Conflict Serializability

N.S. — S_1

T_1	T_2
R(A)	
W(A)	
	R(A)
	W(A)
R(B)	
W(B)	
	R(B)
	W(B)

Non Serial.

S. Consistent — S_2

T_1	T_2
R(A)	
W(A)	
R(B)	
W(B)	
	R(A)
	W(A)
	R(B)
	W(B)

Serial.

Conflicting instructions

T_i I_i	I_j T_j
	A=10
	R(A)
W(A)	

R(A)	R(A)	No conflict
R(A)	W(A)	conflict
W(A)	R(A)	conflict
W(A)	W(A)	conflict

W(A)	
15	
	R(A)
	15

	W(A)
	10
W(A)	
15	

15 W(A)	
	W(A)
	10

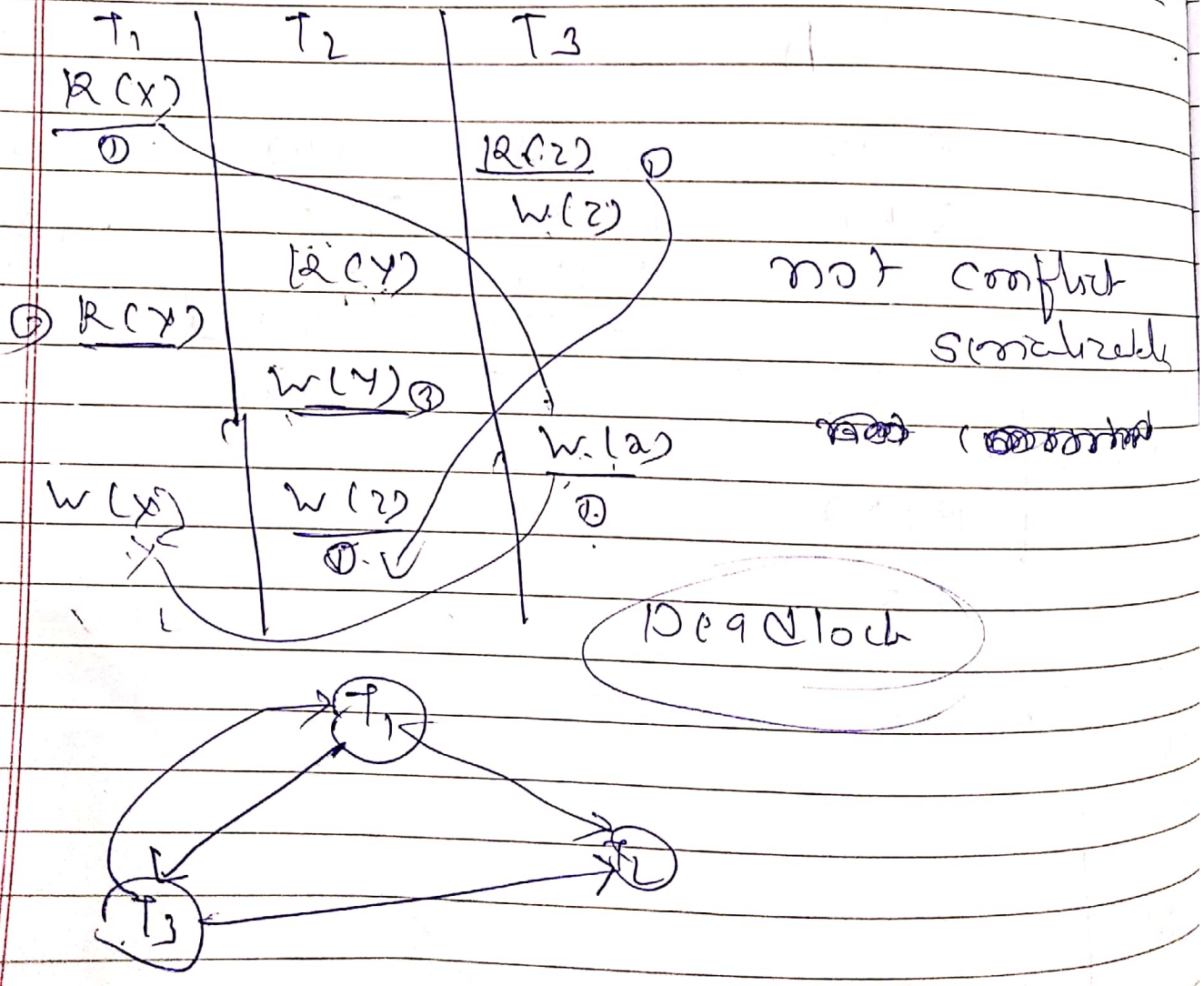
Non serial schedule

→ after swapping non conflicting instructions if we get serial schedule then it is conflict serializable

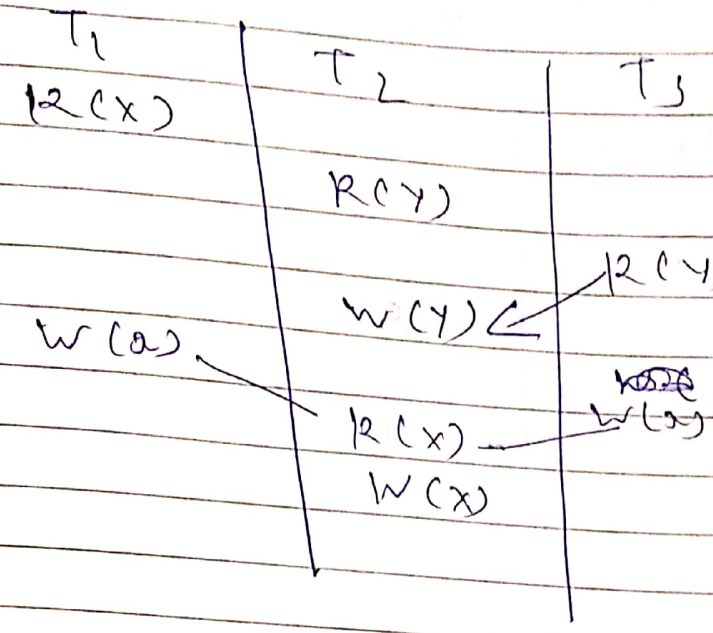
— consistent.

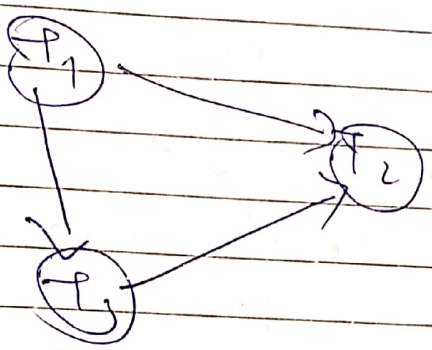
① serial → consistent.

② non serial → conflict serializable → consistent.



no cycle ⇒ conflict strictly schedule.





no cycle
conflict serializable.

$T_1 \rightarrow T_3 \rightarrow T_2$

Dirty Read

T_1	T_2
R(A)	
W(A)	
	R(A)
	W(A)
~~fail~~	C
C	

Unrepeatable Read

T_1	T_2
~~R(A)~~	R(A)
R(A)	
W(A)	
	R(A)

phantom Read

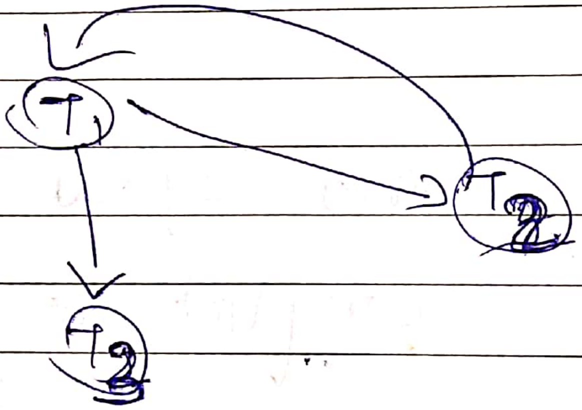
T_1	T_2
	R(A)
R(A)	
Delete(A)	
	R(A)

lost update

R(A)	
W(A)	
	W(A)
	C
C	

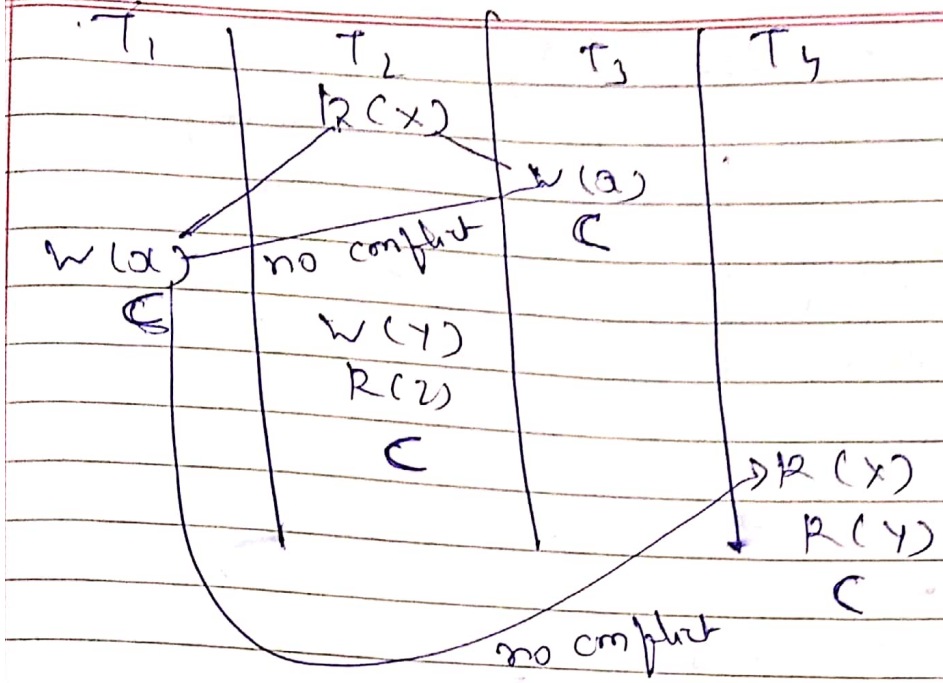
(*)

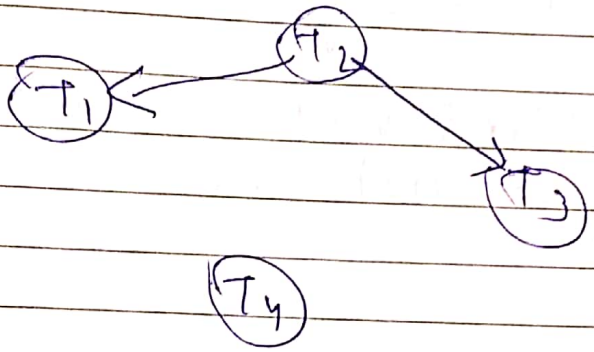
T1	T2	T3
R(B)		R( )
R(A)		
	W(A)	
W(A)		
	W(B)	
		W(A)
W(B)		
		W(B)
		W(C)



cycle exist

not conflict serializable

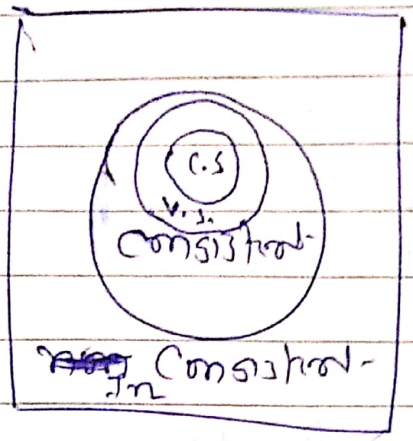




no cycle => conflict serializable

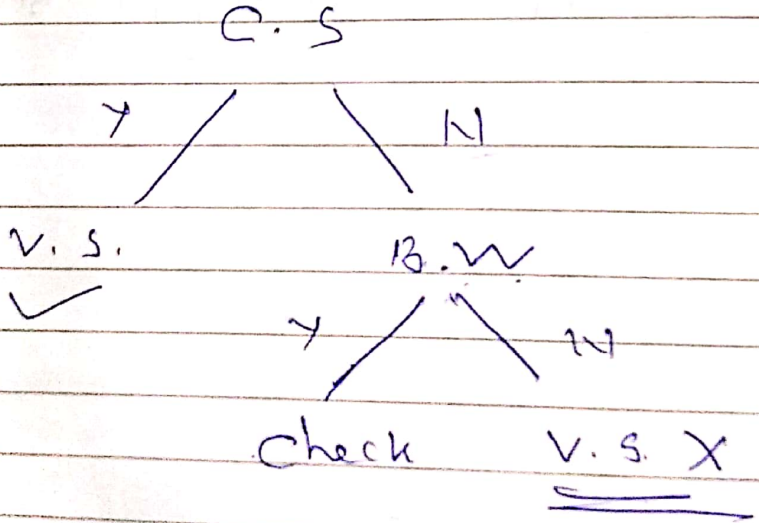
(*) View Serializability.

└ n.p. complete problem.

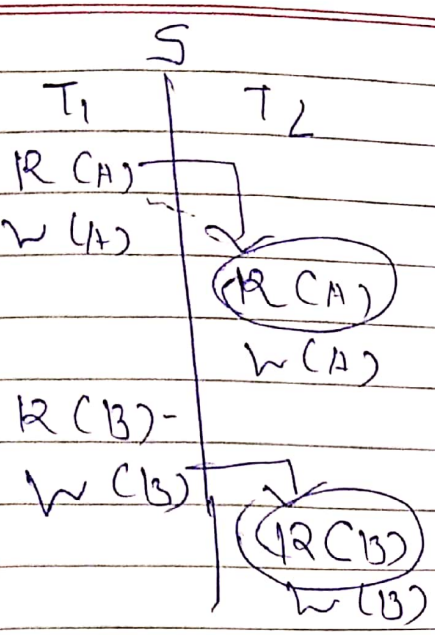


C.S. is V.S.
V.S is not always C.S.

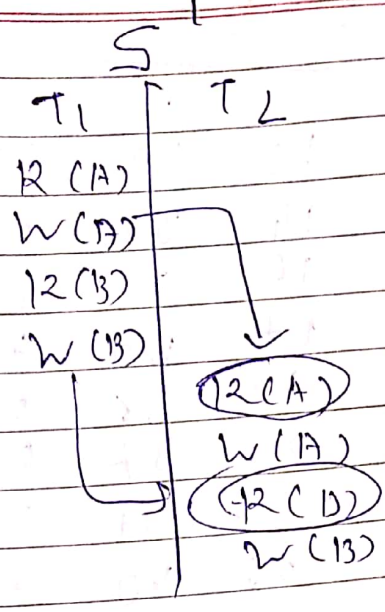
→ atleast one blind write should be there if it not C.S.
⇒ V.S. Check



Non serial



Serial



View equivalent

$S_1 \rightarrow$ S a S' (View equivalent)

if S & S' are view equivalent.

$S \Rightarrow S'$ if View Serializable

	S	S'
First Read		
a	T_1	T_1
b	T_1	T_1
Final Write		
a	T_2	T_2
b	T_2	T_2

Intermediate read.

a	$T_1 \rightarrow T_2$	$T_1 \rightarrow T_2$
b	$T_2 \rightarrow T_1$	$T_2 \rightarrow T_1$
View Serializable		

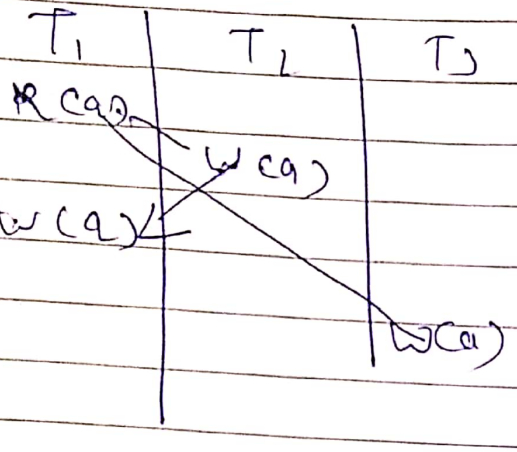
S_1 | 1 2 3 S_5 3 1 2

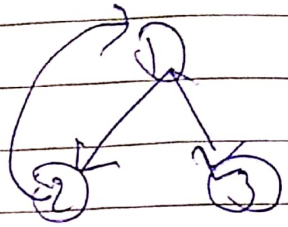
S_2 | 1 3 2 S_4 3 2 1

S_3 2 1 3

S_4 2 3 1

S





not conflict serialized

→ T_2 & T_3 have blind Writes.

Initial Read

S S_1

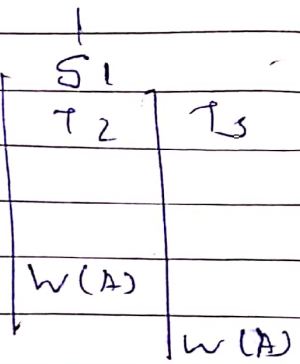
Q → T_1 T_1

Final Write

Q → T_3 T_3

Intermediate read.

no intermediate read.



S & S_1 are View equivalent

⇒ View Serialized

np Complete ⇒ as n! schedule will be there

Committed transaction can not roll back

Recoverable Schedule

mandatory property

failure.

A = 10

S

T_1	T_L
R(A)	
A = A + 10	
W(A)	Dirty Read
	R(A) 20
	A = A − 5
	W(A) = 15
	C
failure	

Value is being read which is temporary.

if there is no Dirty read in transaction, schedule is always recoverable

Irrecoverable schedule.
- does not guarantee consistency.

S

T_1	T_L
R(A)	
A = A + 10	
W(A)	
	R(A)
	A = A − 5
	W(A)
C	f
	C

Dirty Read is there so schedule is recoverable

Dirty Read

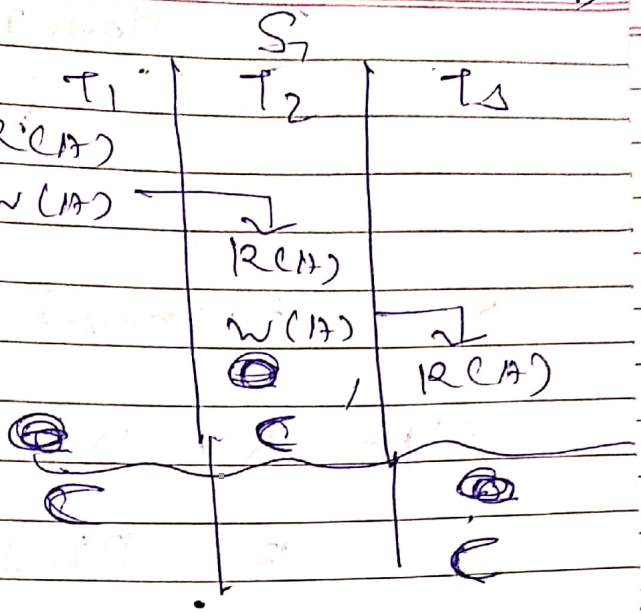
- N → Recoverable ✓
- Y → Commit should be done in order of dirty read.
  - R ✓
  - R ✗

(CL)

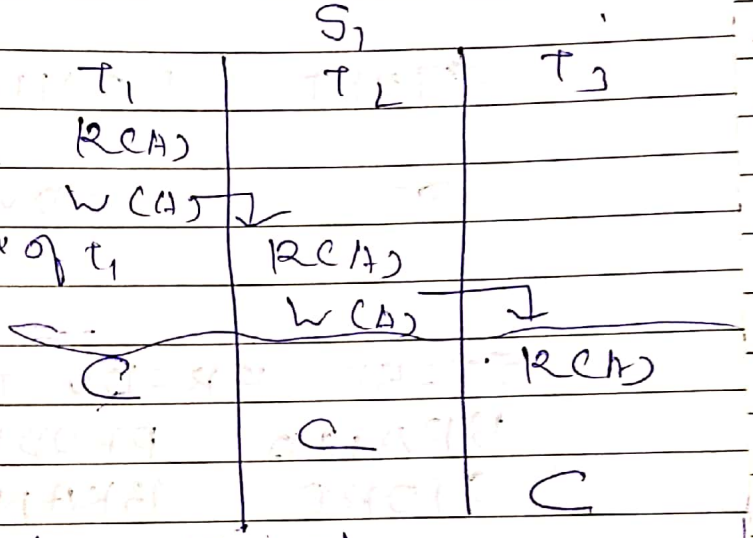
Cascadeless (optional not mandatory)

Cascadeless Schedule

C.S. ✓
D.R. ✓
$T_1 \rightarrow T_2 \rightarrow T_3$
Commit
$T_2 \rightarrow T_1 \rightarrow T_3$
Irrecoverable ✓
schedule



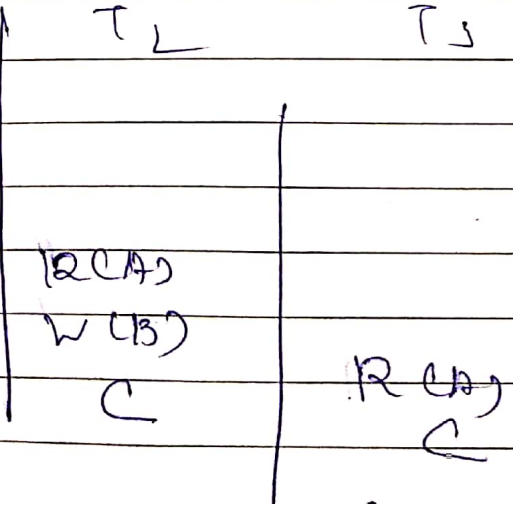
Cascading Rollback.
T_1 Rollback
T_2 roll back because of T_1
& $T_3 \rightarrow T_2$



Schedule is cascadeless if it
- don't have cascading rollbacks.

Dirty Read should not be there

Cascadeless schedule
→ no rollbacks as no dirty read.



strict schedule

S_1

T_1	T_2
R(A)	
W(A)	
	W(A)
C	
	R(A)
	C

Blind Write

S_2

T_1	T_2
R(A)	
W(A)	
C	
	W(A)
	R(A)
	C

strict schedule

S_3

T_1	T_2
R(A)	
~~W(A)~~	R(b)
W(A)	
	W(b)
C	
	R(A)
	C

~~strict~~ strict schedule

no other transaction can ~~read~~ read, ~~or~~ write until ~~of~~ unless current transaction commit write