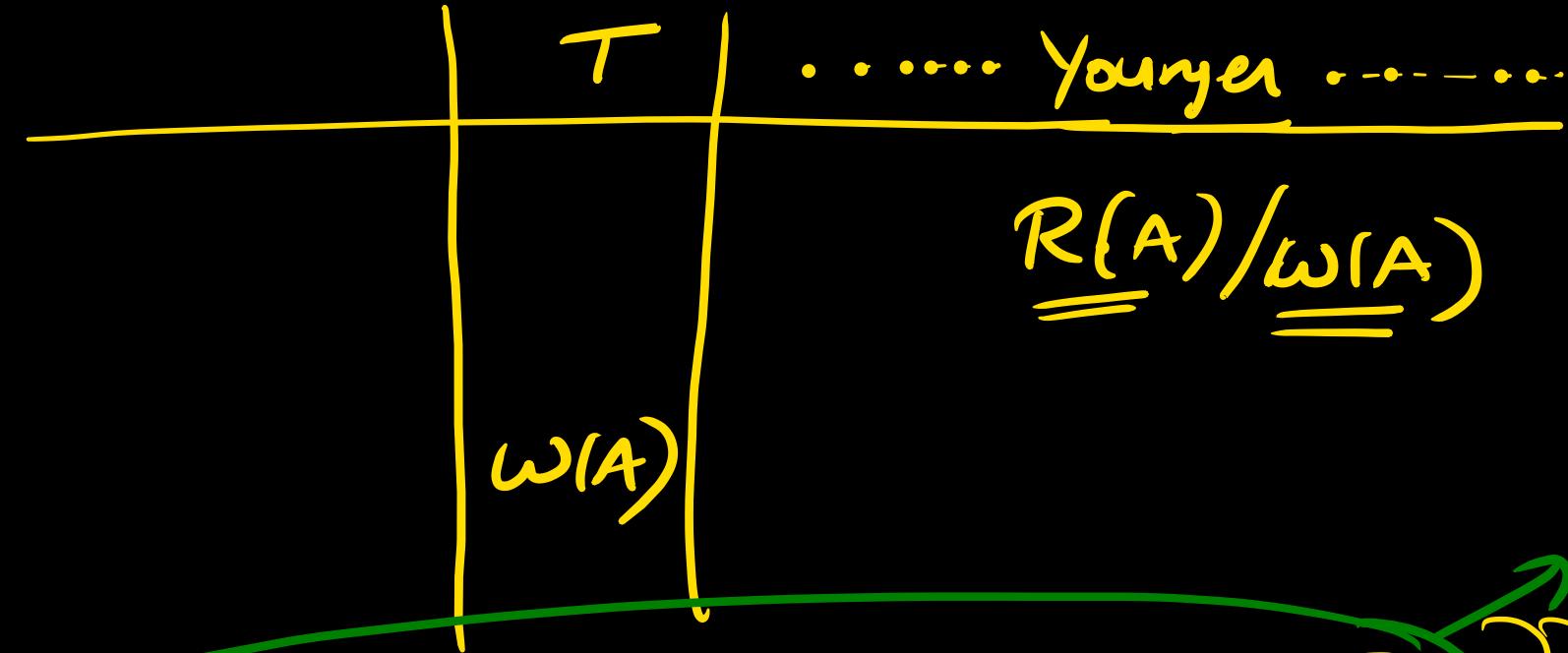


2) Transaction T issues ~~(\oplus)~~ $w(A)$:

~~UK Examples~~
 \rightarrow tomorrow



if $\underline{RTS}(A) > TS(T) \rightarrow$ rollback T
 else if $\underline{WTS}(A) > TS(T) \rightarrow$ rollback T
 else allows $\underline{\underline{w(A)}} \text{ by trans } (\underline{\underline{T}})$
 = set $\underline{\underline{WTS(A)}} = \underline{\underline{TS(T)}}$ ✓

if these two conditions failed, it means that no younger transaction has done $\underline{\underline{read}} w(A)$.

Mon, Fri 6 PM - 8 PM

also do
=

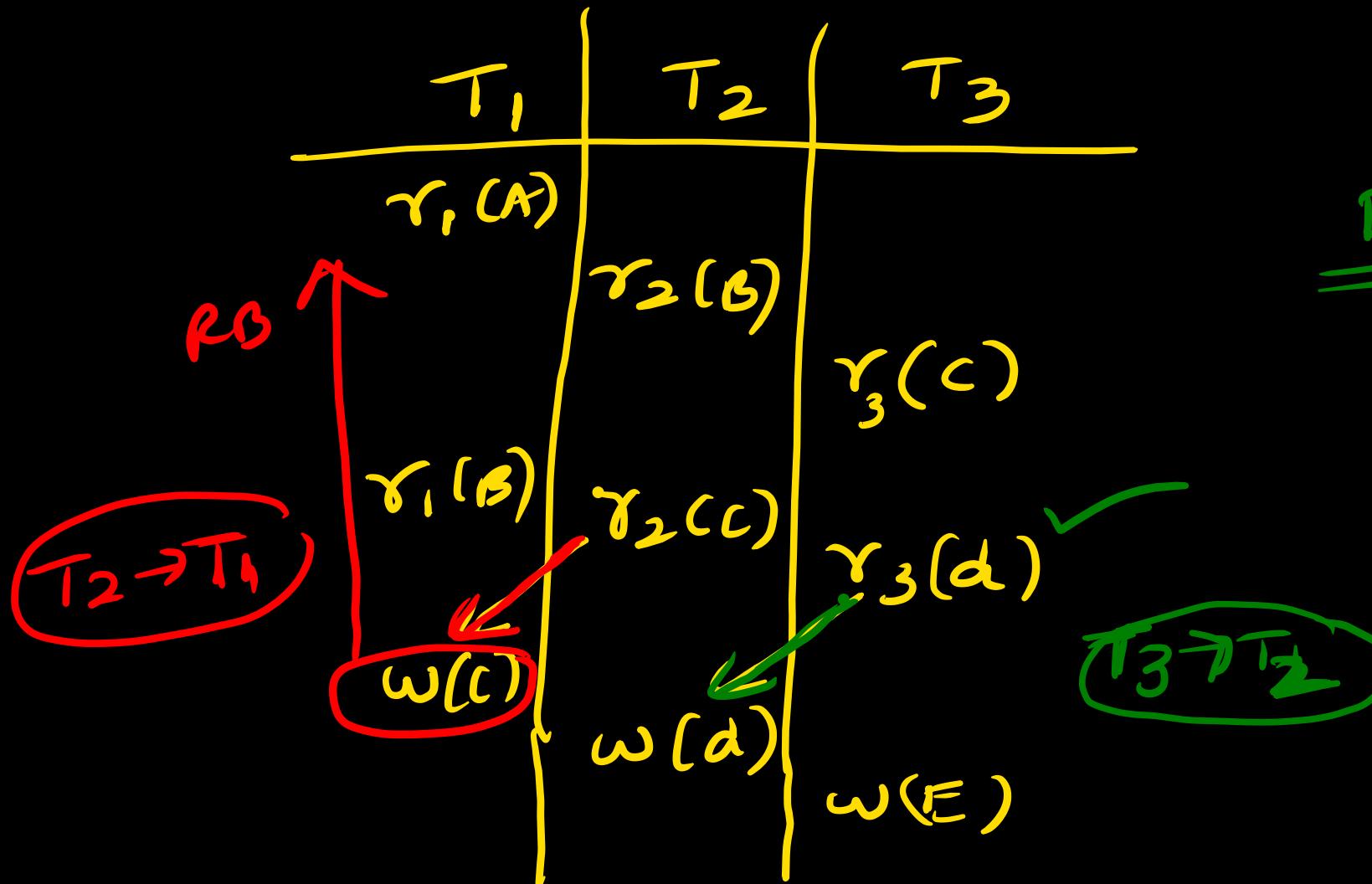
gray Mon, T, W, Th, F, S, S - 8:15 10:15
practice.

Tues and thus Sat \rightarrow digital (6:00 - 8: PM)

which transaction rolls back using basic TS ordering protocol

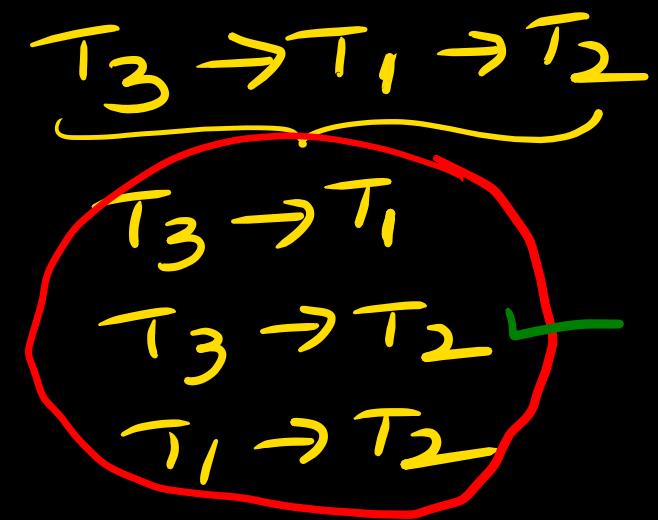
S: $\gamma_1(A) \gamma_2(B) \gamma_3(C) \gamma_1(B) \gamma_2(C) \gamma_3(D) \omega_1(C) \omega_2(D) \omega_3(E)$

TS val $(T_1, T_2, T_3) = 20 \ 30 \ 10$

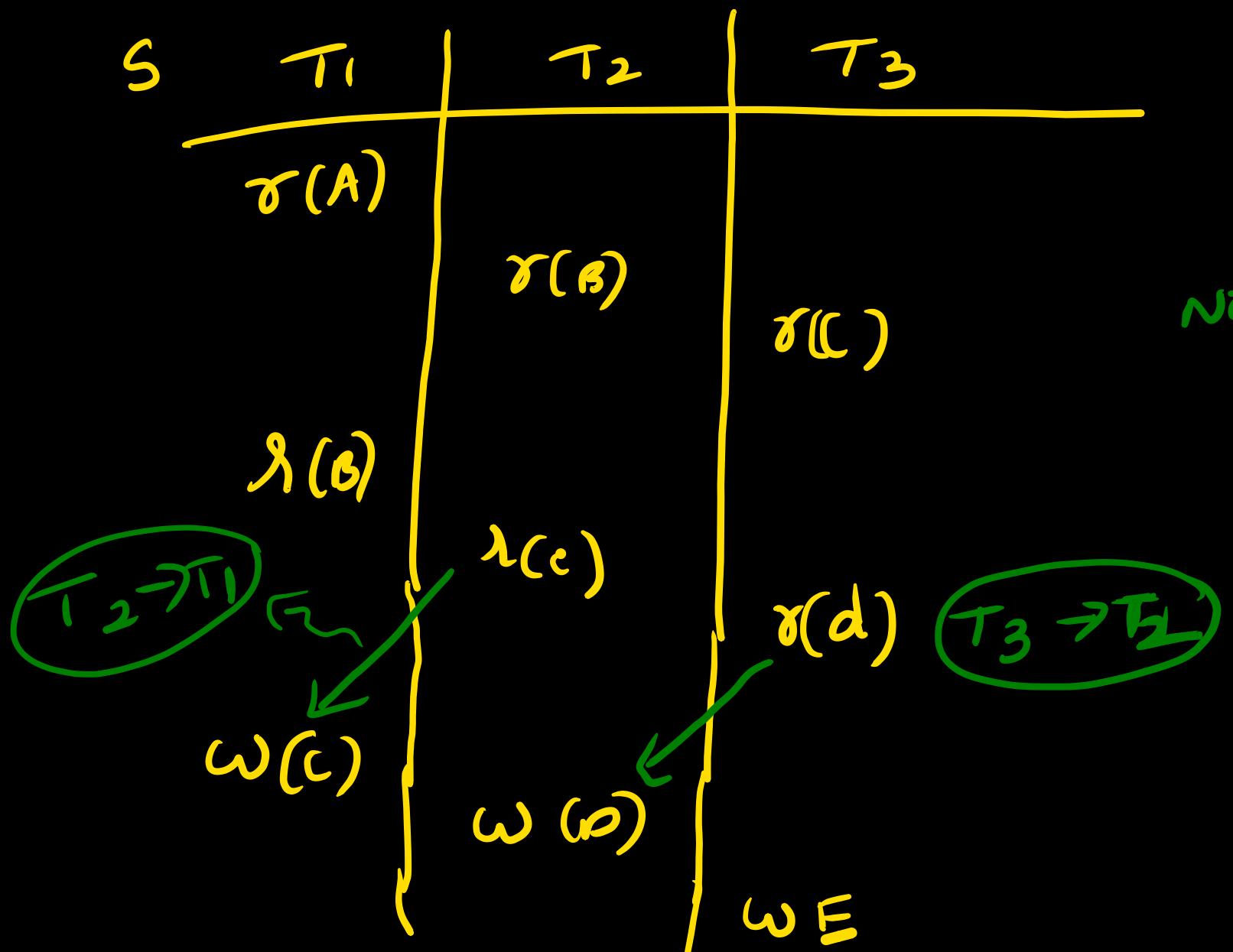


BTSO

T_1 is rolled back.



$$TS\ val(T_1, T_2, T_3) = (30, 20, 10)$$

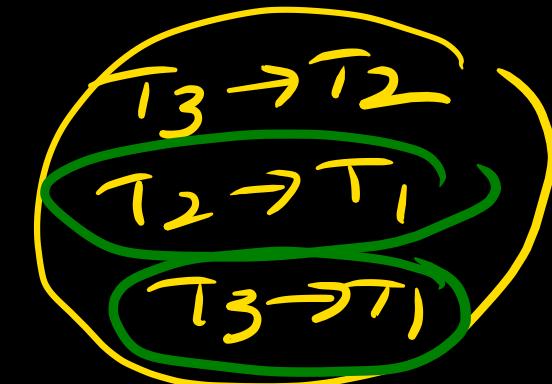


serial schedule

$T_3 T_2 T_1$

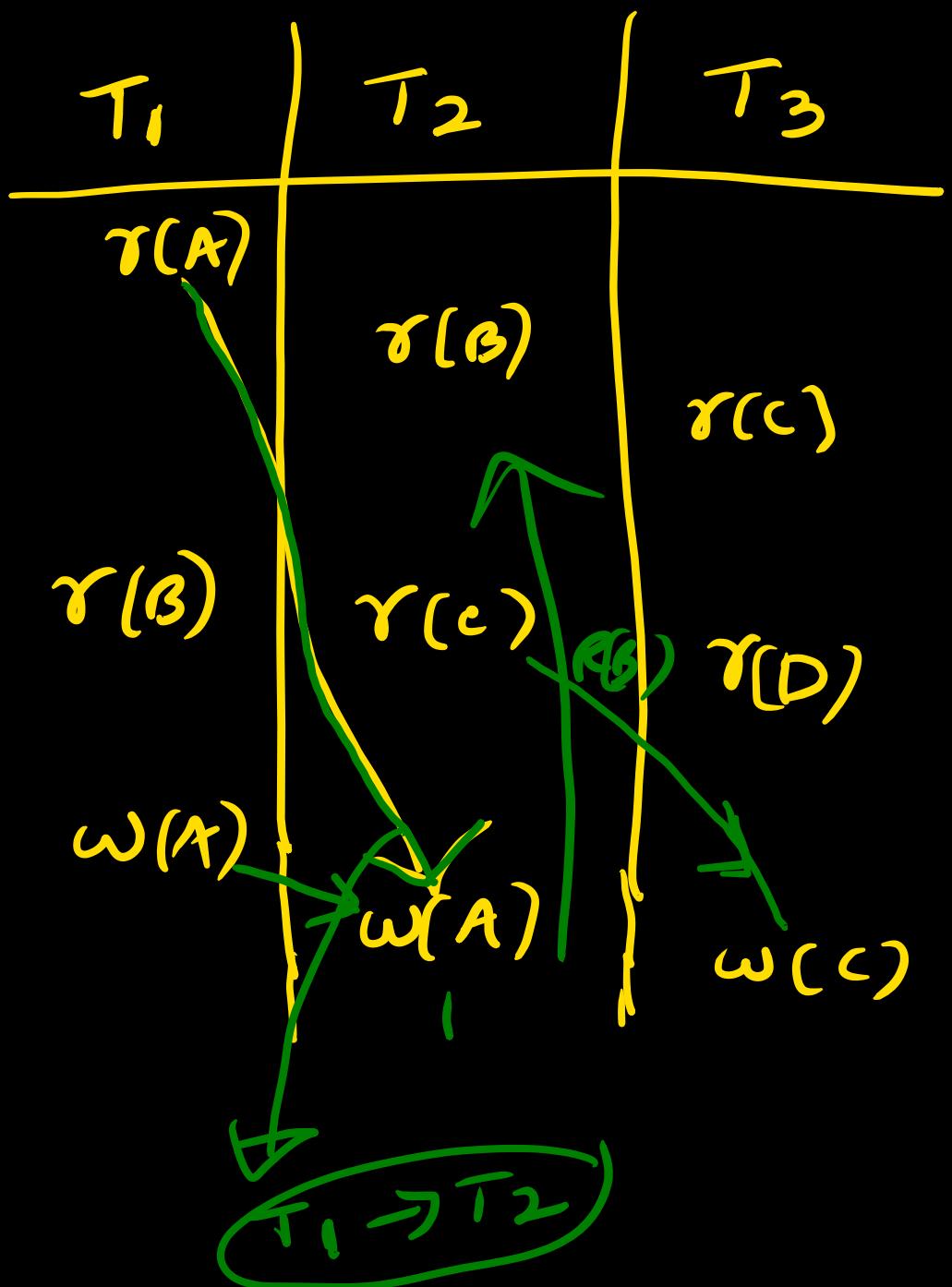
$T_3 \rightarrow T_2 \rightarrow T_1$

no roll backs



TS val $(T_1, T_2, T_3) = (20, 10, 30)$

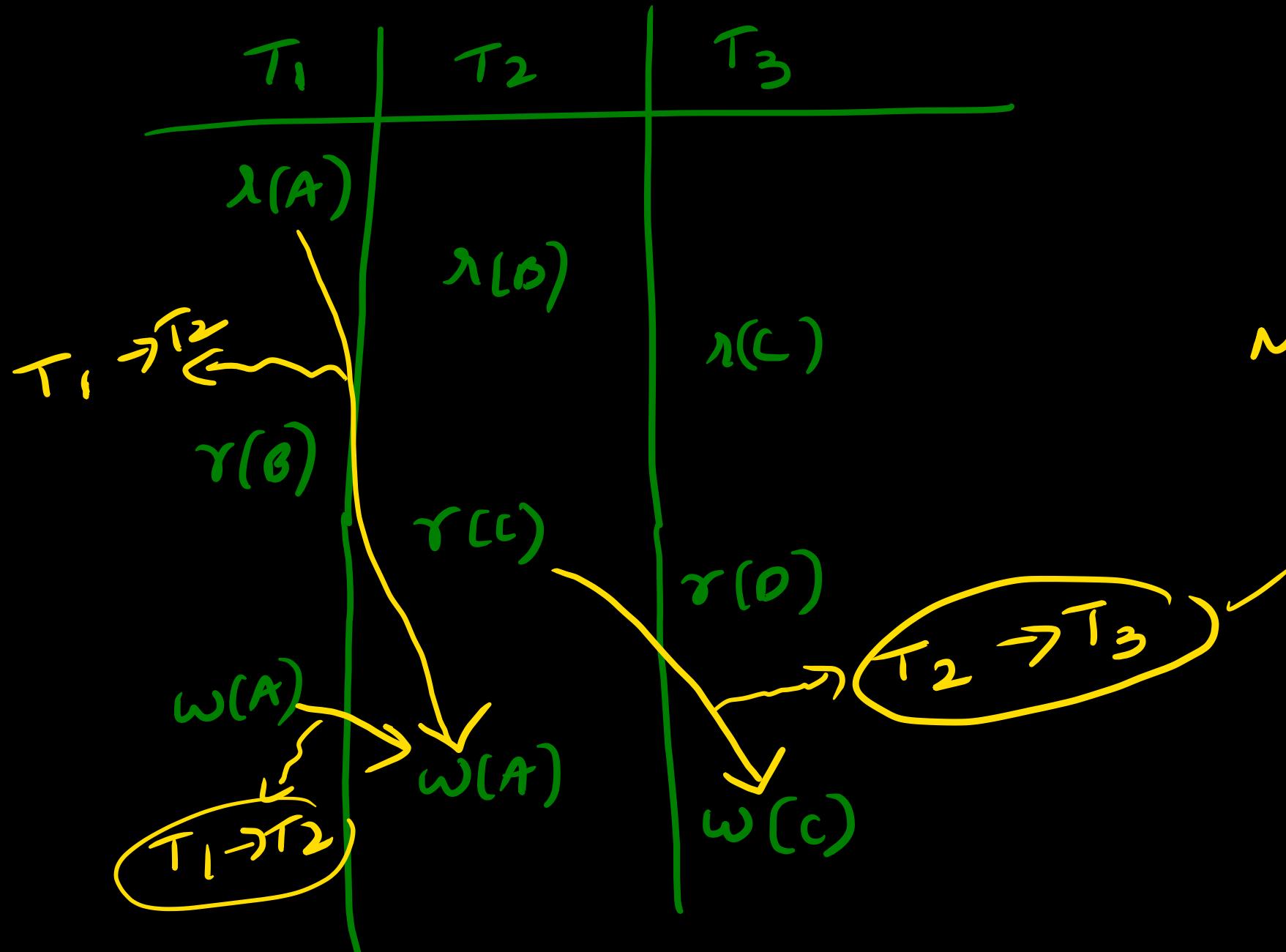
$T_2 \rightarrow T_1 \rightarrow T_3$



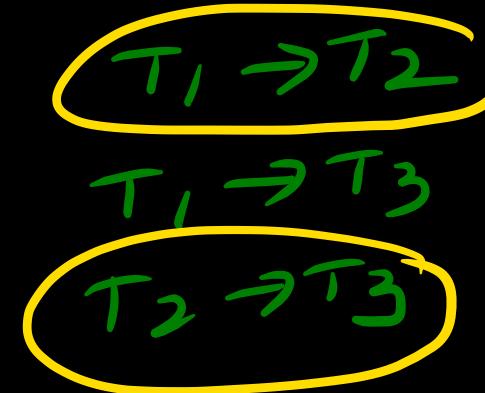
$T_2 \rightarrow T_1$
 $T_1 \rightarrow T_3$
 $T_2 \rightarrow T_3$

$T_2 \rightarrow$ back.

TS val $(\tau_1, \tau_2, \tau_3) : (10, 20, 30)$

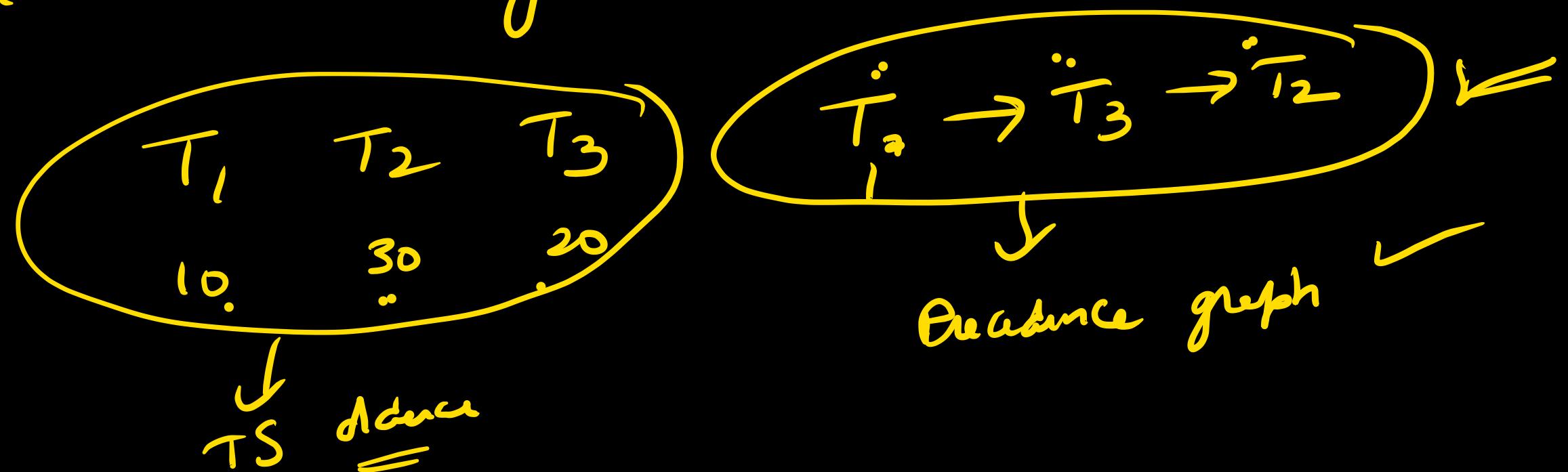


$\tau_1 \rightarrow \tau_2 \rightarrow \tau_3$



NO ~~REB~~ RDM back.

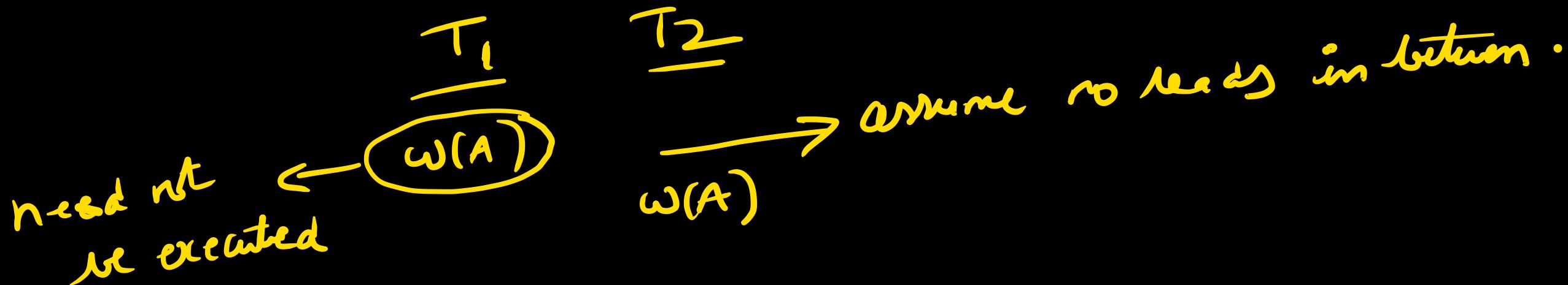
One conclusion: Basic time stamp ordering protocol allows schedule S, if S is conflict serializable and conflict equal order must be based on TS ordering.



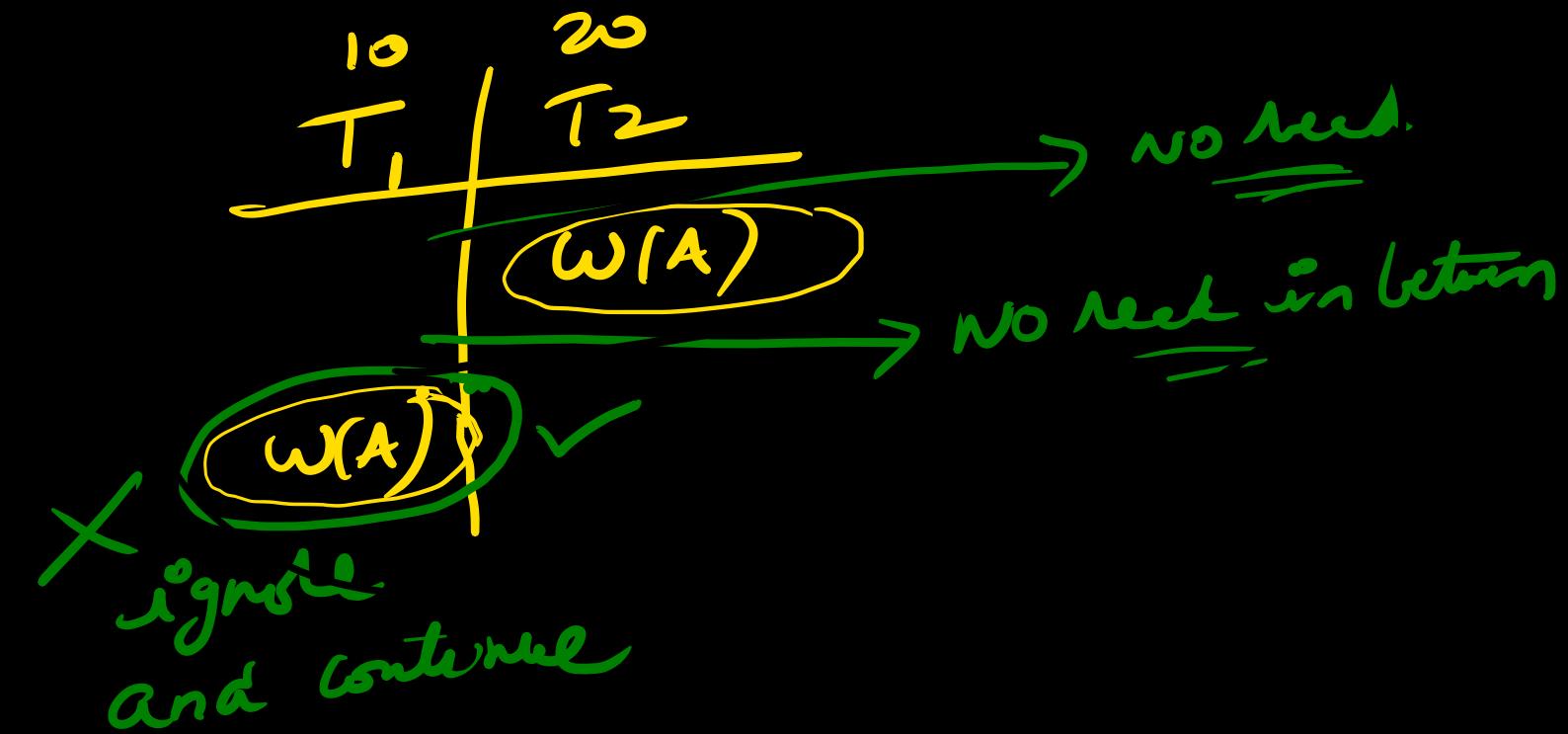
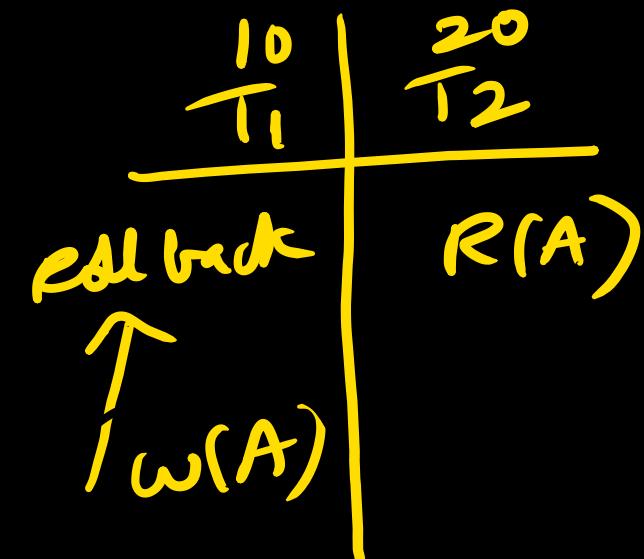
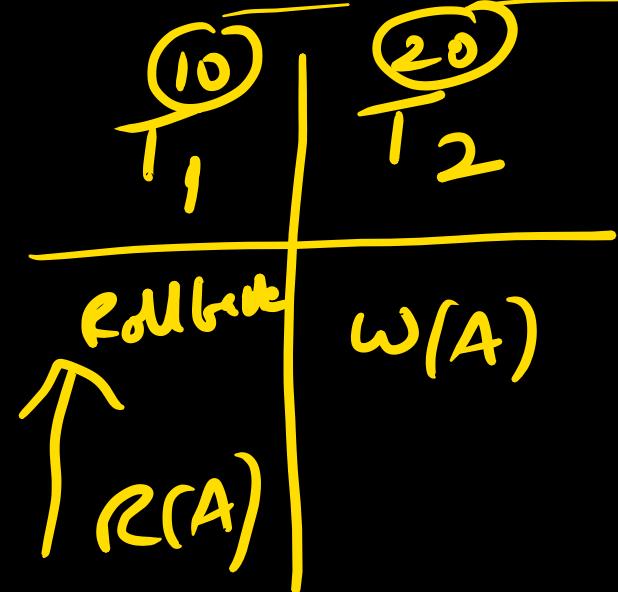
Thomas write TS ordering protocol:

10 20
T₁ T₂

Both of them ~~wants~~ wants ~~ref~~ $\omega(A)$. According to TS ordering
T₁ should write first and then T₂. T₂ overwrites T₁



Thomas with TS delay protocol:

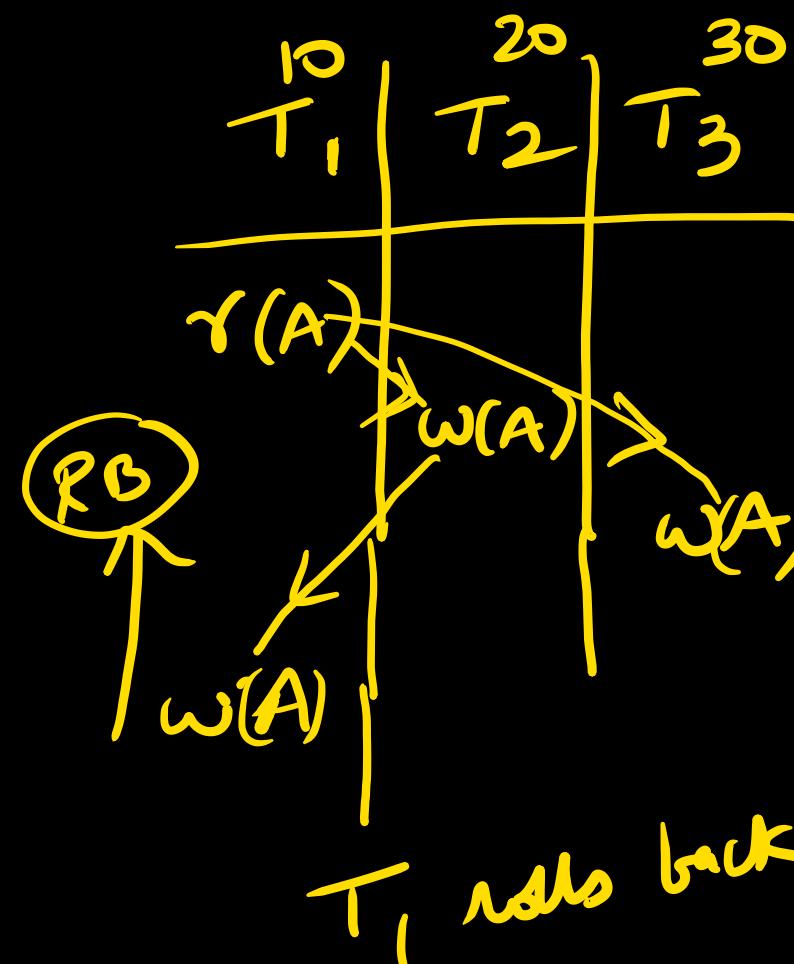


T_1 's $\omega(A)$ is supposed to happen before T_2 's $\omega(A)$.
By any way T_1 's $\omega(A)$ is overridden by T_2 's $\omega(A)$
so we can ignore it if it comes out of place.

S: $\gamma_1(A) \omega_2(A) \omega_1(A) \omega_3(A)$

TS of $(T_1 T_2 T_3) = (10 \ 20 \ 30)$

Basic TS ordering:



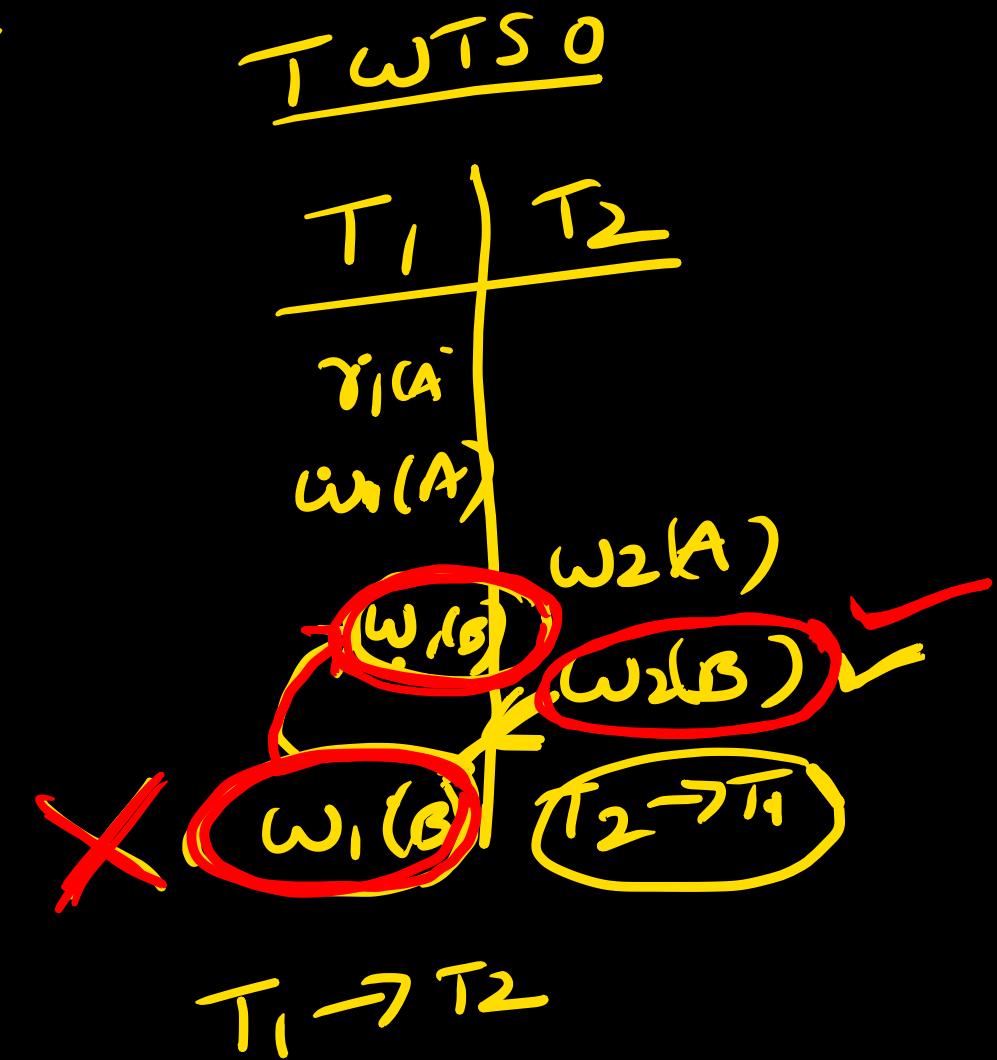
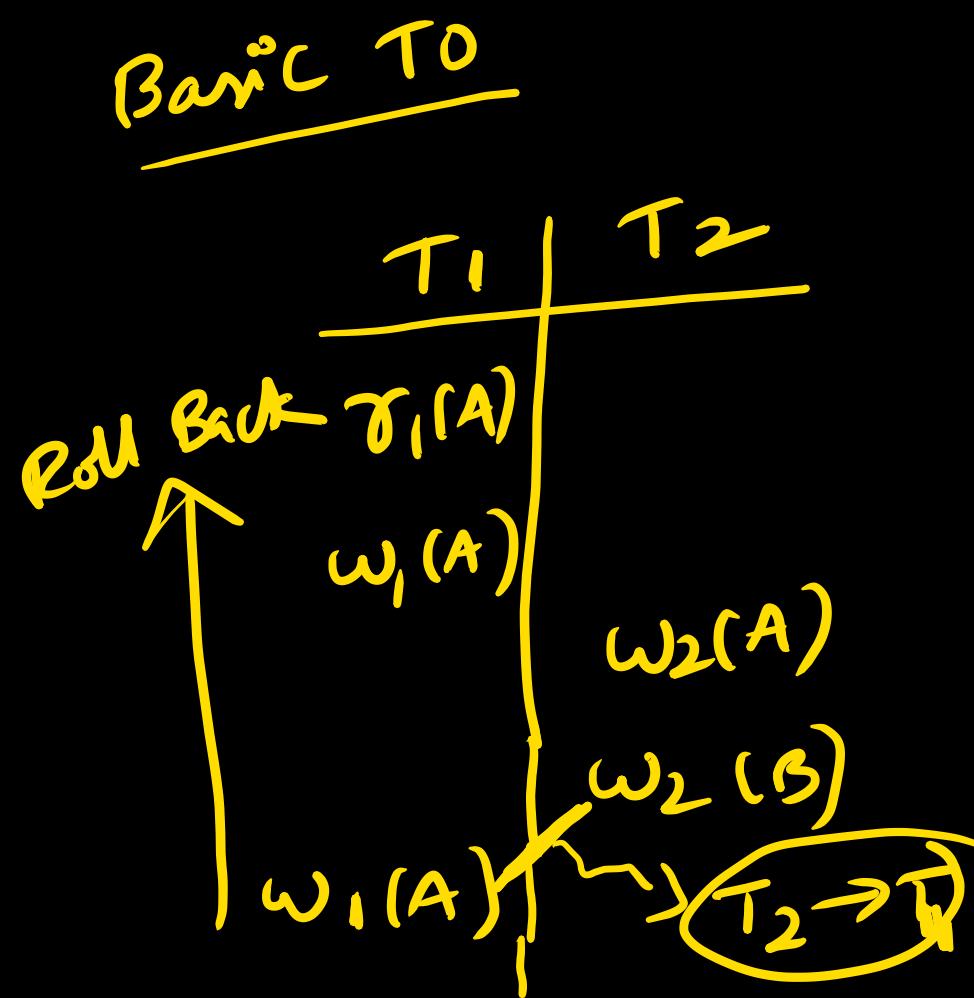
$T_1 \rightarrow T_2 \rightarrow T_3$
 $T_1 \rightarrow T_3$
 $T_2 \rightarrow T_3$

Thomas with TS ordering



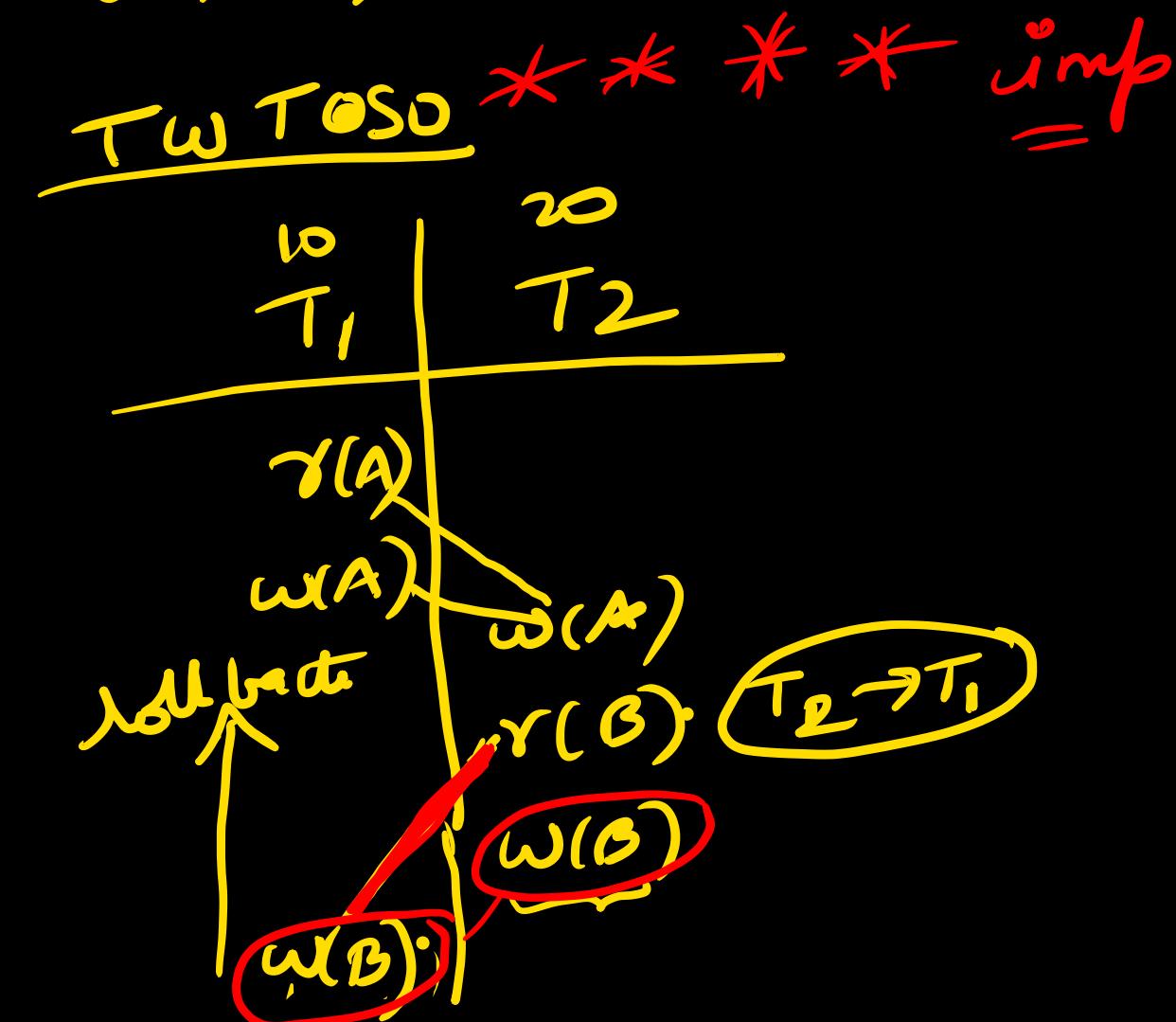
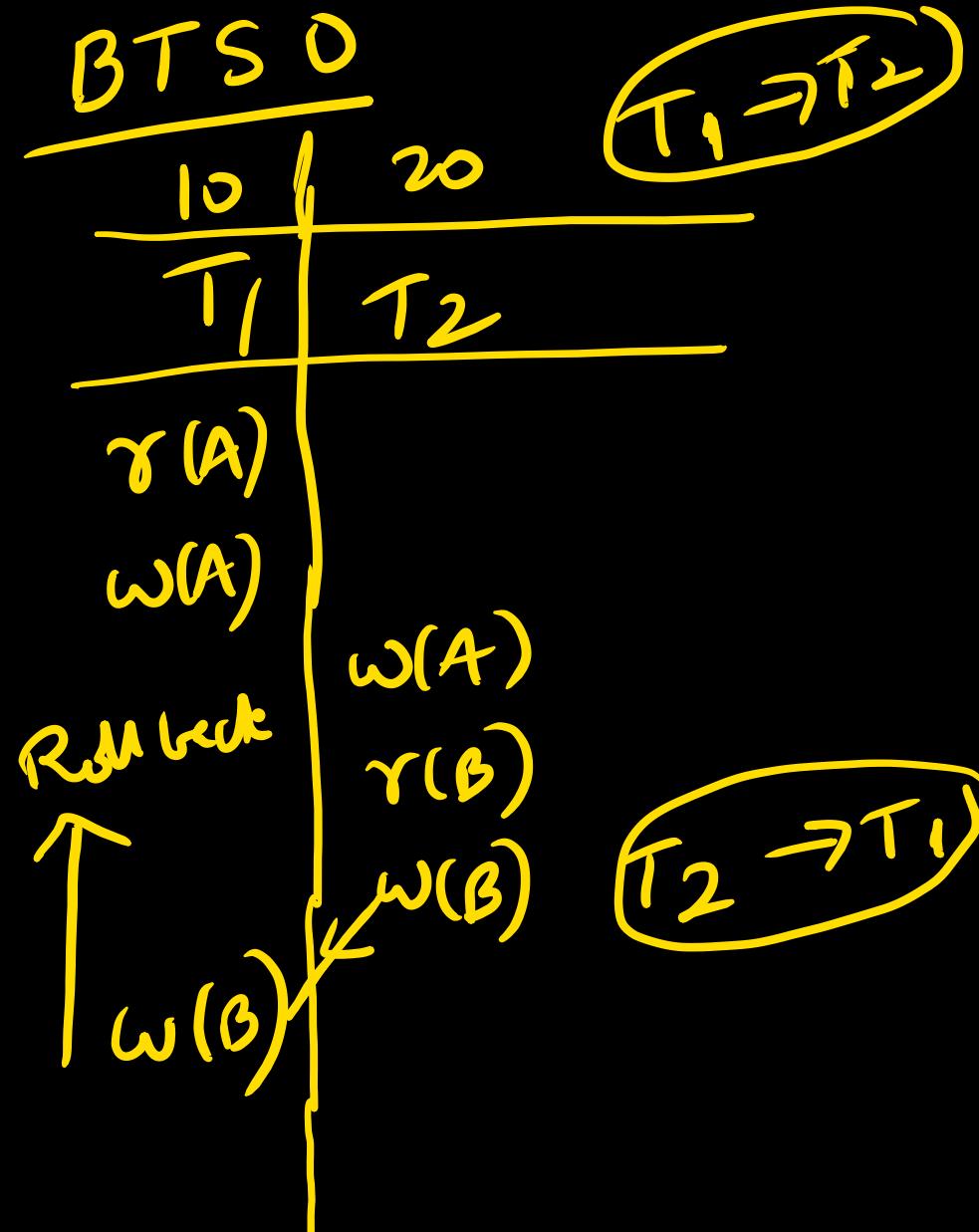
No RB.

$S: \gamma_1(A) \omega_1(A) \omega_2(A) \omega_2(B) \omega_1(B)$ $TS\ val(T_1, T_2) = (10, 20)$

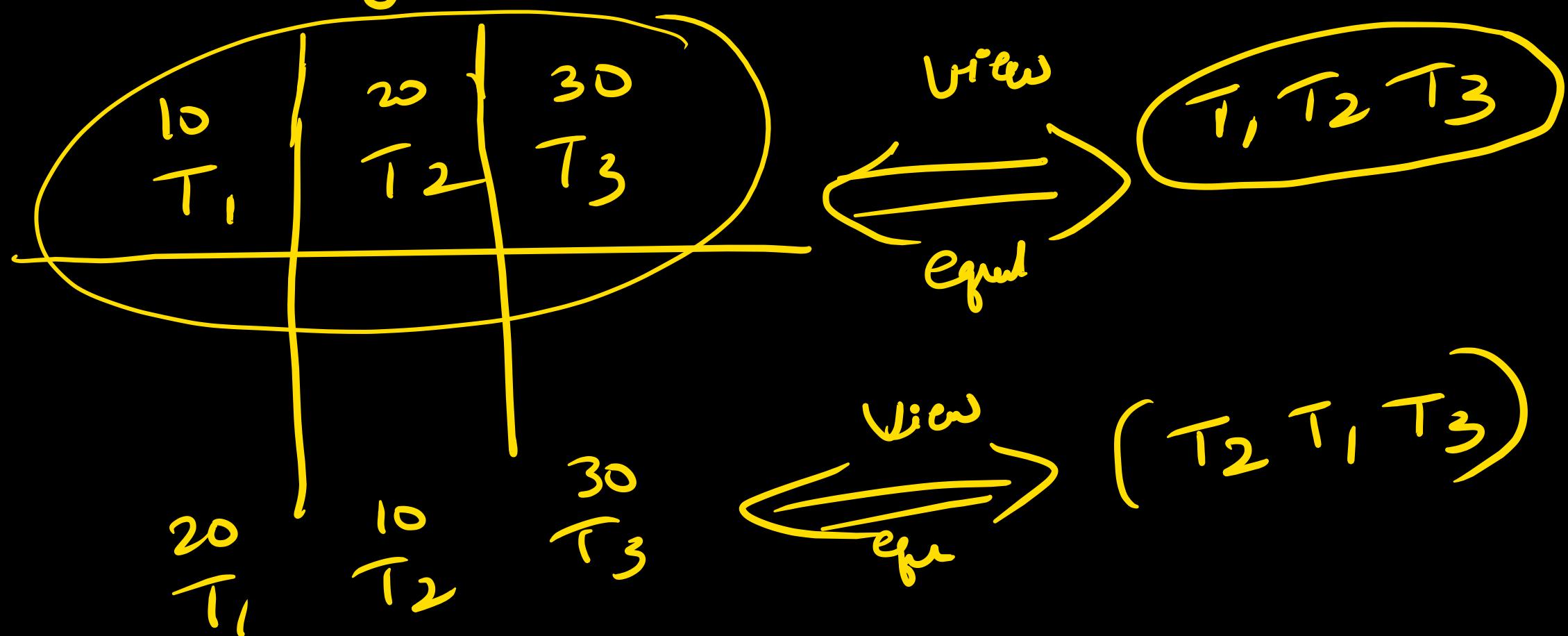


$S: \gamma_1(A) \omega_1(A) \omega_2(A) \gamma_2(B) \omega_2(B) \omega_1(B)$

TS val $(T_1, T_2) = (10, 20)$

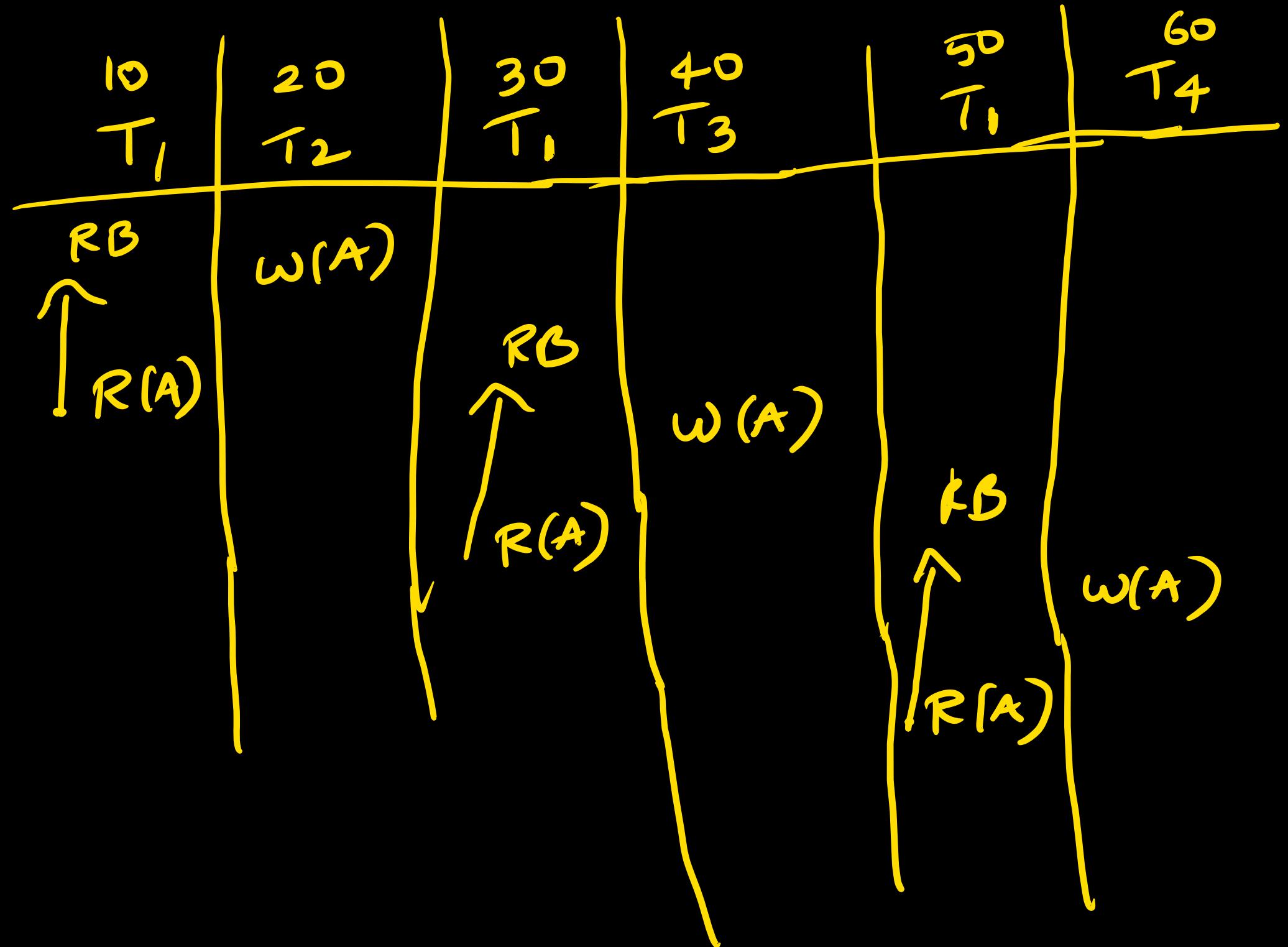


Thomas with TS ordering protocol allows a schedule (S) only if
the result of schedule (S) is view equal to serial schedule based
on TS ordering



BTSO/TWTSO protocol:

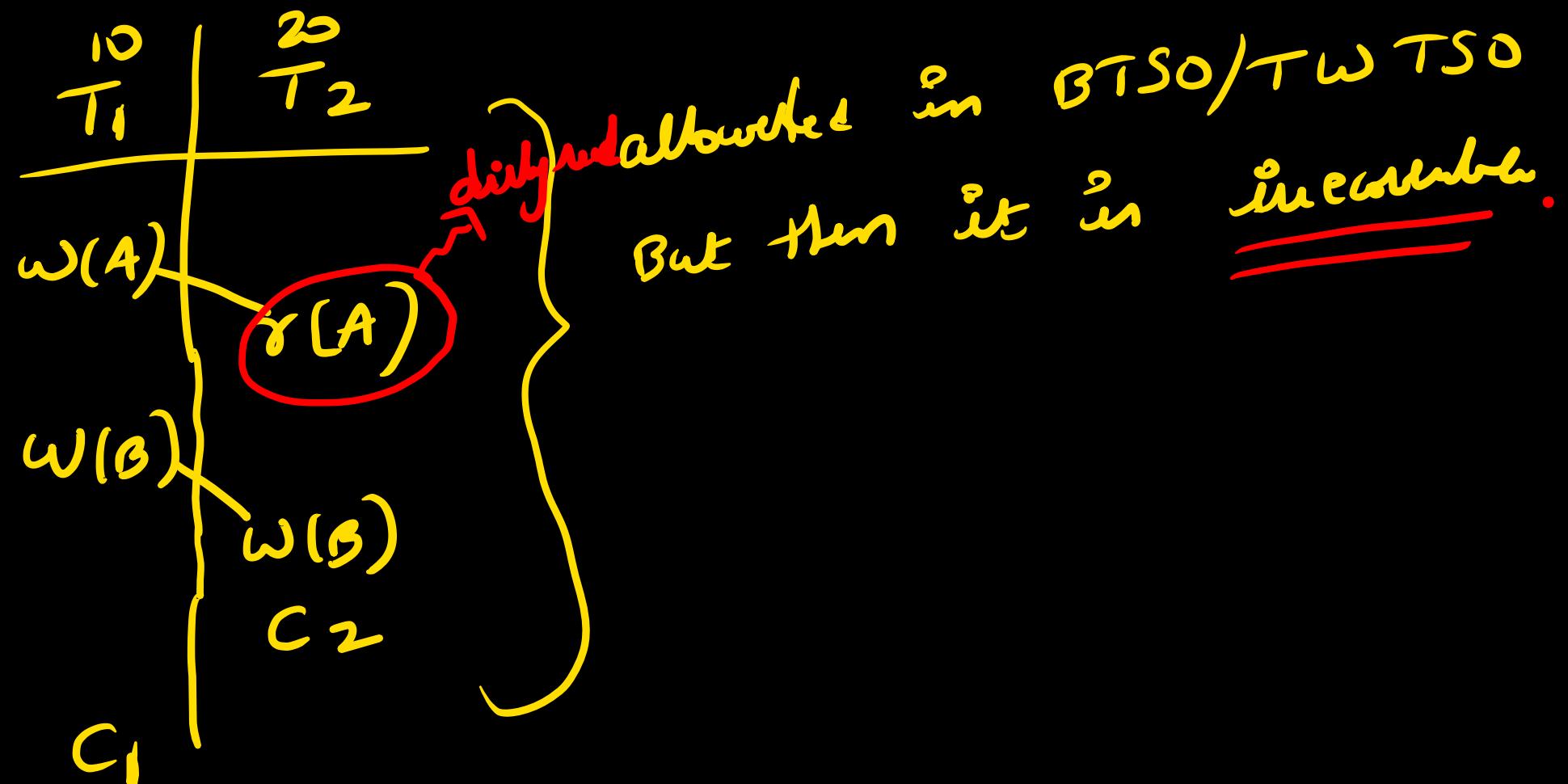
- Guarantees Serializability
- Dead lock free (no locking is used)
- not free from starvation



T_1 may now get
a chance.
is possible

→ BTSO/TWTSO: may have incredible problem, Cascade RD
problem and lock update pattern.

Ex: $S: \omega_1(A) \gamma_2(A) \omega_1(B) \omega_2(B) C_2 C_1$

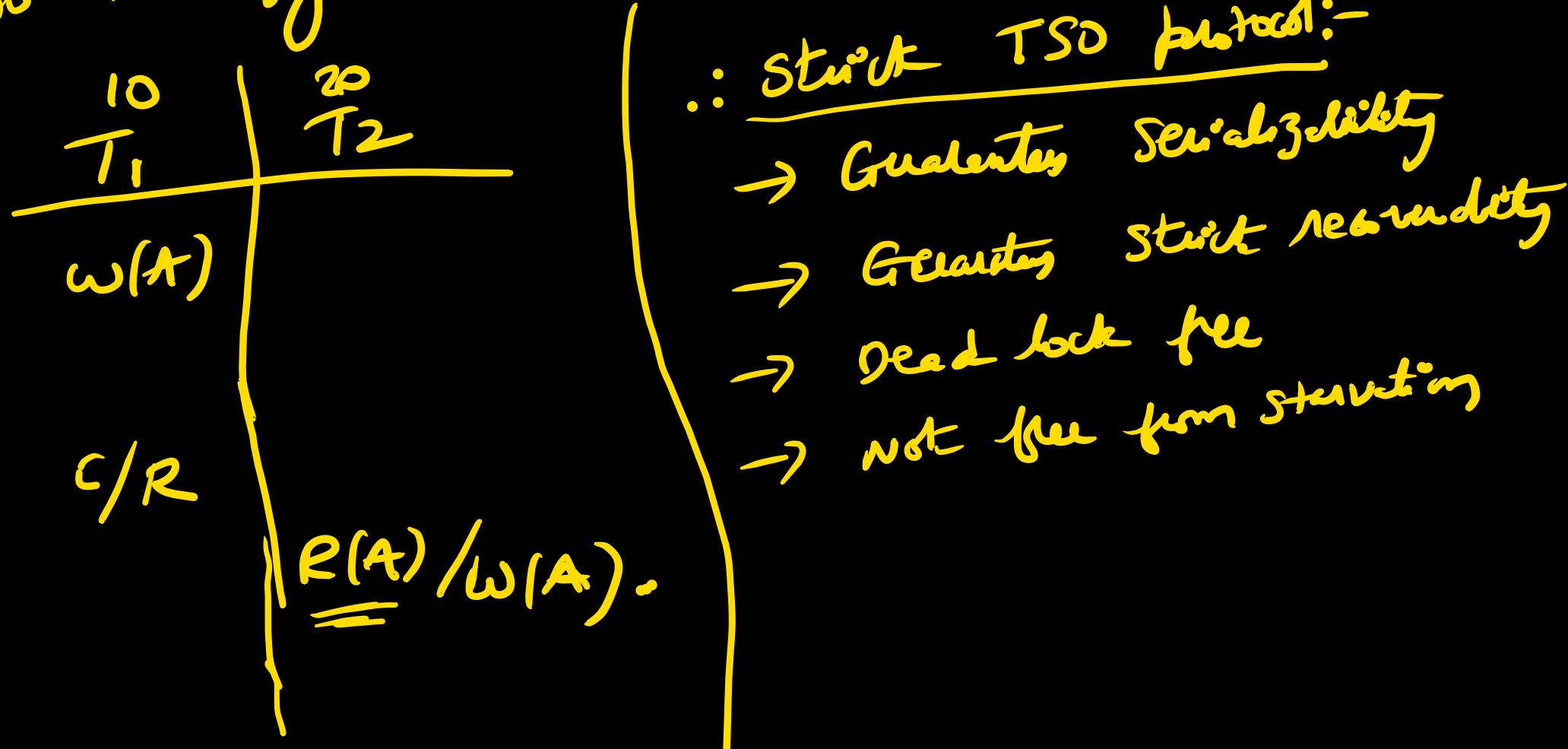


Strict TS ordering protocol:-

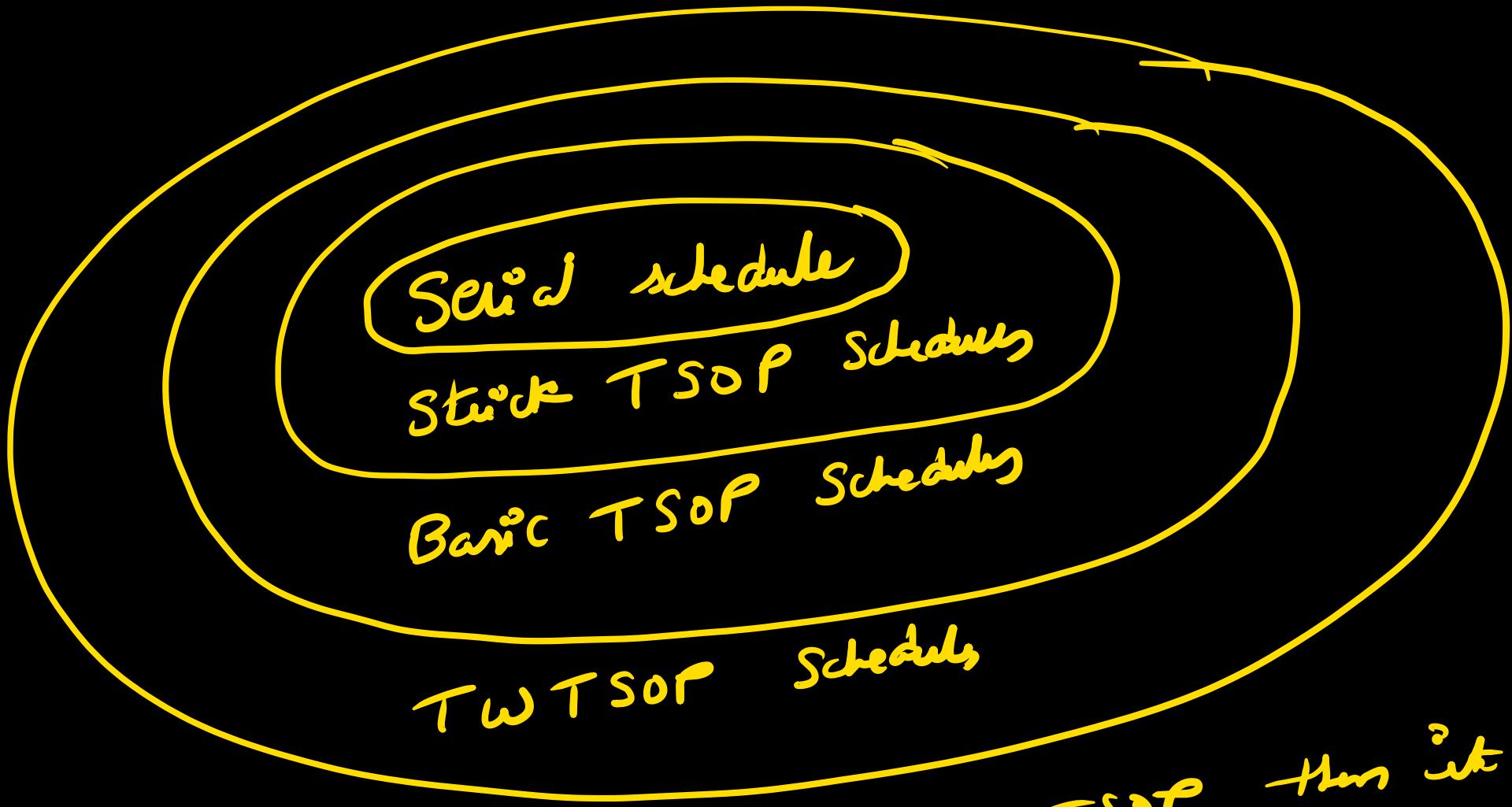
→ must follow same rules of basic time stamp ordering protocol

(and)

→ also satisfy strict recoverable scheduling



P



If ~~an~~ a schedule is executed in TwTSOP, then it is
executed in Basic(TSOP) also (may change).



DBMS is over.

Cover everything in depth.

99% questions will be from this notes.
=====