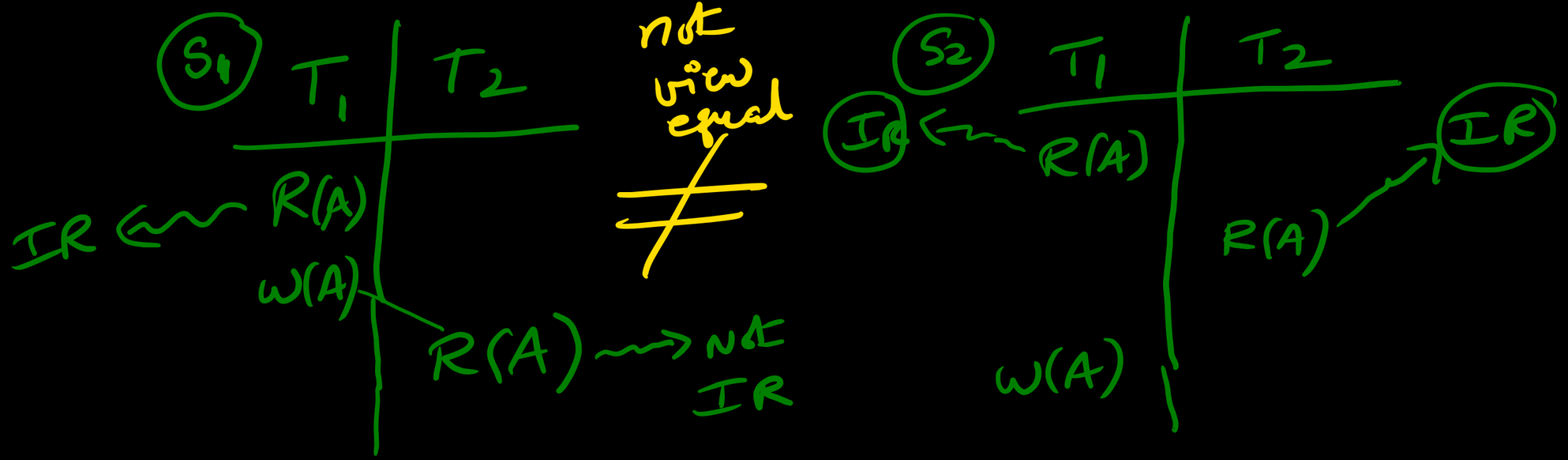
Initial read:

Initial read means read before any write.



S_1 and S_2 are view equal iff

(1) ~~initial~~ every initial read of S_1 must be also initial read of S_2. All initial reads must be same



(1) Initial ~~Acct~~ reads must be same

(2) updated reads:

Every updated read of S_1 and S_2 must be same

S_1:

T_i	T_j
	$w(x)$
$R(x)$	

updated reads are same

S_2:

T_i	T_j
	$w(x)$
$R(x)$	

Same.

T_i is reading 'x' written by T_j in S_1

S1:

T_1	T_2	T_3
W(A)		
	W(A)	
		R(A)

T_3 is reading A written by T_2.

not view equal

S2:

T_1	T_2	T_3
W_A		
	W(A)	
		R(A)

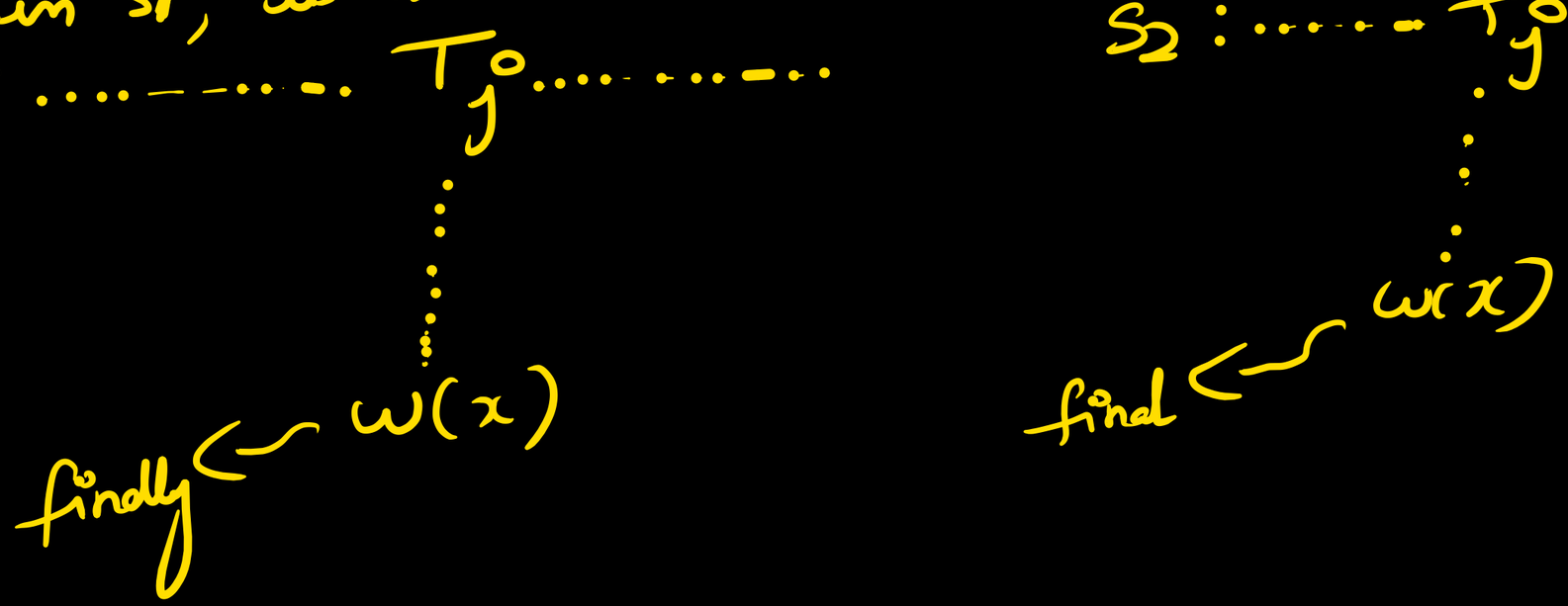
T_3 is reading A written by T_1.

(2) condition failed (updated read)

(1) initial reads ✓

(2) updated reads ✓

(3) Final write: If some transaction T_j writes the data (x) finally in S_1, it should also happen the same in S_2



S_1

T_1	T_2	T_3
$w_1(A)$		
	$w_2(A)$	
		$w_3(A)$

$w_3(A)$ → final write by T_3

not view equal

$\neq$

S_2:

T_1	T_2	T_3
w_A		
		w_A
	w_A	

w_A → final write by T_2

Test S_1 S_2 schedules are view equal or not:

(1) initial read ✓

(2) updated read ✓

(3) Final write ✓

S_1 : $r_1(A)$ $r_2(A)$ $w_1(A)$ $w_2(A)$

S_2 : $r_1(A)$ $w_1(A)$ $r_2(A)$ $w_2(A)$

IR

not initial read.

S_1 S_2 Not are not view equal

~~Conflict Equal~~.

~~SF~~

2) S_1: $r_1(A)$ $r_2(A)$ $r_3(A)$ $w_1(B)$ $w_2(B)$ $w_3(B)$

S_2: $r_2(A)$ $r_1(A)$ $r_3(A)$ $w_2(B)$ $w_1(B)$ $w_3(B)$

$T_1 \rightarrow T_2$
$T_2 \rightarrow T_1$
} Precedence graph is not same

Same IR

~~Conflict~~ ① Satisfied.

② No updated reads

③ Finally write on 'B' is by T_3

∴ These two are view equal.

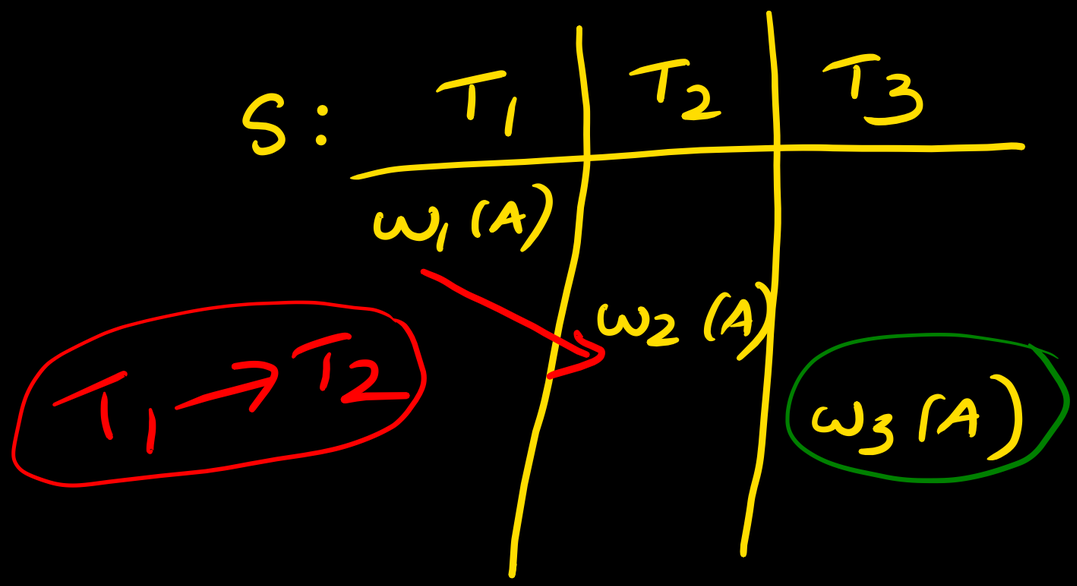
But They are not Conflict equal.

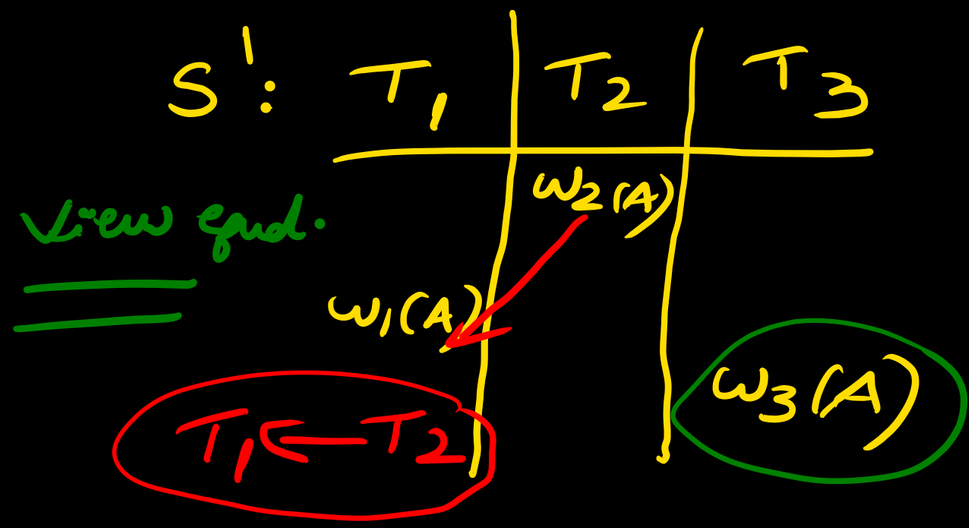
Initial seeds

updated seeds

Final ~~seed~~ units

Should be checked for every data item





View equ.

① NO initial reads

② NO updated reads

③ Final write is by T_3 in both.

∴ ✓ Conflict equ.

Precedence graph is diff.

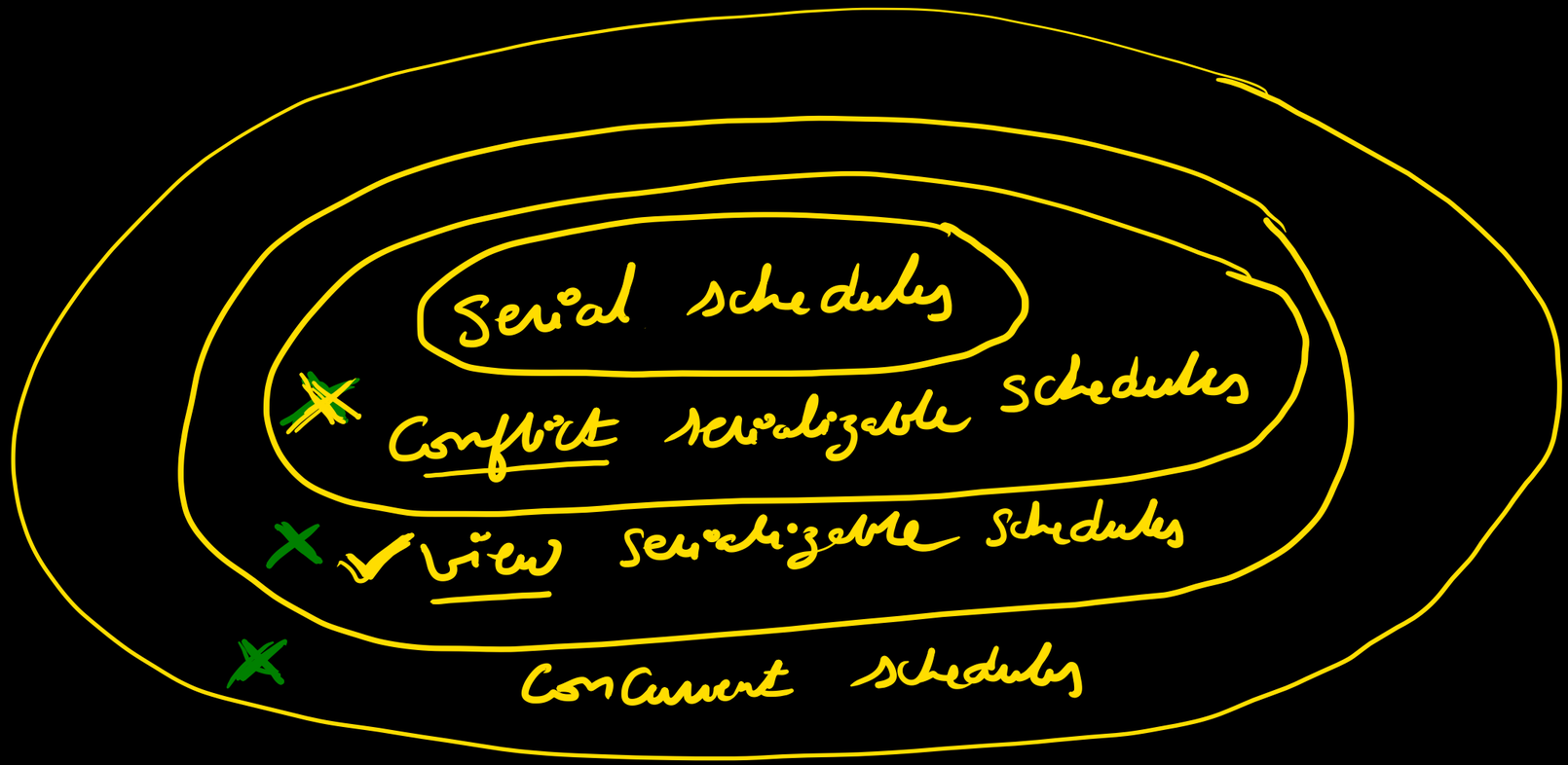
∴ It is not conflict equal.

→ If S_1 S_2 Conflict equal, then every R-W, W-R and WW Conflict pairs of S_1 and S_2 must be in same precedence.

→ If S_1 S_2 are view equal then every initial read, updated read and final write of S_1 and S_2 should be in same order.

→ If S_1 S_2 are conflict equal, then S_1 S_2 are view equal too

→ If S_1 S_2 are not conflict equal, then they may or may not be view equal.



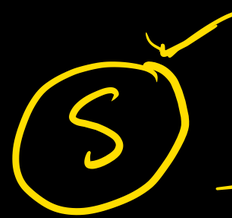
every CS is VS

There are some VS schedules, which are not Conflict Ser

Testing of view serializability:

④ → 4! we need a
5 → 5! shortcut.

Schedule has 3 transaction



T_1	T_2	T_3
$r(A)$		
⋮		
	$w(A)$	
	⋮	
		$r(A)$
		⋮

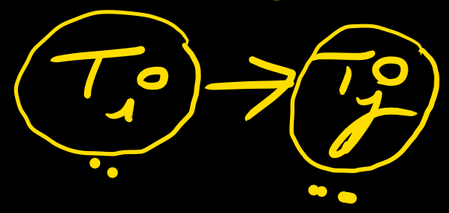
3 subs

IR, UR, FW

① Initial Read:

data item	Initial read	write
x	T_i^0	T_j^0

$\Rightarrow$ view equal serial



y	T_i T_j^0	T_j T_k	$\Rightarrow$ $T_i^0 \rightarrow (T_j, T_k)$	✗
z	T_i^0	NO	$\Rightarrow$ no restoring	

(2) ~~used~~ updated read :

Given Schedule (S)

$W_i^0(x) \rightarrow R_j^0(x)$

No other transaction writing data item x.

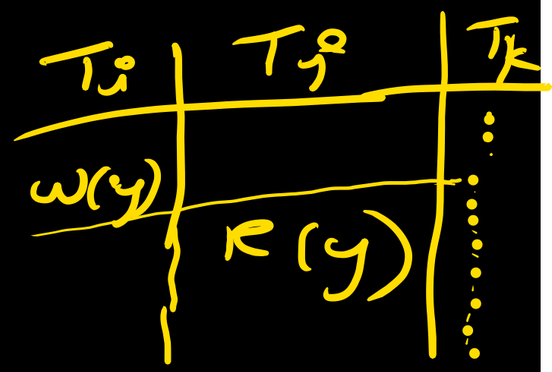
$W_i(y) \rightarrow R_j(y)$

Some other transaction

T_k also writes y

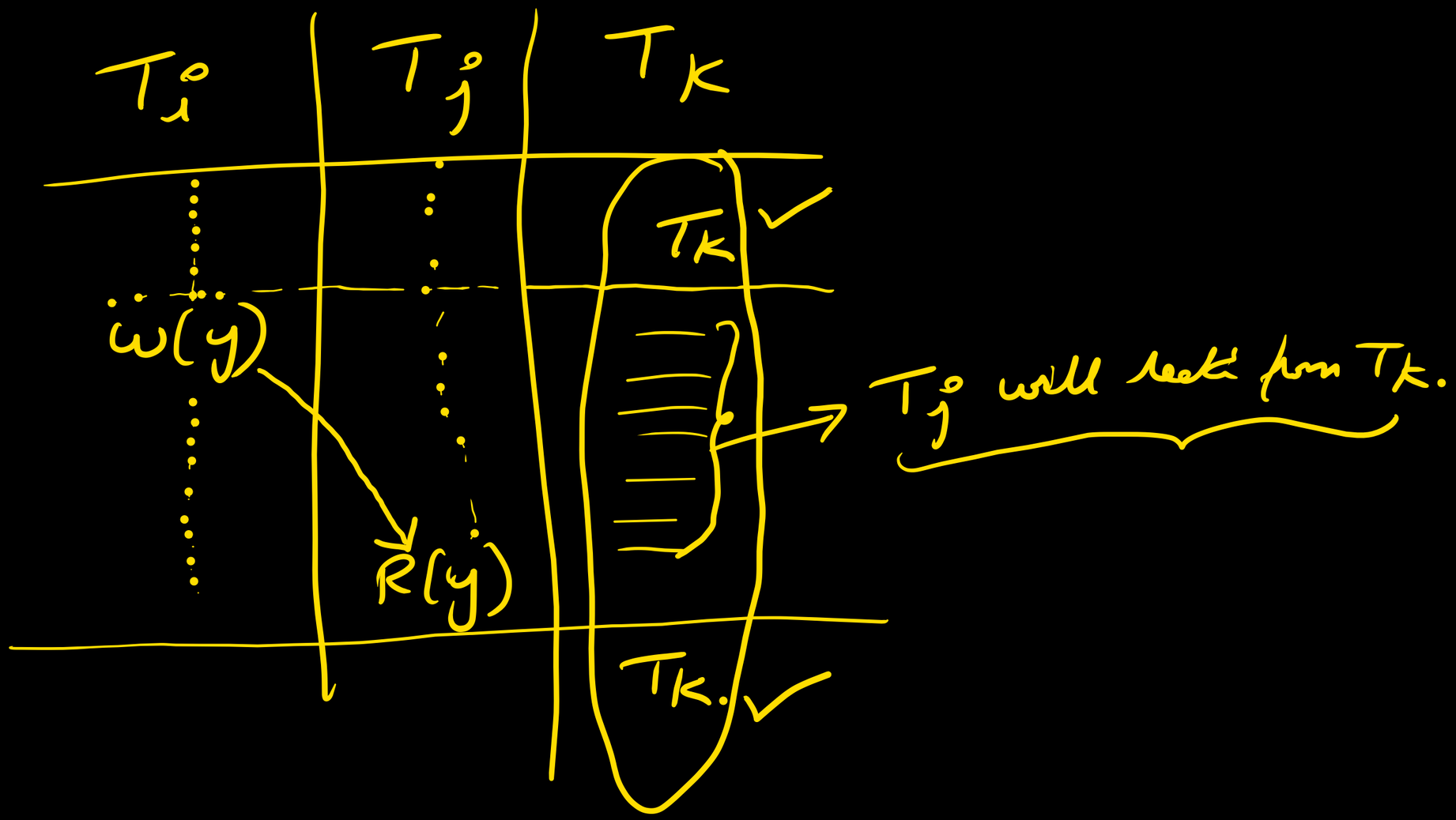
view equal send

$T_i^0 \rightarrow T_j^0$



$T_k \rightarrow T_i^0 \rightarrow T_j^0$

$T_i^0 \rightarrow T_j \rightarrow T_k$


T_i
T_j
T_k
T_k ✓
$w(y)$
$R(y)$
T_k ✓
T_j will read from T_k.

③ Find write:

data item

X : Given schedule T_i T_j find write.

Y : T_i T_j T_k → Fw

Z : T_j

W : No transaction writes data

view equal serial

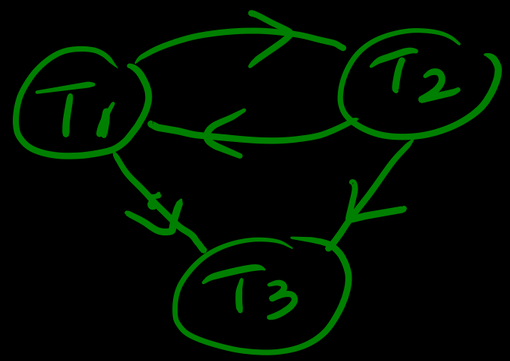
$T_i \rightarrow T_j$

$(T_i, T_j) \rightarrow T_k$.

T_j can appear anywhere

no restriction

S_1: $r_2(A)$ $w_1(A)$ $w_2(A)$ $w_3(A)$



not conflict serializable.

S: $r_2(A)$ $w_1(A)$ $w_2(A)$ $w_3(A)$

Initial read on A:

IR	write
T_2	T_1 T_2 T_3

$T_2 \rightarrow (T_1, T_3)$

No updated reads

Final write on A:

Given schedule:

T_1 T_2 T_3 FW

$(T_1, T_2) \rightarrow T_3$

T_2 T_1 T_3

Serial schedule which is view equal to S.

∴ 'S' is view serializable

$S:\ \gamma_2(A)\ \gamma_2(B)\ \omega_2(A)\ \omega_1(A)\ \gamma_3(A)\ \omega_1(B)\ \omega_2(B)\ \omega_3(B)$

CS $\wedge$ not VS $\wedge$ not.