TECH VAULT

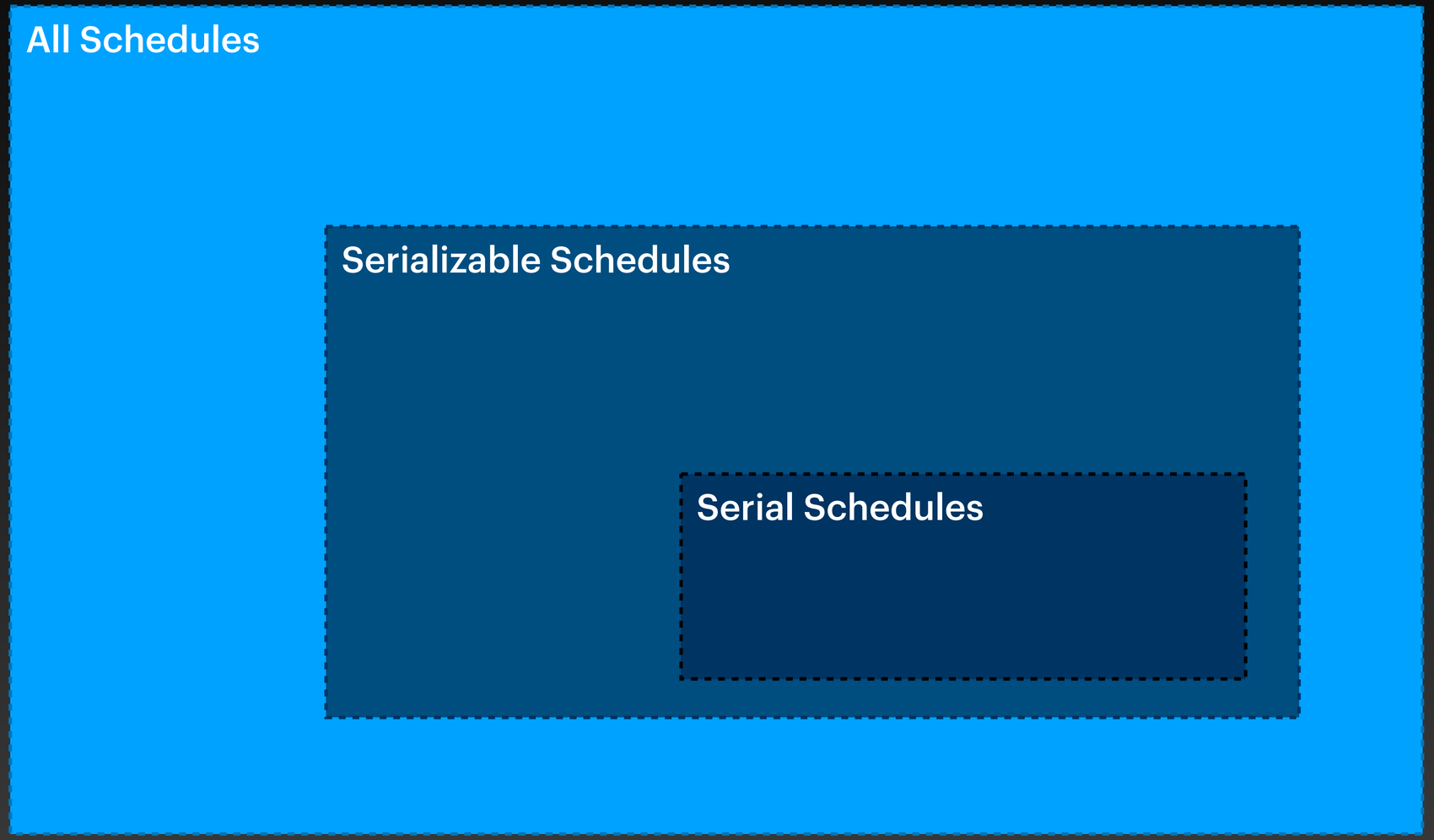
CONCURRENCY CONTROL

Two-Phase Locking

Part (1)

AMR ELHELW

Schedule Serializability



Begin

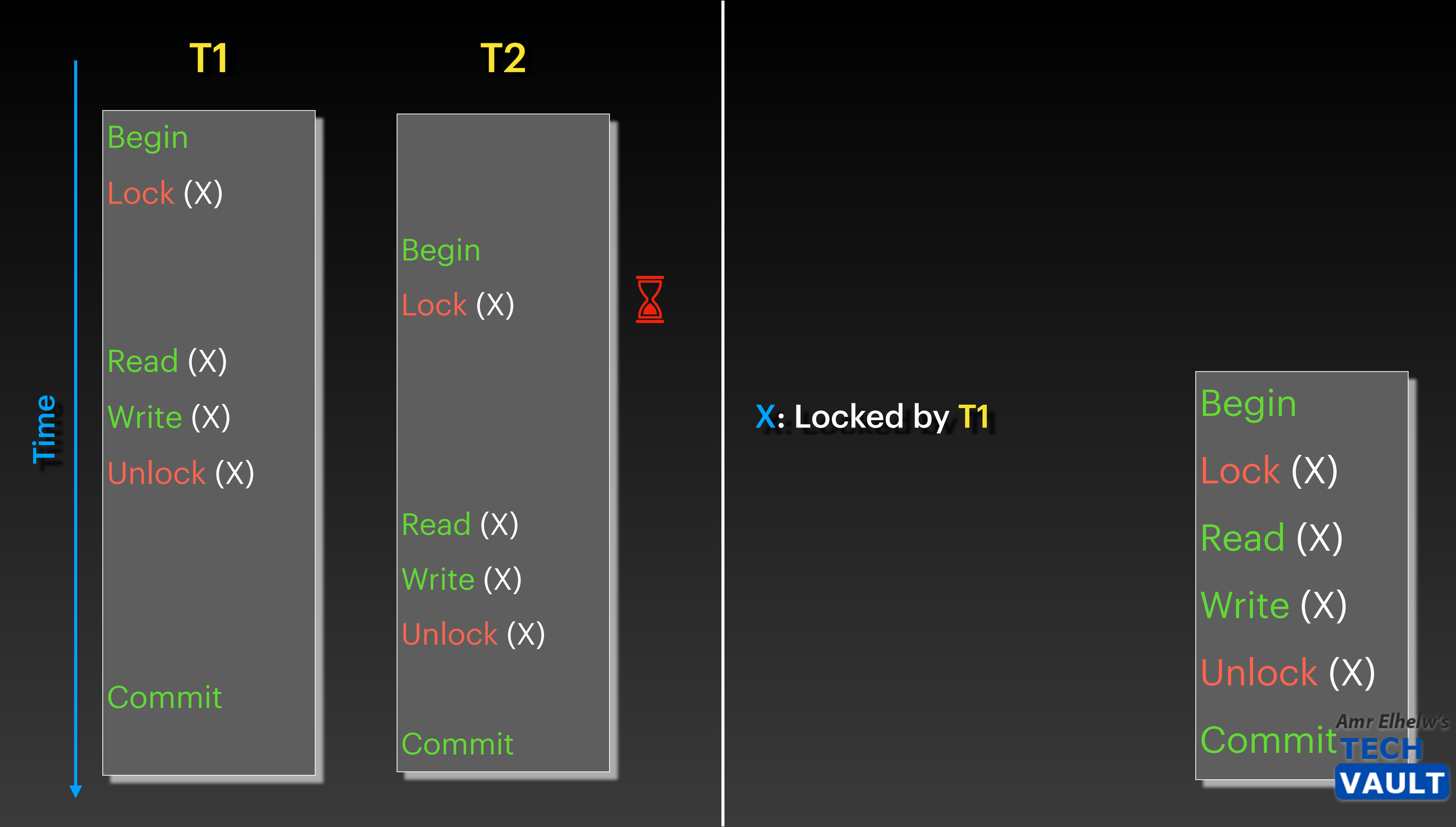
Lock (X)

Read (X)

Write (X)

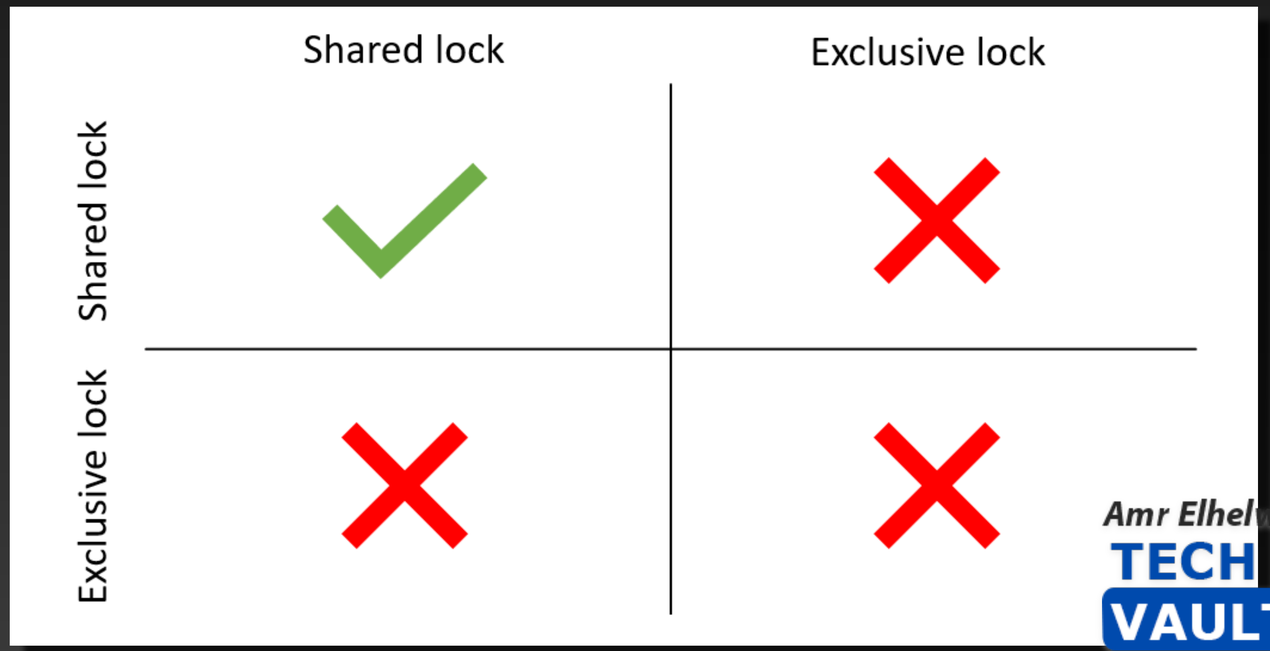
Unlock (X)

Commit


T1
T2
Time
Begin
Lock (X)
Read (X)
Write (X)
Unlock (X)
Commit
Begin
Lock (X)
Read (X)
Write (X)
Unlock (X)
Commit
X: Locked by T1
Begin
Lock (X)
Read (X)
Write (X)
Unlock (X)
Commit
Amr Elhelw's
TECH
VAULT

- Shared lock (S-Lock)
  - *a.k.a. Read Lock*
- eXclusive lock (X-lock)
  - *a.k.a. Write Lock*

Compatibility Matrix



X-Lock (A)

Write (A)

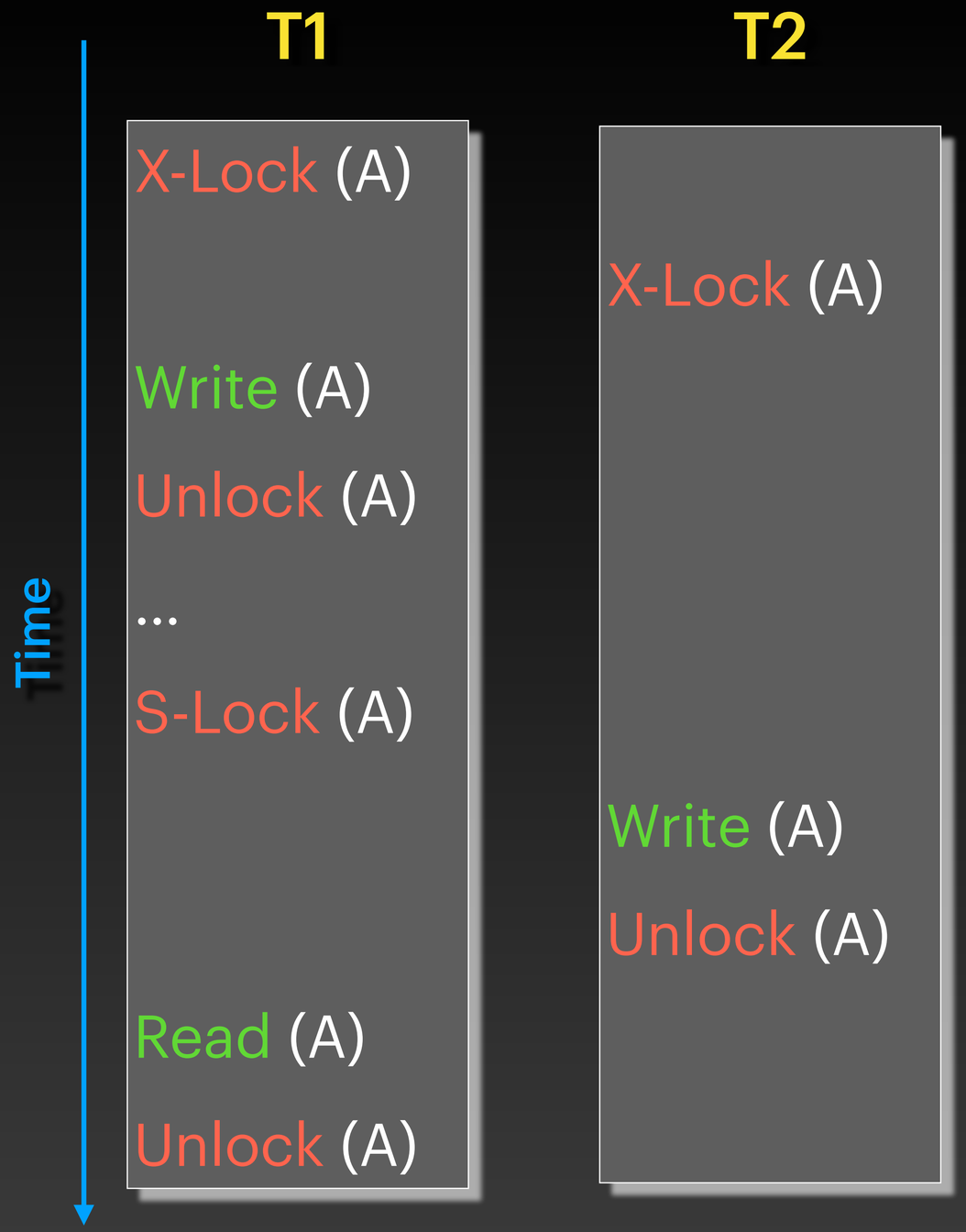
Unlock (A)

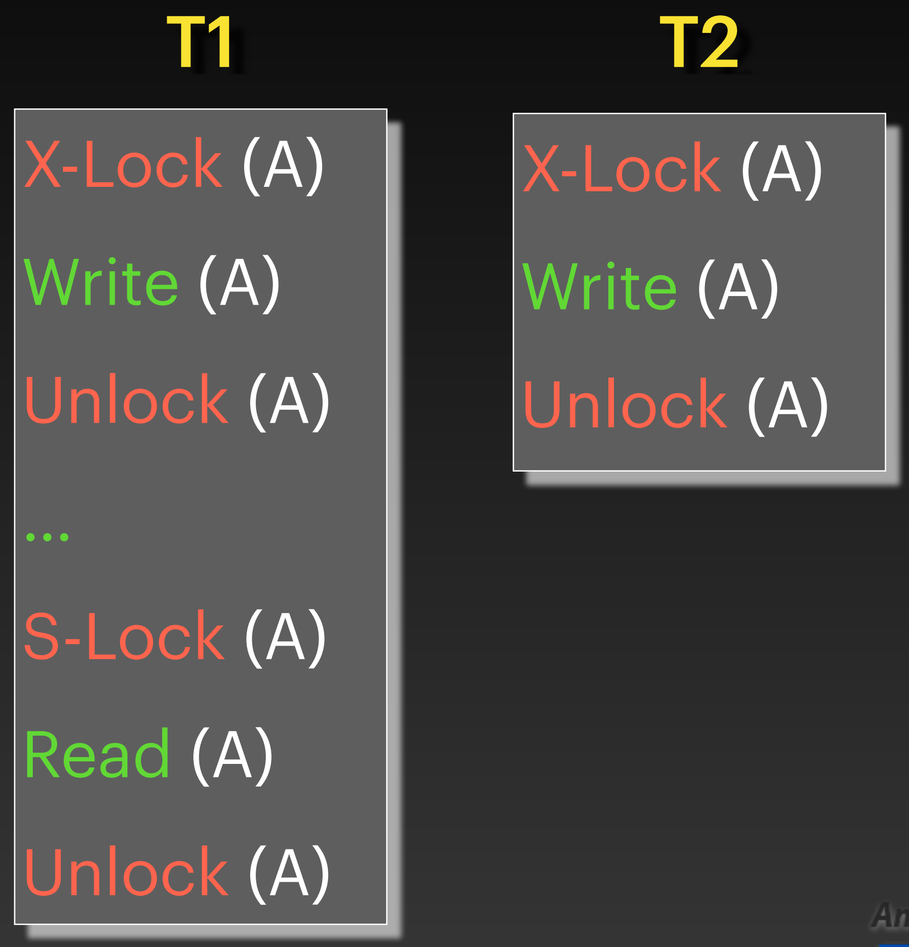
...

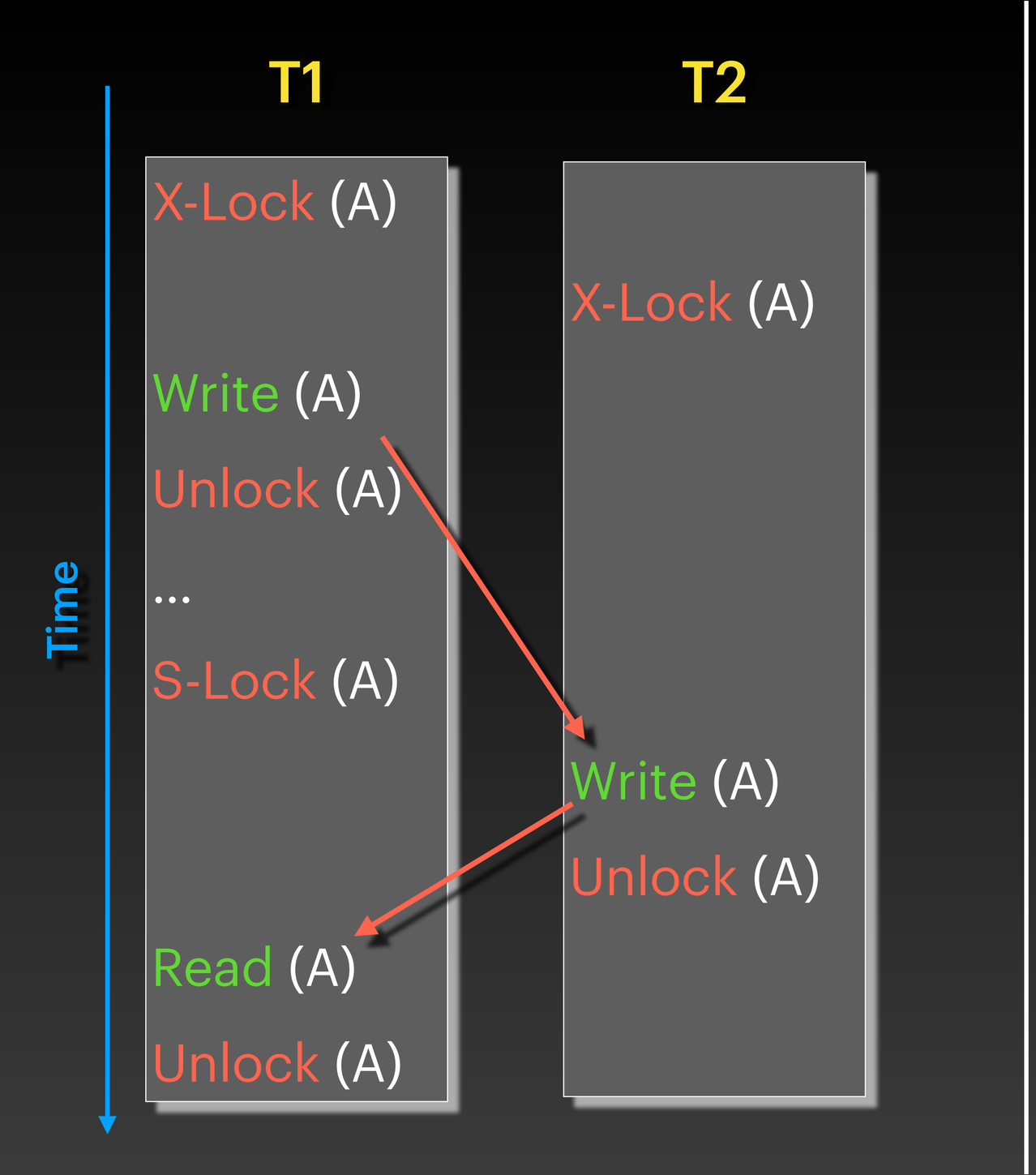
S-Lock (A)

Read (A)

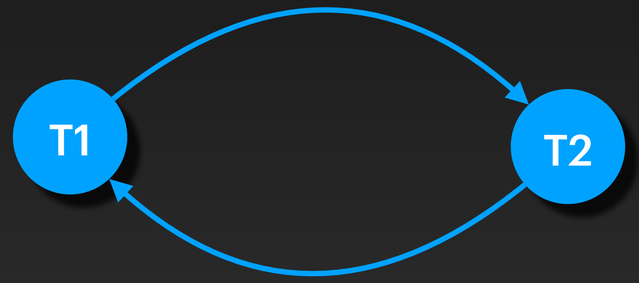
Unlock (A)



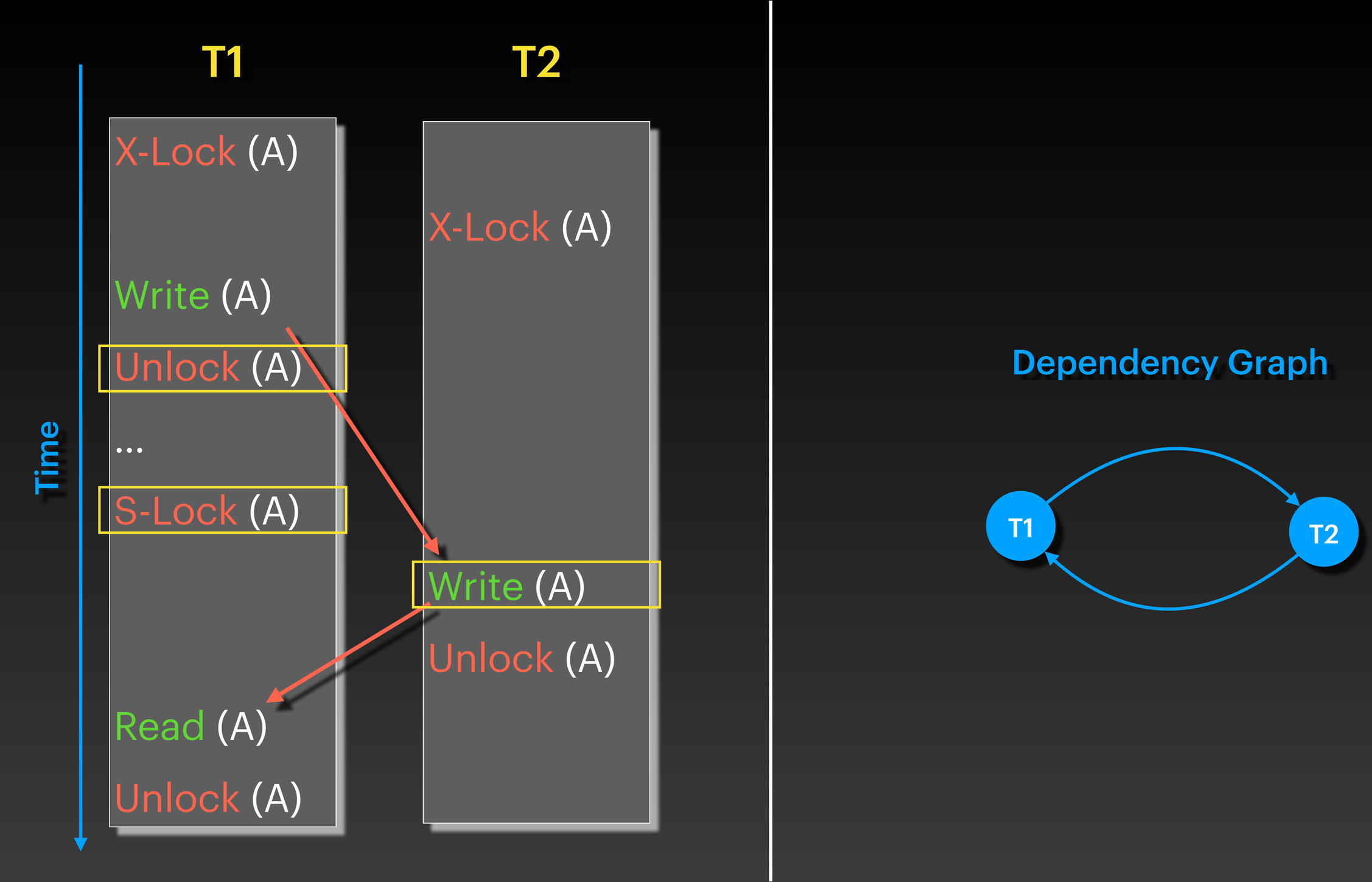




Dependency Graph



Not serializable!



Dependency Graph

T1

Time

X-Lock (A)

Write (A)

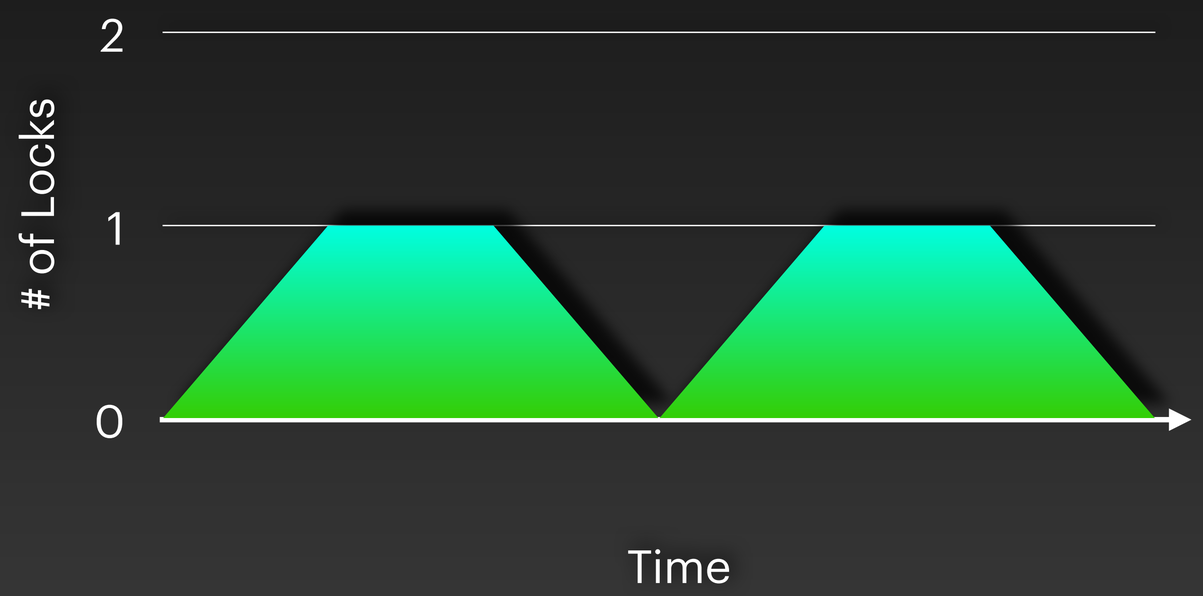
Unlock (A)

...

S-Lock (A)

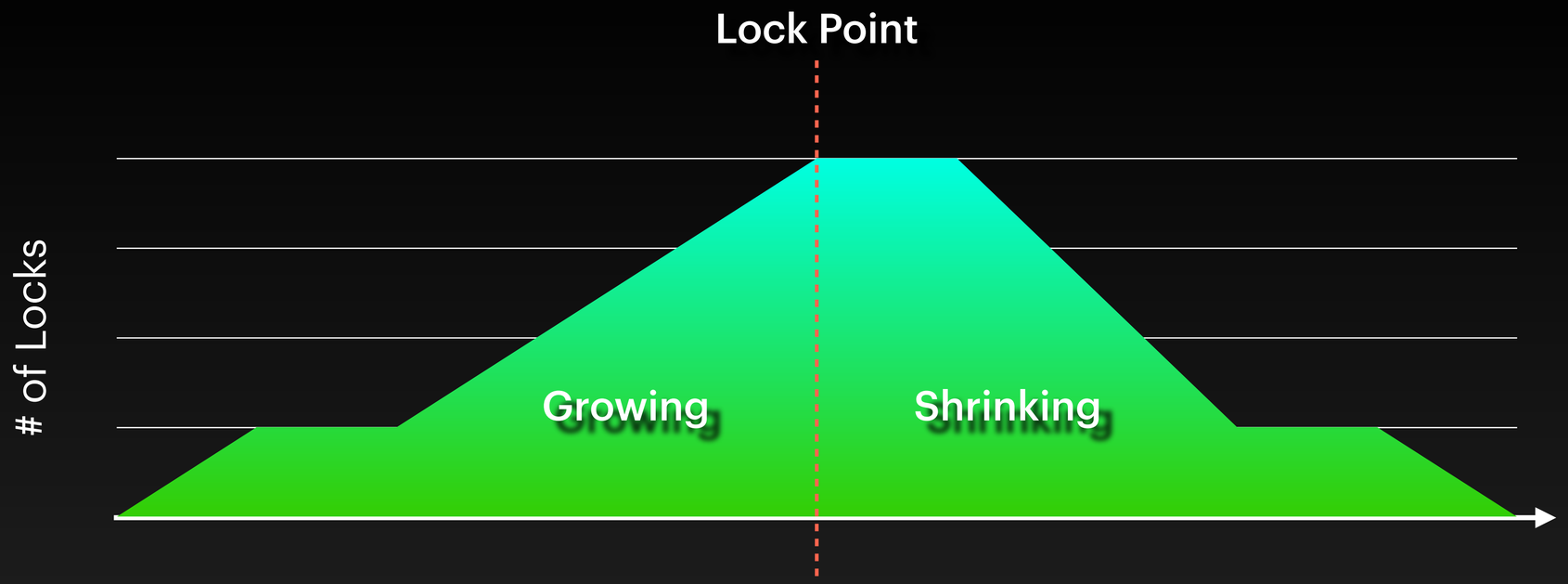
Read (A)

Unlock (A)



Two-Phase Locking (2PL)

- Phase 1 - Growing
  - Transaction can acquire/upgrade locks, without releasing/downgrading any locks
- Phase 2 - Shrinking
  - Transaction can release/downgrade locks. It cannot acquire/upgrade locks.



2PL

Not 2PL

X-Lock (A)

Write (A)

Unlock (A)

...

S-Lock (A)

Read (A)

Unlock (A)

Not 2PL

X-Lock (A)

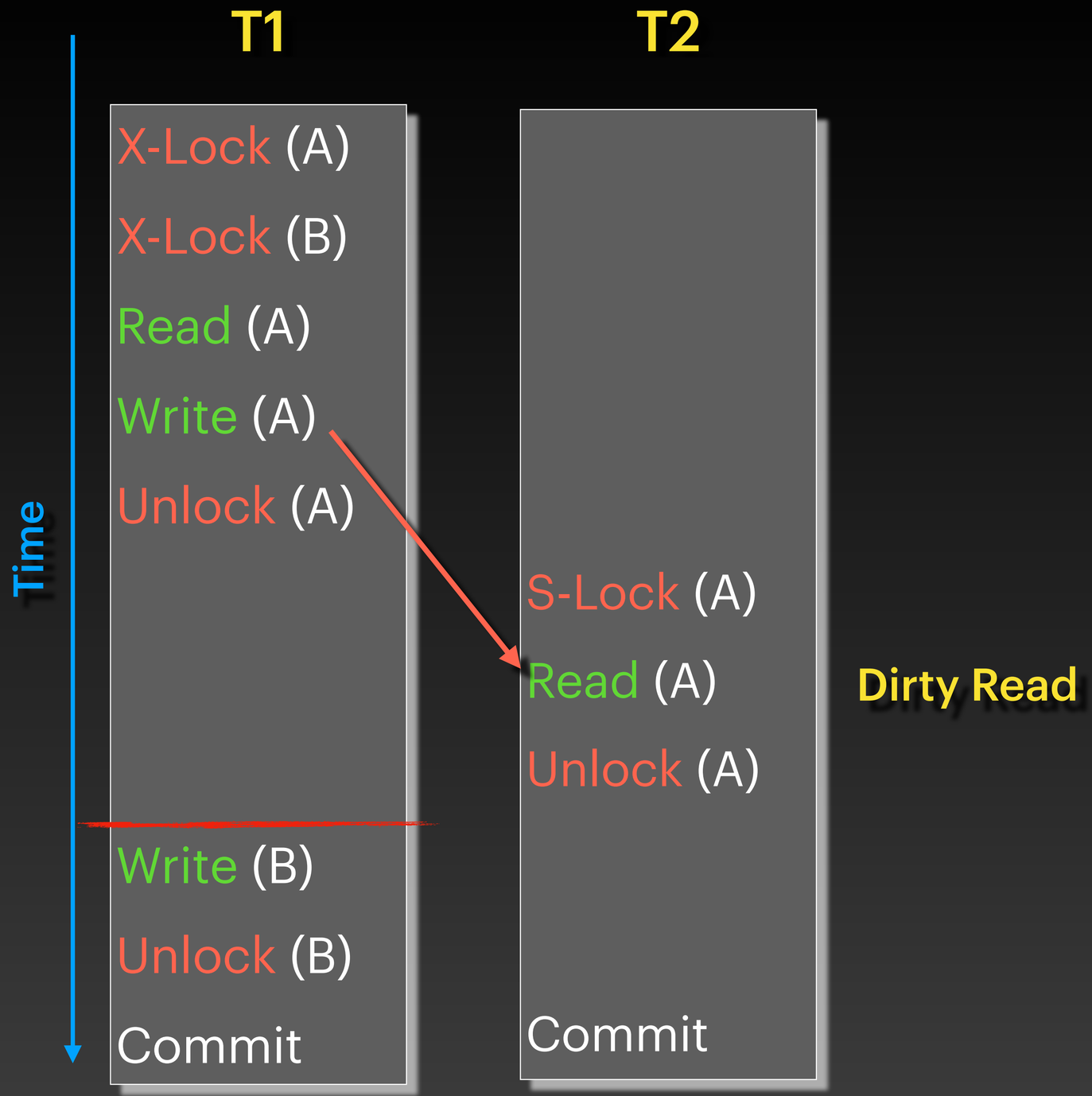
Write (A)

...

Read (A)

Unlock (A)

2PL



Dirty Read

- Valid 2PL
- Cascading Rollback

Strict 2PL

- Transaction can only release exclusive (write) locks after it has ended (committed or rolled back)
- Prevents dirty reads
- Prevents cascading rollbacks
    - Produces Strict schedules

Strong Strict (Rigorous) 2PL

- Transaction can only release ALL locks after it has ended (committed or rolled back)
- More restrictive than Strict 2PL
- Easier to implement than Strict 2PL (and more common)