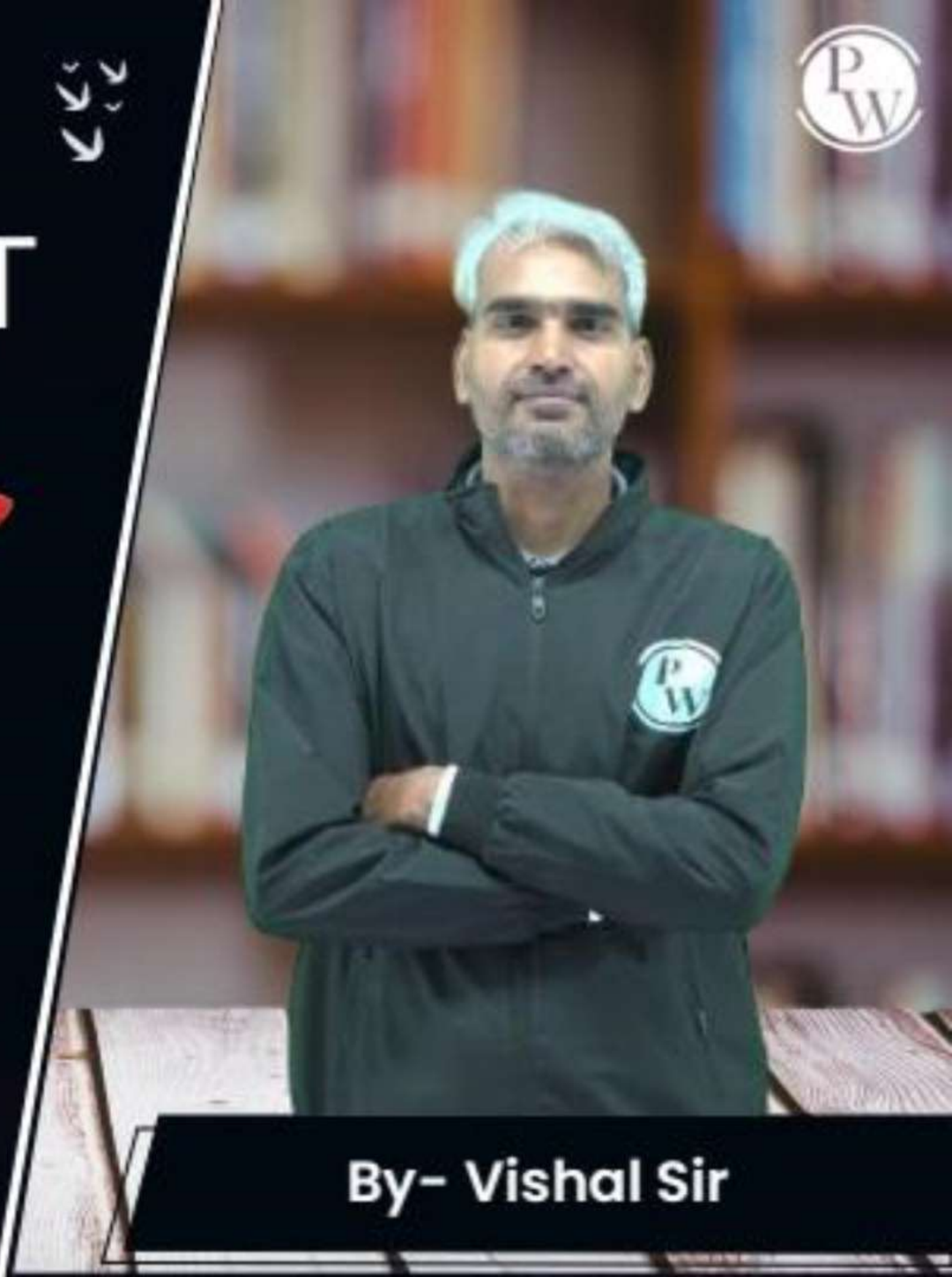
Computer Science & IT

Database Management System

Transaction & Concurrency control

Lecture No. 09



By- Vishal Sir

Recap of Previous Lecture

- **Topic** View equivalence condition
- **Topic** Practice questions on View serializable schedule
- **Topic** Concurrency control protocols

Slide

Topics to be Covered

- Topic: Simple use of shared and exclusive locks
- Topic: Basic Two phase lock (2PL)
- Topic: Basic 2PL and lock upgrading/downgrading
- Topic: Problems possible with Basic 2PL
- Topic: Strict 2PL

Slide

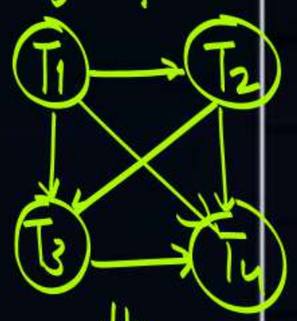
H.W.

#Q. Check whether the schedule is view serializable schedule or not?

If view serializable schedule then identify all view equivalent serial schedules.

T1	T2	T3	T4
R(A)			
	R(A)		
		R(A)	
			R(A)
W(B)			
	W(B)		
		W(B)	
			W(B)

Precedence graph



↓ Topological order

$= T_1 \to T_2 \to T_3 \to T_4$

∴ Conflict equivalent serial schedule

① Initial Read
A: No Constraint
B: No Constraint

② WR Seq:
A: No WR Seq ✓
B: No WR Seq ✓

③ Final Update
A: No Write
B: $(T_1, T_2, T_3) \to T_4$

Overall Constraint
$(T_1, T_2, T_3) \to T_4$
$3! = 6$ ways to arrange T_1, T_2 & T_3 before T_4.
∴ 6 view equivalent serial schedules

#Serial schedules that are view equivalent to given schedule, but not conflict equivalent
= No. of view equivalent serial schedule − No. of conflict equivalent serial schedule
$= 6 - 1 = 5$

H.W.

#Q. Check whether the schedule is view serializable schedule or not?

If view serializable schedule then identify all view equivalent serial schedules.

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

Precedence graph

Cycle

∴ Not a C.S.S.

① Initial Read

A: $T_1 \rightarrow (T_2, T_3)$ ≡ $T_1 \rightarrow T_2$, $T_1 \rightarrow T_3$, $T_3 \rightarrow T_2$

$T_3 \rightarrow T_2$

B: $T_2 \rightarrow$ No Constraint

② Write: Read seq.

A: $T_3 \rightarrow T_2$

$T_k = \phi$

B: $T_2 \rightarrow T_1$

$T_2 \rightarrow T_3$

$T_k = \phi$

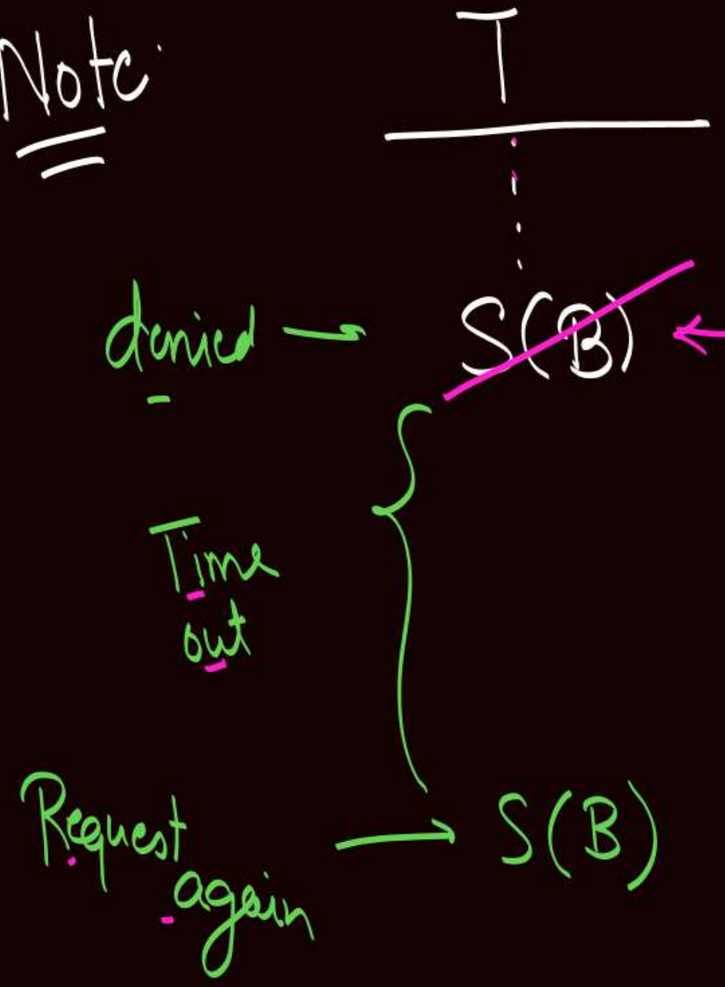
∴ Cycle. Hence Not a V.S.S.

* Note:- To ensure serializability a non-serializable schedule must not be allowed to execute

If we say that a specific protocol ensures serializability it means a non-serializable schedule is never allowed using that protocol. {But it does not mean that all serializable schedules will be allowed to execute by that protocol}

They may or may not be allowed but a non-serializable schedule is never allowed

Note:



if transaction request for a lock on a dataitem and if that lock is denied at that point, then that transaction will go in a time-out period, and it will request again once the time-out period is over

Topic : Simple use of shared and exclusive locks

Simple use → i.e. no restriction { i.e. a transaction can lock and/or unlock the dataitems at any point }

'S' is a non-serializable schedule, but it is allowed to execute using Shared & Exclusive locks

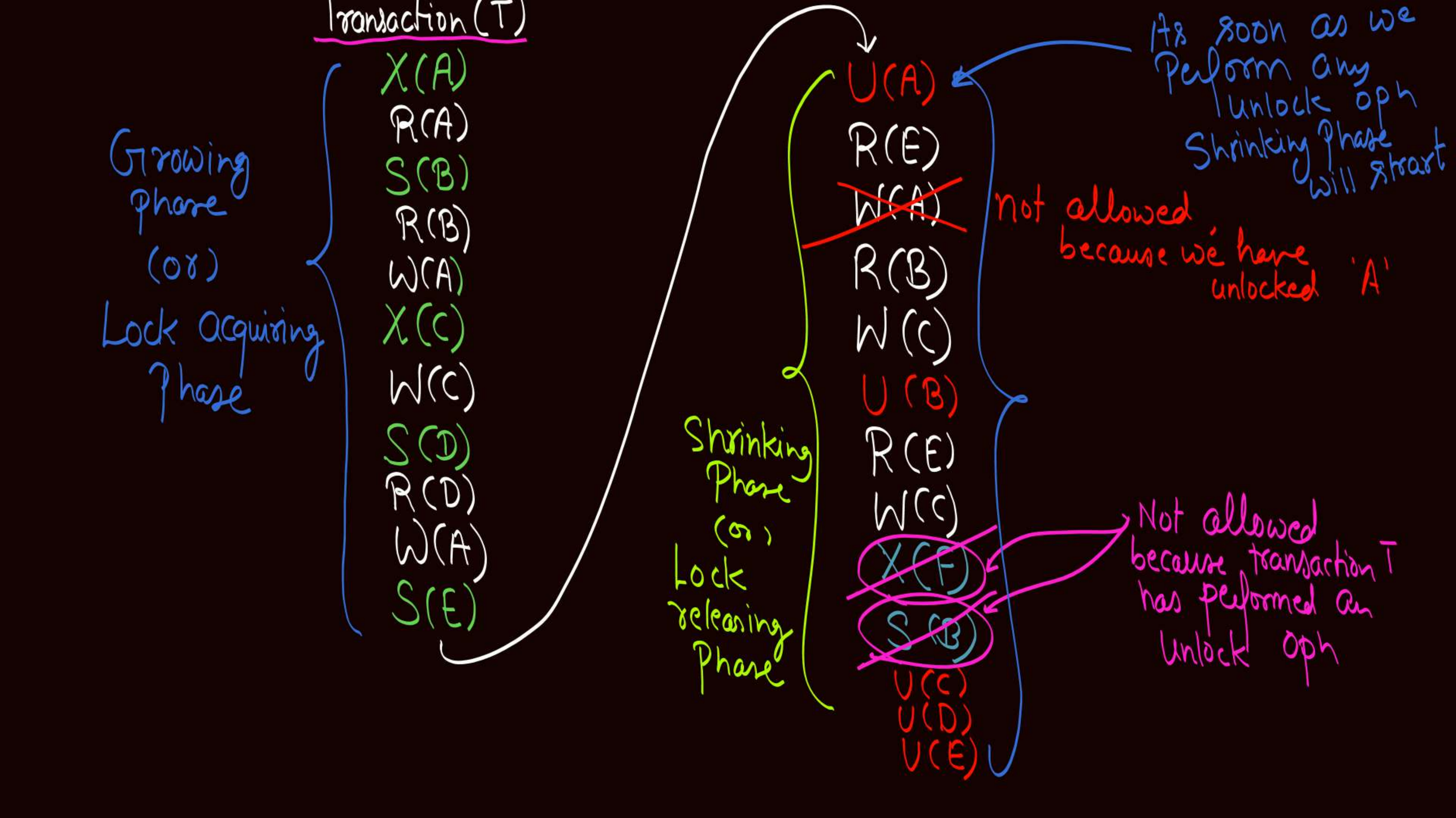
∴ Simple use of Shared and Exclusive lock does not Ensure Serializability

Topic : Basic Two Phase Locking Protocol

There are two phases in Two phase locking protocol

1. Growing phase / Lock acquiring phase
2. Shrinking phase / Lock releasing phase

In Two Phase Locking Protocol, transaction T is allowed to request for a lock (on any data item) only if transaction T has not performed any unlock operation.

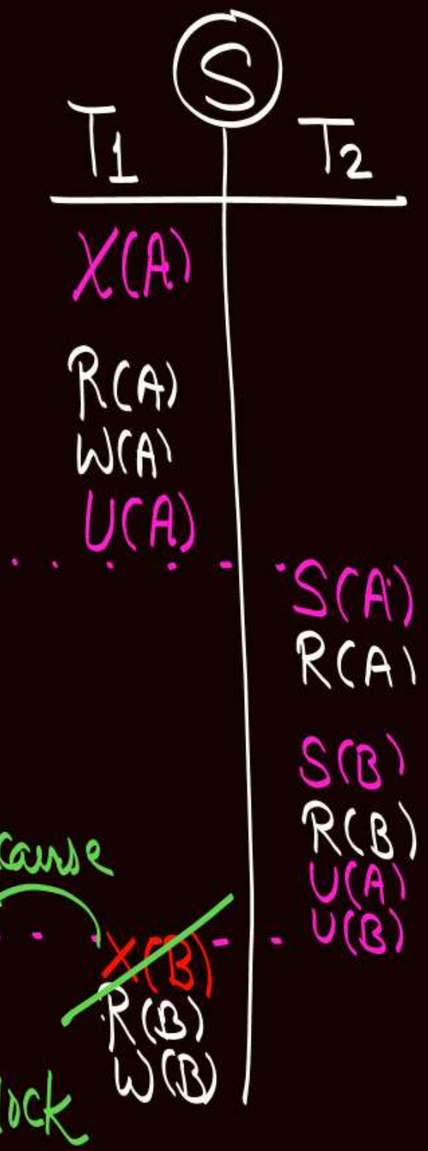

Transaction (T)
X(A)
R(A)
S(B)
R(B)
W(A)
X(C)
W(C)
S(D)
R(D)
W(A)
S(E)
Growing Phase (or) Lock acquiring Phase
U(A)
R(E)
W(A)
R(B)
W(C)
U(B)
R(E)
W(C)
X(F)
S(B)
U(C)
U(D)
U(E)
Shrinking Phase (or) Lock releasing Phase
As soon as we perform any unlock oph Shrinking Phase will start
not allowed because we have unlocked 'A'
Not allowed because transaction T has performed an unlock oph

Q: Check whether the given schedule is allowed to execute using basic two phase locking protocol or not

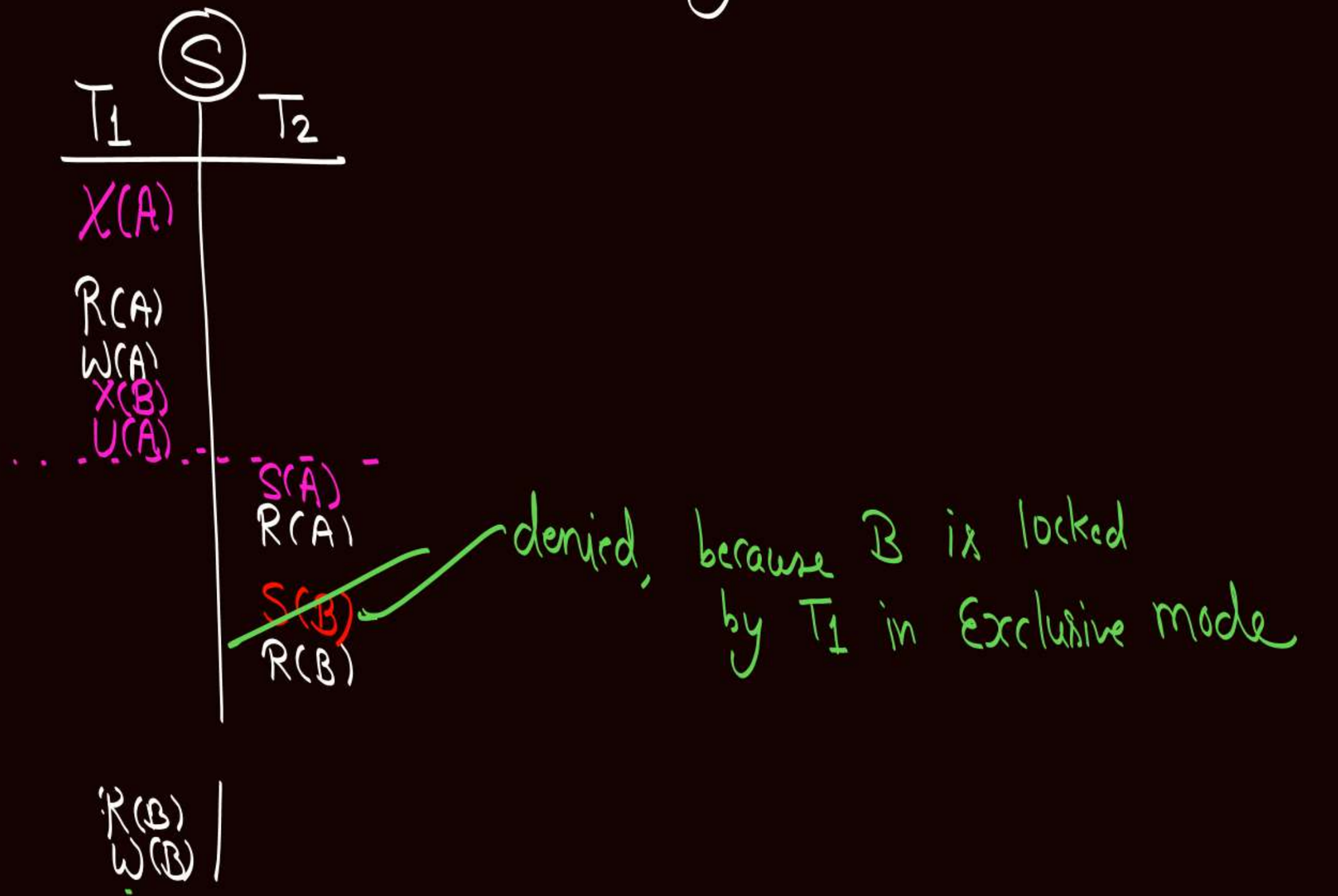
'A' is already locked by T_1 in Exclusive mode

denied ∴ it will go in time out

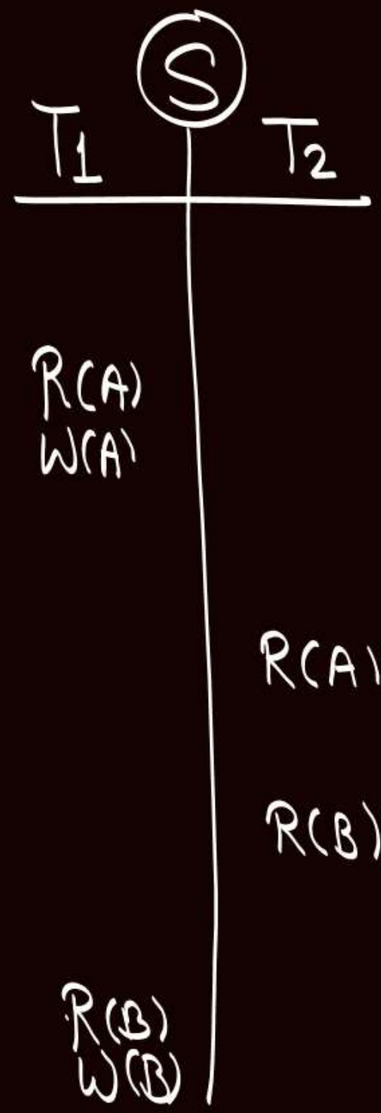
Q: Check whether the given schedule is allowed to execute using basic two phase locking protocol or not



Q: Check whether the given schedule is allowed to execute using basic two phase locking protocol or not



Q: Check whether the given schedule is allowed to execute using basic two phase locking protocol or not

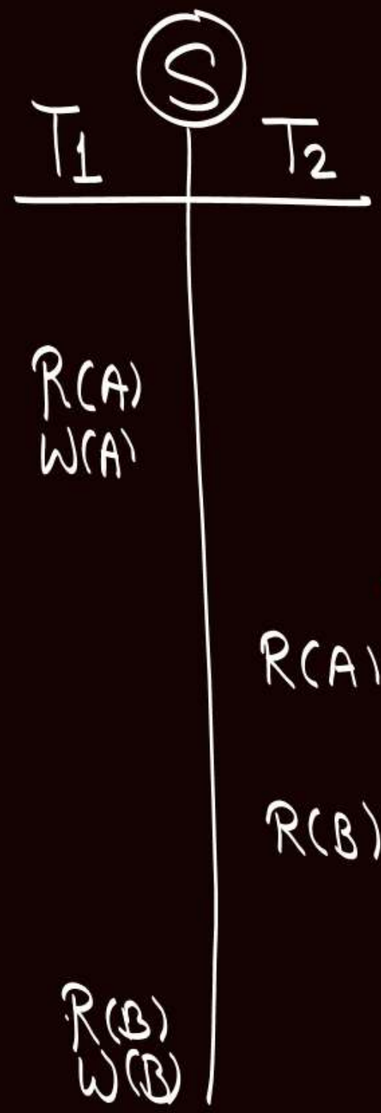


Given schedule is a non-serializable schedule, and it is not allowed by 2PL

Topic : Basic Two Phase Locking Protocol

1. 2PL ensures serializability, i.e., a non-serializable schedule is never allowed to execute using 2PL.
2. If schedule is allowed to execute using 2PL, then schedule is conflict serializable schedule, but every conflict serializable schedule need not be allowed to execute by 2PL. { i.e. there are some conflict serializable schedules which are not allowed by 2PL.
3. If schedule is not a conflict serializable schedule, then that schedule is not allowed to execute using 2PL. { does not matter whether the schedule is view serializable schedule or not }

Q: Check whether the given schedule is allowed to execute using basic two phase locking protocol or not



Given schedule is a non-serializable schedule, and it is not allowed by 2PL

Note:- A non-serializable schedule is never allowed to execute using 2PL

eg:

S_1

T_1	T_2
X(A)	
R(A)	
W(A)	
X(B)	
U(A)	
	S(A)
	R(A)
R(B)	
W(B)	
U(B)	
	S(B)
	R(B)
	U(A)
	U(B)

Given schedule is allowed to execute using basic 2PL.

"and"

it is a Conflict serializable schedule

Precedence graph (Acyclic)

Basic 2PL ensures Conflict serializability

⇓

i.e. a non-conflict serializable schedule is never allowed to execute using basic 2PL

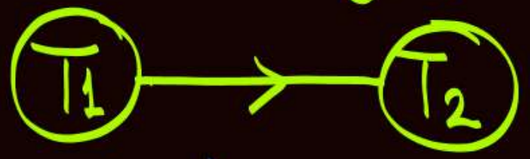
⇓

If schedule is allowed by basic 2PL, then it is guaranteed to be Conflict serializable, but Converse of the statement need not be true

eg: S_2

T_1	T_2
	X(B)
	R(B)
S(A)	
R(A)	
S(B) denied because of T_2	
R(B)	
	W(B)

Precedence graph



Acydic

∴ Schedule is a Conflict serializable schedule

Given schedule is a Conflict serializable schedule but not allowed by basic 2PL

eg: S_3

T_1	T_2
X(B)	
R(B)	
X(B) denied bcoz of T_1	
W(B)	
	W(B)

Precedence graph

Cyclic.

∴ Not a C.S.S.

but S_3 is Equivalent to serial schedule $T_1 \rightarrow T_2$

Hence, S_3 is a serializable schedule

Schedule is not a Conflict serializable schedule (but still a serializable schedule) and not allowed by basic 2PL

If schedule is not a C.S.S. then it is never allowed to execute using basic 2PL

Topic : Basic Two Phase Locking Protocol

1. 2PL ensures serializability, i.e., a non-serializable schedule is never allowed to execute using 2PL.

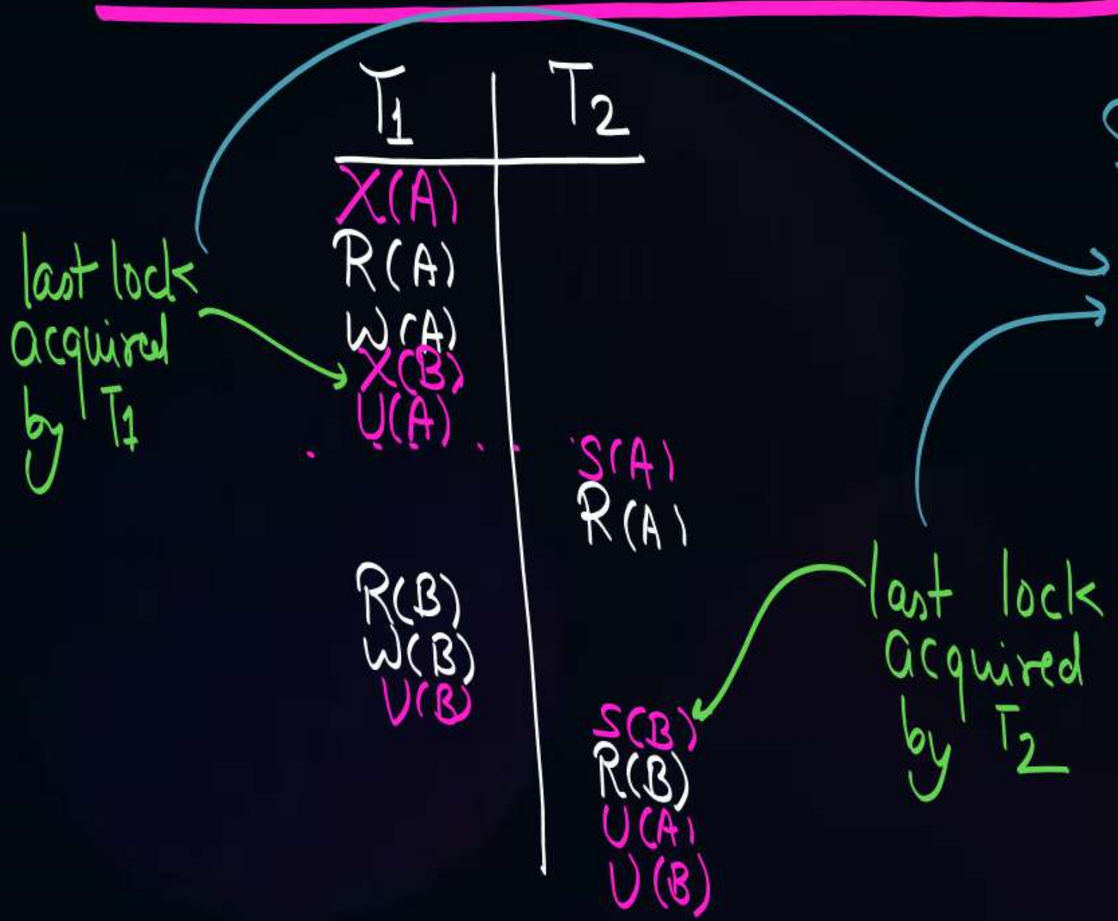
{Precisely, 2PL ensures Conflict serializability}

[redacted]

4. If schedule is allowed to execute using 2PL, then schedule is conflict serializable schedule and conflict equivalent serial schedule can be given by order of lock points.

Topic : Lock Point

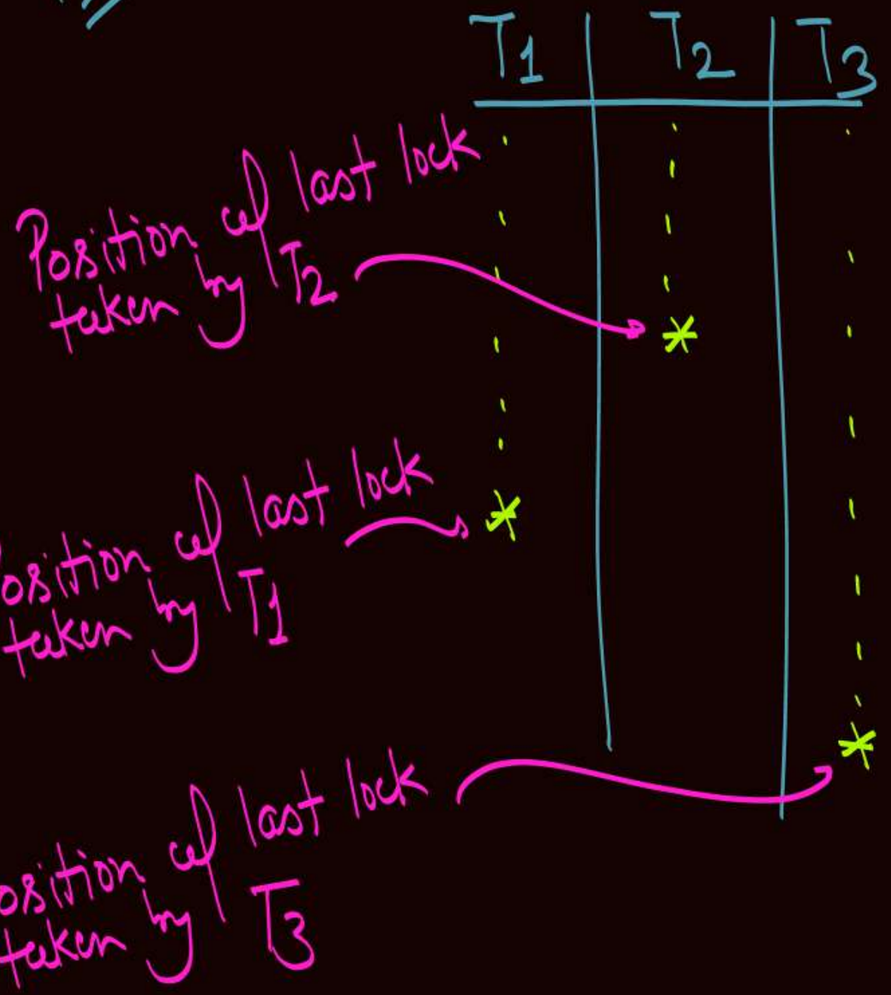
The Point at which the growing phase ends, i.e., the point at which transaction takes the final lock it needs to carry on its work.



Schedule is allowed by basic 2PL
∴ Schedule is a Conflict Serializable Schedule
"and"
Order of lock points is
T_1 then T_2
∴ Conflict equivalent Serial Schedule is $T_1 \rightarrow T_2$

Note:-

Schedule 'S'



If schedule 'S' is allowed to execute by basic 2PL, and order of lock points is as shown,

then one of the Conflict Equivalent serial schedule corresponding to given schedule will be

$T_2 \rightarrow T_1 \rightarrow T_3$ { w.r.t. order of lock points }

Topic : Basic 2PL with lock upgrading

from shared to exclusive

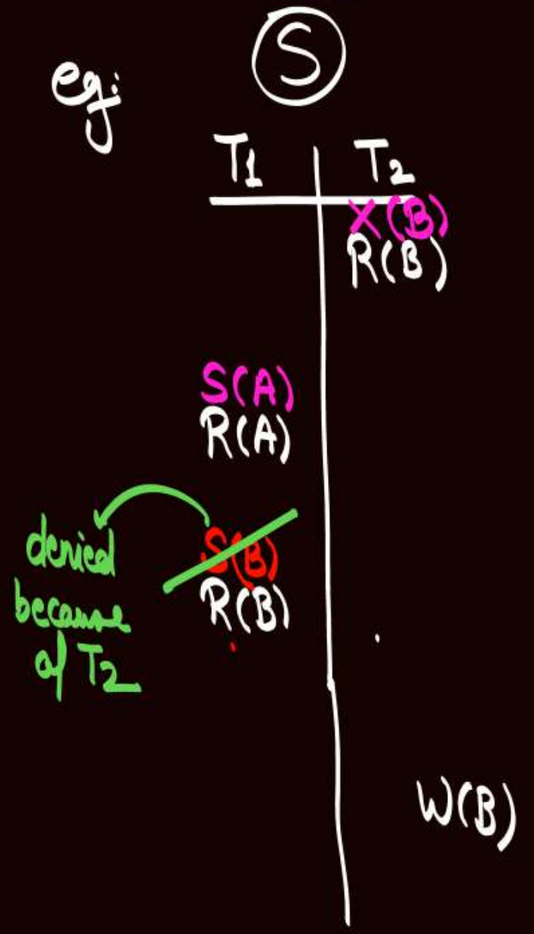
from Exclusive to shared.

If lock conversion (upgrading/downgrading) is allowed, then we can upgrade shared lock into exclusive locks in the Growing Phase, and we can downgrade from the exclusive lock to shared lock in the shrinking phase.

* If we are in shrinking phase then we can not upgrade from shared to Exclusive lock.
- If we are already in the shrinking phase then we can not upgrade but we can downgrade, and if we are in the growing phase then we can perform the downgrade operation but shrinking phase of the transaction starts from that downgrade opn itself.

2PL without lock upgrading

eg:



Precedence graph

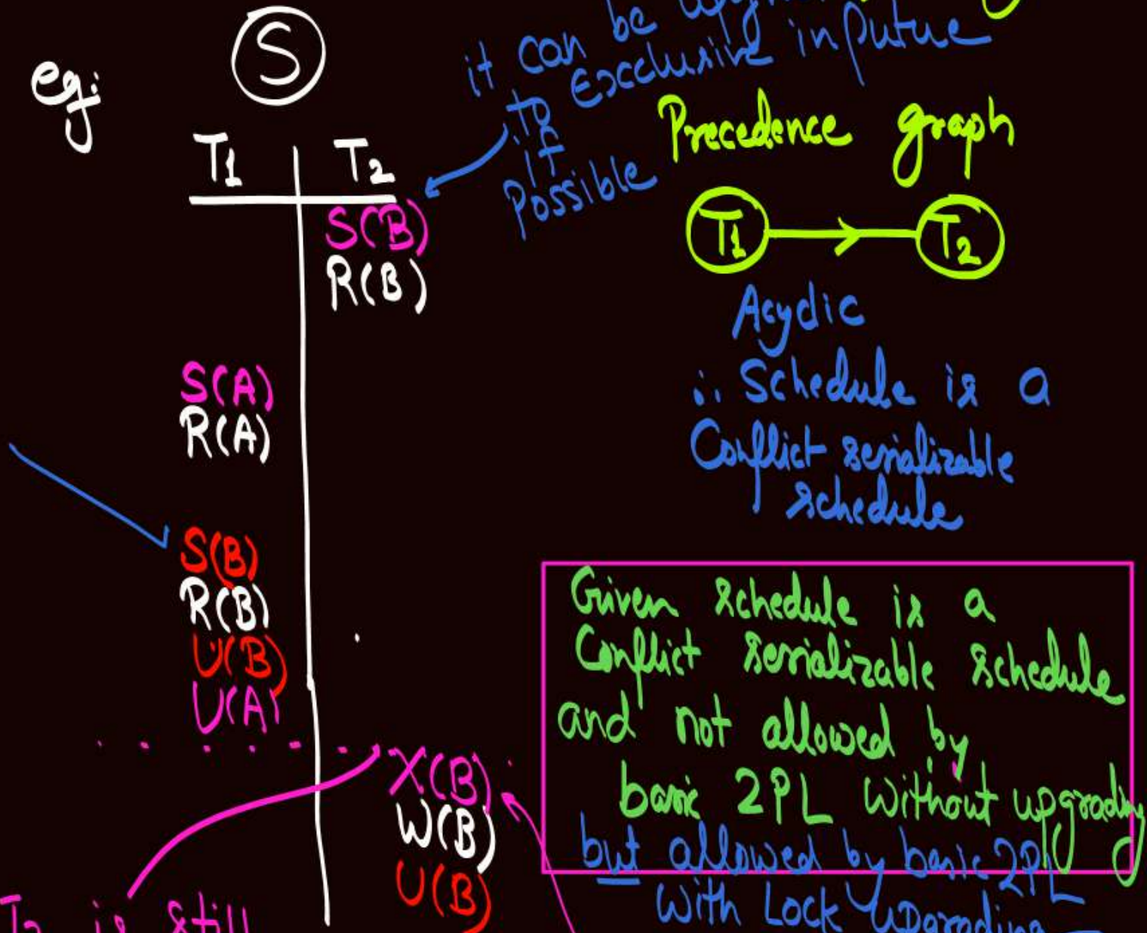
Acydic

∴ Schedule is a Conflict serializable schedule

Given schedule is a Conflict serializable schedule but not allowed by basic 2PL without lock upgrading

2PL with Lock upgrading

eg:



Precedence graph

Acydic

∴ Schedule is a Conflict serializable schedule

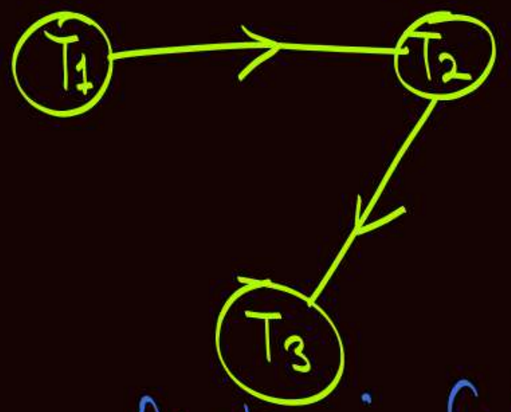
Given schedule is a Conflict serializable schedule and not allowed by basic 2PL without upgrading but allowed by basic 2PL with Lock upgrading

HW. eg.

Schedule S

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

Precedence graph



Acyclic ∴ C.S.S.

H.W. Q.1. Check whether the Schedule is allowed to execute using Basic 2PL without lock upgrading or not

H.W. Q.2 Check whether the Schedule is allowed to execute using Basic 2PL with lock upgrading or not

2 mins Summary

- Topic: Simple use of shared and exclusive locks
- Topic: Basic Two phase lock (2PL)
- Topic: Basic 2PL and lock upgrading/downgrading
- Topic: Problems possible with Basic 2PL
- Topic: Strict 2PL

THANK - YOU