

$$S2 = \underline{R_1(A) W_1(A) R_1(B)} \quad \underline{W_2(A) R_2(A) W_1(B) R_2(B) W_2(B)}$$

$$S3 = \underline{\underline{R(A) W(A) R_1(B) W_1(B)}} \quad \underline{\underline{R_2(A) W_2(A) R_2(B) W_2(B)}}$$

T_1

T_2

$S1$ is conflict serializable.

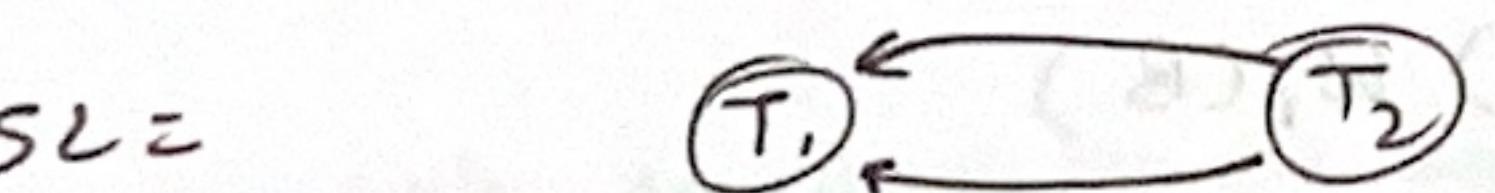
$S1 \neq S2 \neq S3$ are conflict equivalence.

Q: Consider the following schedules involving 2 transactions, which one of following statement is true:-

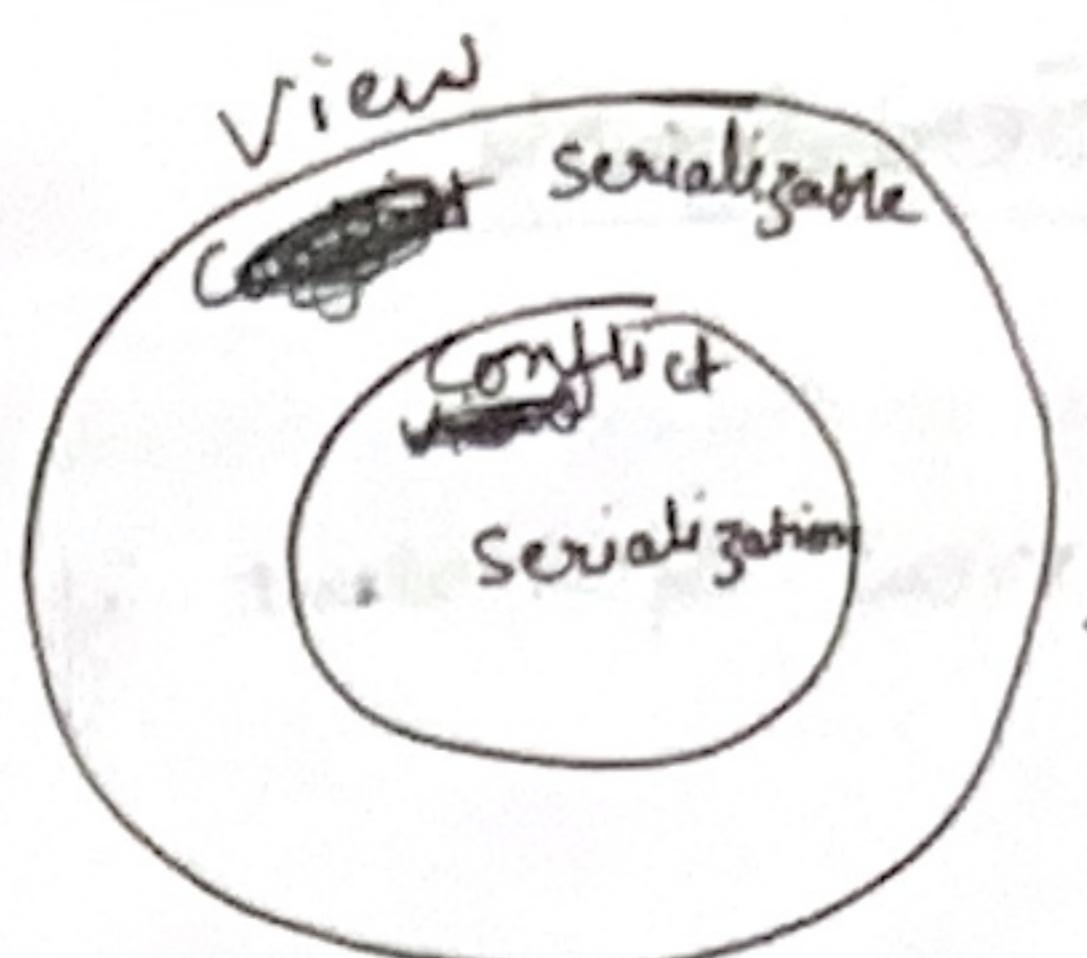
$$S1 = R_1(x) R_1(y) R_2(x) R_2(y) \xrightarrow{W_2(y) W_1(x)}$$

$$S2 = R_1(x) R_2(x) R_1(y) W_2(y) \xrightarrow{W_1(y) R_1(y) W_1(x)}$$

- (A) Both $S1$ & $S2$ are conflict serializable
- (B) only $S1$ is CS
- (C) Only $S2$ is CS
- (D) None.



Only $S2$ is conflict serializable.



If a schedule is conflict serializable then it will be view serializable but if a schedule is view serializable it may or may not be conflict serializable.

\Rightarrow A schedule is view serializable if it is view equivalent to its serial ~~transaction~~ schedule.

Management of ACID Properties

{Application} A = Transaction Manager

{Consistency} C = Application Programmer

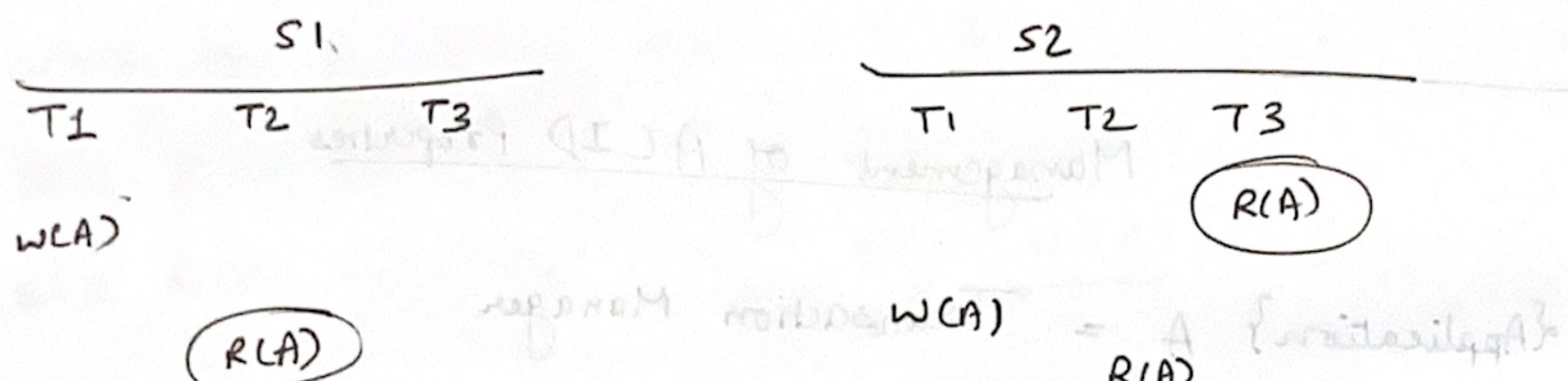
{Isolation} I = Concurrency Control Manager

{Durability} D = Recovery manager.

View Serializable

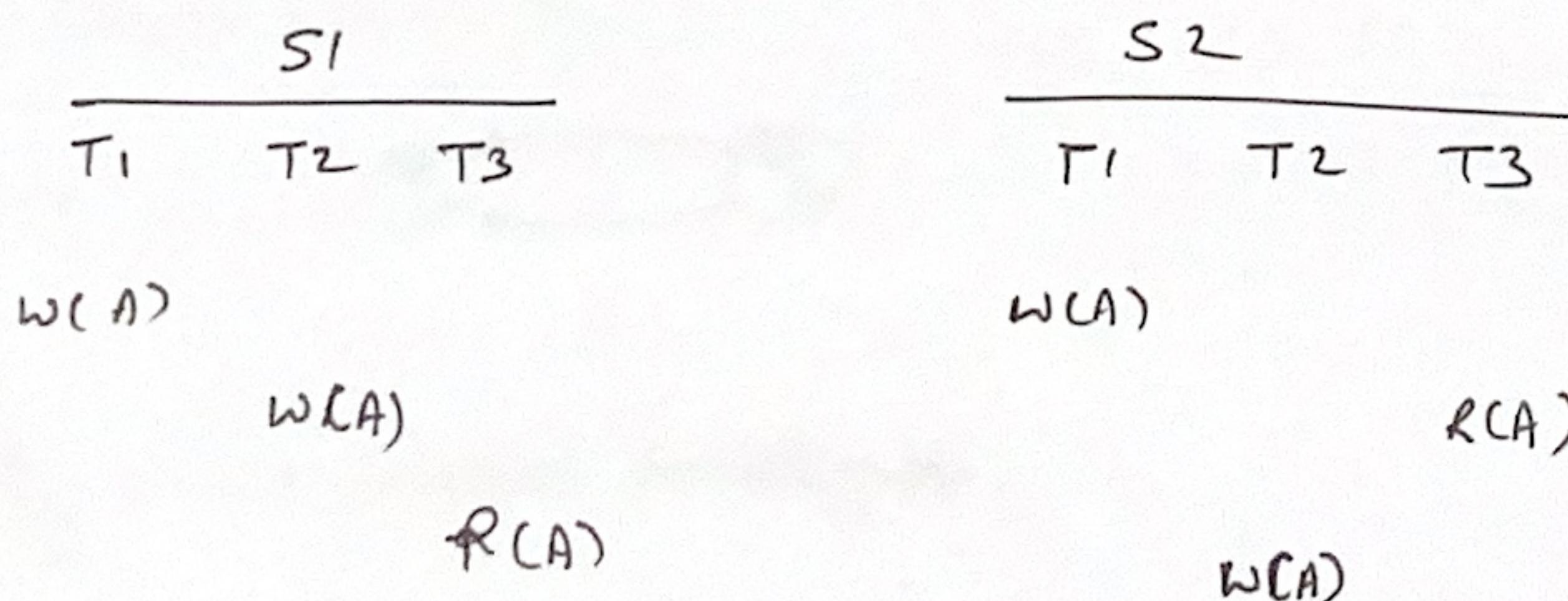
→ Two schedules are view equivalent if the below conditions are satisfied.

- ① Initial Read:- if a transaction T reads data item A from the database in S₁ then in S₂ also _{T₁} should read initial value of A



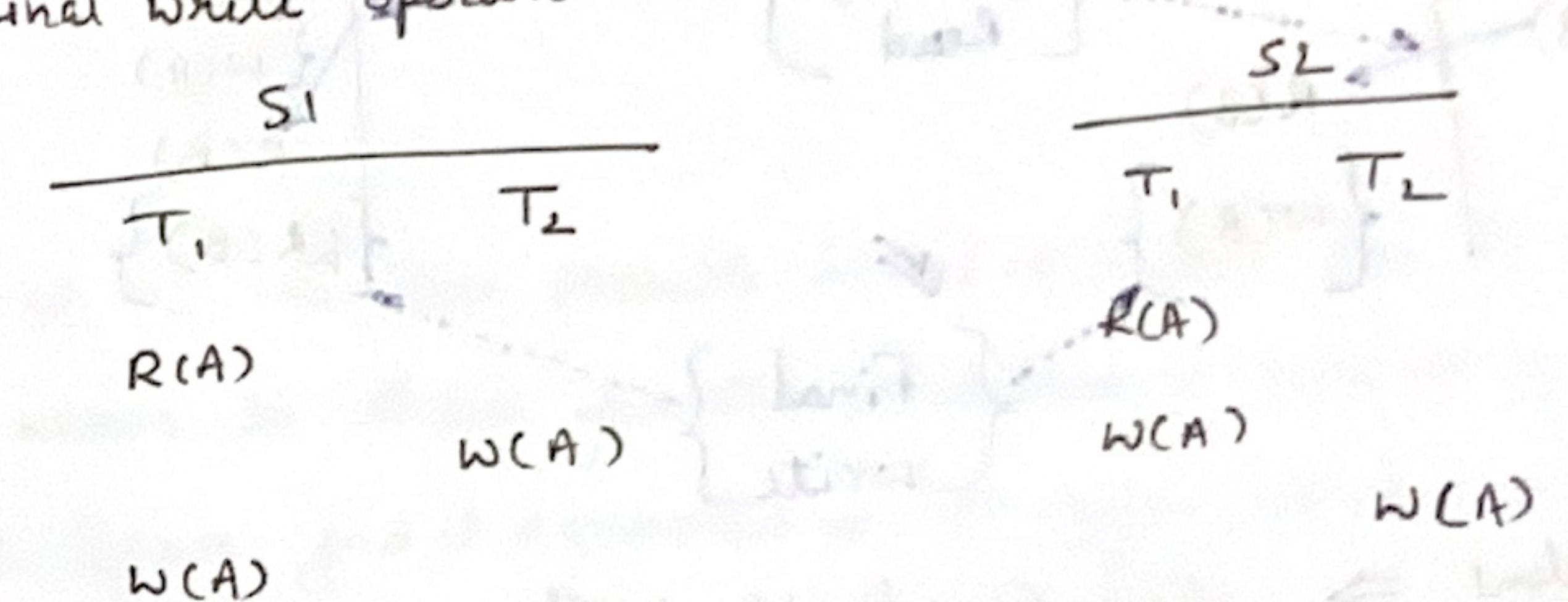
Not View equivalent as initial read of A in S₁ is T₂ while in S₂ it is T₃.

- ② Updated Read⇒ If T_i is reading A which is updating by T_j in S₁ then in S₂ also T_i should read A which is updated by T_j.



{ Not view equivalent as _{T₃} is reading after write from T₂ while in S₂ _{T₃} is reading after T₁'s write. }

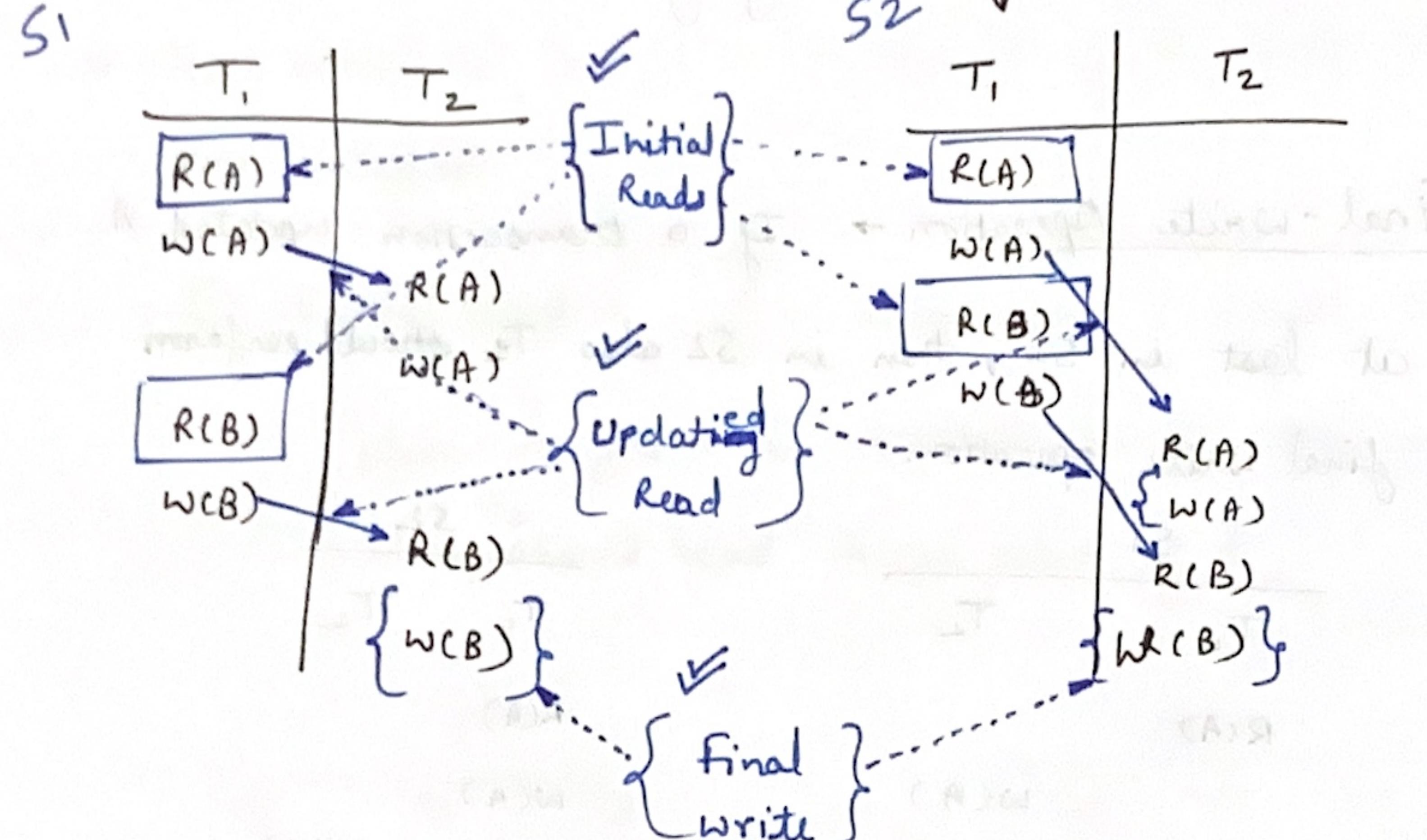
- ③ Final-Write Operation⇒ If a transaction updated A at last in S₁, then in S₂ also T₁ should perform final write operation.



{ Not view equivalent as final write is on T₁ but in S₂ it is in T₂ }

Blind Write ⇒ Performing write operations without read operation is known as Blind write.

$$S1 = R_1(A) W_1(A) R_2(A) W_2(A) R_1(B) W_1(B) R_2(B) W_2(B)$$



View equivalent \Rightarrow View Serializable ✓

S1 =

T ₁	T ₂
R(A)	<u>R(A)</u>
W(A)	<u>W(A)</u>
R(B)	<u>R(B)</u>
W(B)	<u>W(B)</u>
R(B)	R(A)
W(B)	W(A)
R(B)	R(B)
W(B)	W(B)

{ Not view equivalent }

T ₁	T ₂
R(A)	
W(A)	
R(B)	
W(B)	
	R(A)
	W(A)
	R(B)
	W(B)

S1 & S3
(Initial Read) X

S2 & S3 are two possible serial schedule for S1 that's with both we have to check for the conditions of equivalency in them. If we find it equivalent with anyone we don't need to check with other. But if let's S2 and S3 are not equivalent then we have to check for S1 & S3 as well

S ₁		S ₂		S ₃	
T ₁	T ₂	T ₁	T ₂	T ₁	T ₂
R(x)		R(x)			R(x)
R(y)		R(y)			R(y)
				R(x)	
				R(y)	
				W(x)	
					W(y)

S1 & S2
Initial Read ✓
Updated Read X

S1 & S3
Initial Read X