BCSE302L - Database Systems

Module - 5

RECOVERABLE SCHEDULE

SCHEDULE

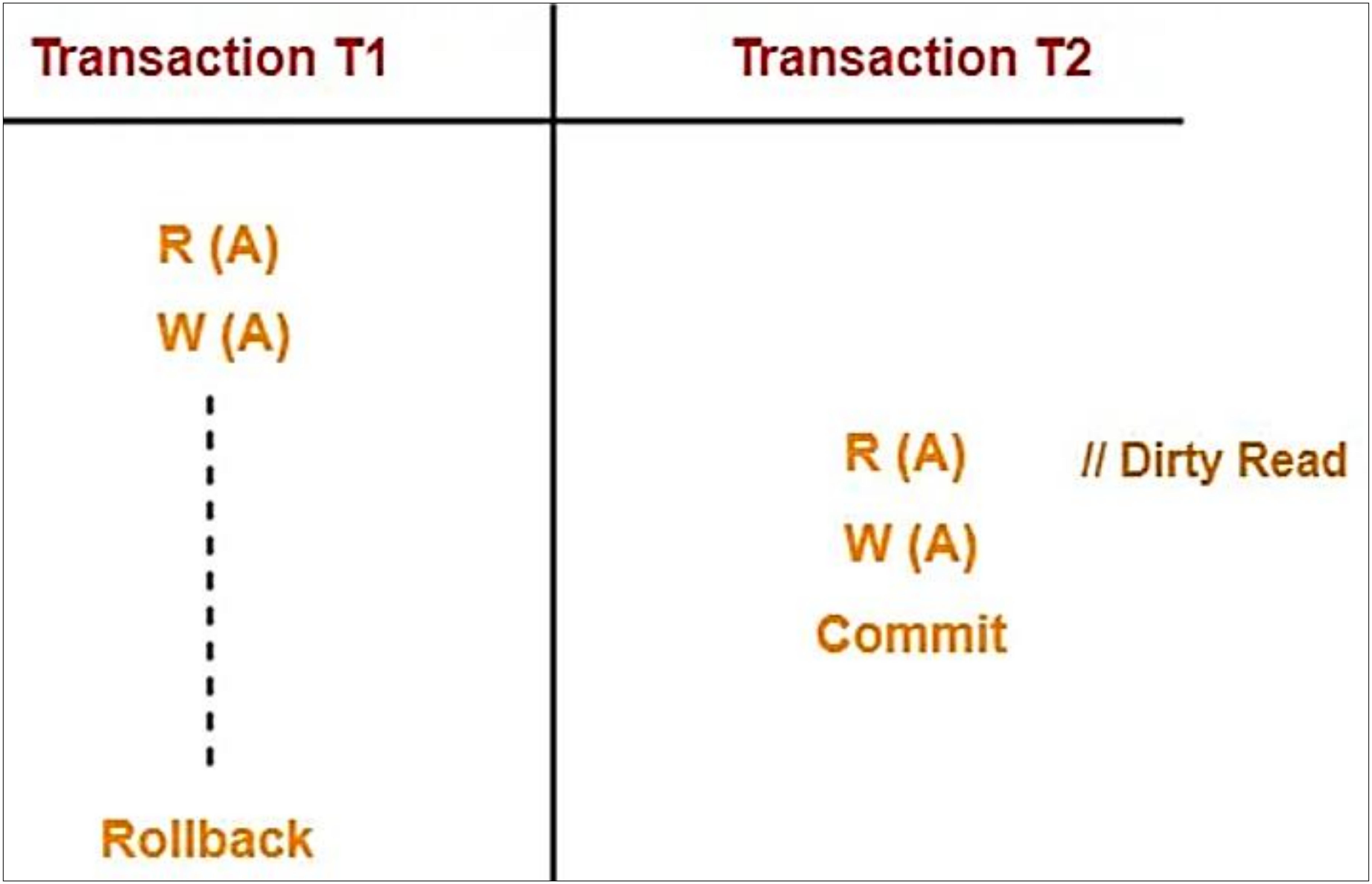
TYPES

- Schedules
  - Serial Schedule
  - Non serial Schedule
    - Serializable Schedule
      - Conflict Serializable Schedule
      - View Serializable Schedule
    - **Non-serializable**
      - **Recoverable Schedule**
        - **Cascading Schedule**
        - **Cascadeless Schedule**
        - **Strict Schedule**
      - **Non-recoverable**

NON-RECOVERABLE SCHEDULE

- If in a schedule,
  - A transaction performs a dirty read operation from an uncommitted transaction
  - And commits before the transaction from which it has read the value
  - then such a schedule is known as an **non-recoverable** or **Irrecoverable Schedule.**

NON-RECOVERABLE SCHEDULE



Transaction T1	Transaction T2
R (A)	
W (A)	
	R (A) // Dirty Read
	W (A)
	Commit
Rollback	

NON-RECOVERABLE SCHEDULE

- T2 performs a dirty read operation.
- T2 commits before T1.
- T1 fails later and roll backs.
- The value that T2 read now stands to be incorrect.
- T2 can not recover since it has already committed.

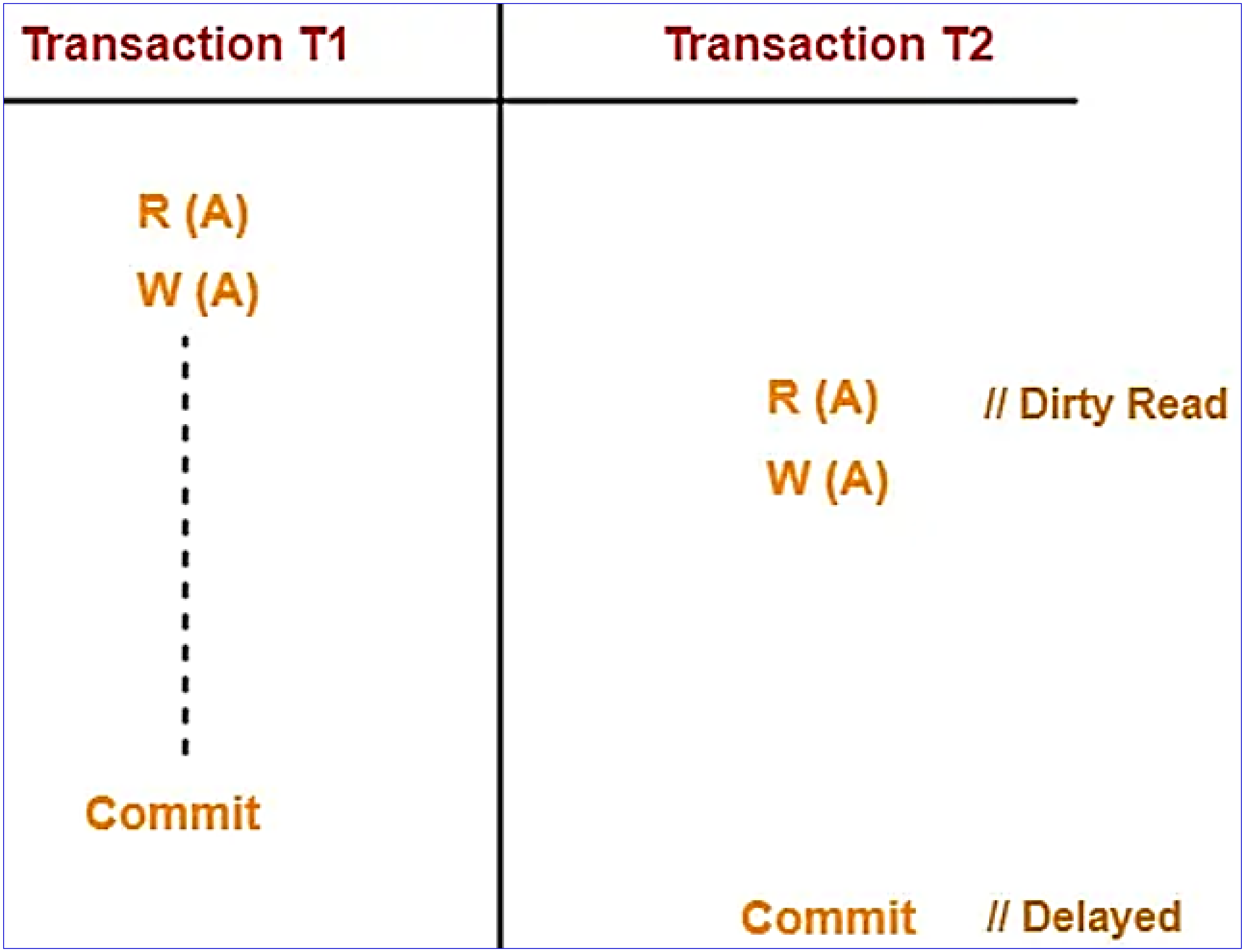
NON-RECOVERABLE SCHEDULE

TRANSACTION T1	TIME	TRANSACTION T2
---	--	
----	t1	UPDATE t
RETRIEVE t	t2	----
COMMIT	t3	---------
---	t4	ROLLBACK

RECOVERABLE SCHEDULE

- If in a schedule,
  - A transaction performs a dirty read operation from an uncommitted transaction
  - And its commit operation is delayed till the uncommitted transaction either commits or roll backs
  - then such a schedule is known as a **Recoverable Schedule**.
- The commit operation of the transaction that performs the dirty read is delayed.
- This ensures that it still has a chance to recover if the uncommitted transaction fails later.

RECOVERABLE SCHEDULE



Transaction T1	Transaction T2
R (A)	
W (A)	
	R (A) // Dirty Read
	W (A)
Commit	
	Commit // Delayed

RECOVERABLE SCHEDULE

- T2 performs a dirty read operation.
- The commit operation of T2 is delayed till T1 commits or roll backs.
- T1 commits later.
- T2 is now allowed to commit.
- In case, T1 would have failed, T2 has a chance to recover by rolling back.

Testing a schedule is recoverable or not?

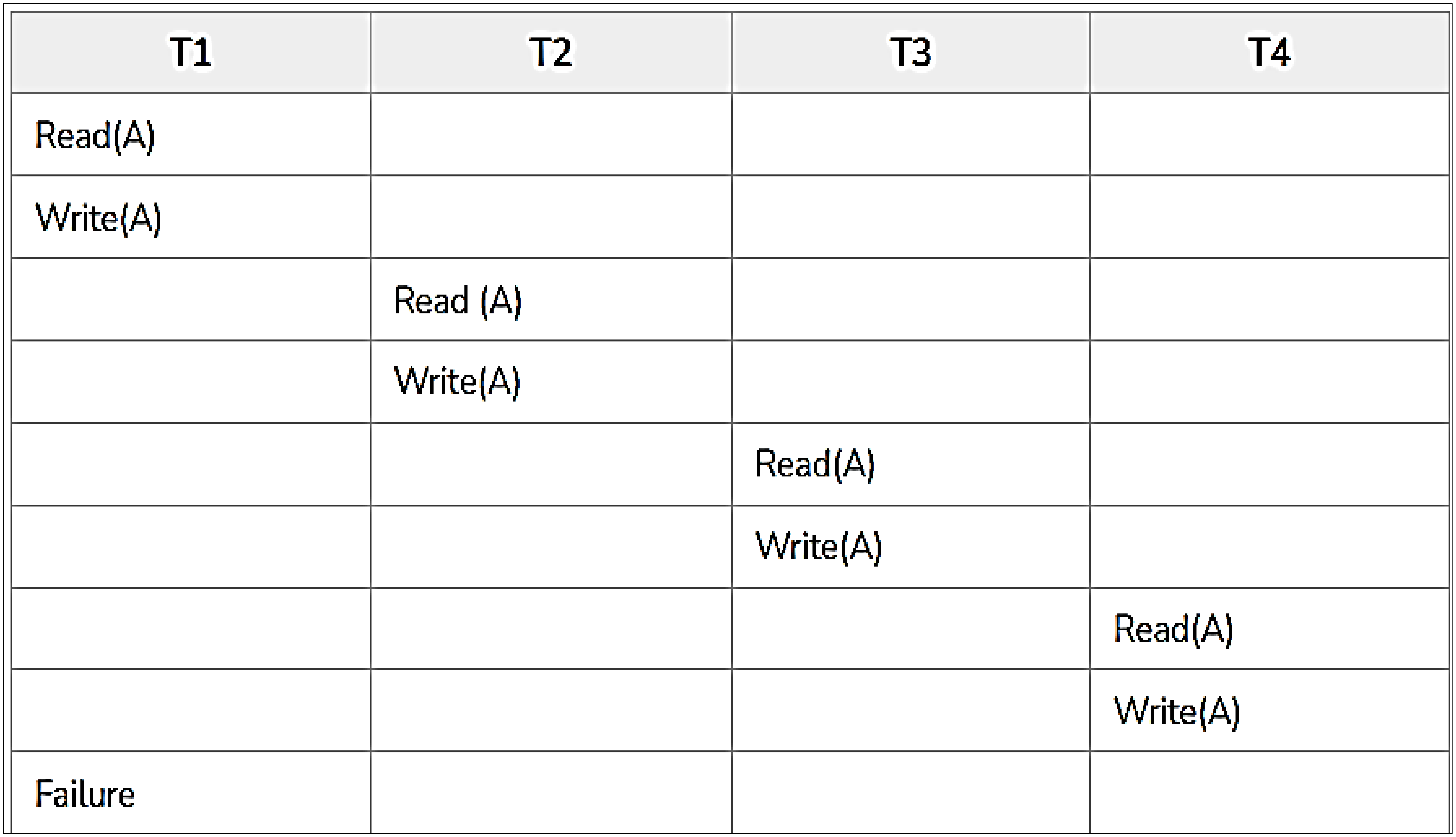
- **Rule-1**
  - All conflict serializable schedules are recoverable.
  - All recoverable schedules may or may not be conflict serializable.
- **Rule-2**
  - No dirty read means a recoverable schedule.

RECOVERABLE SCHEDULE

- **Cascading Schedule**
  - A cascading schedule is classified as a recoverable schedule.
  - A recoverable schedule is basically a schedule in which the commit operation of a particular transaction that performs read operation is delayed until the uncommitted transaction either commits or roll backs.

- **Cascading rollback** is a type of rollback in which if one transaction fails, then it will cause rollback of other dependent transactions.

RECOVERABLE SCHEDULE

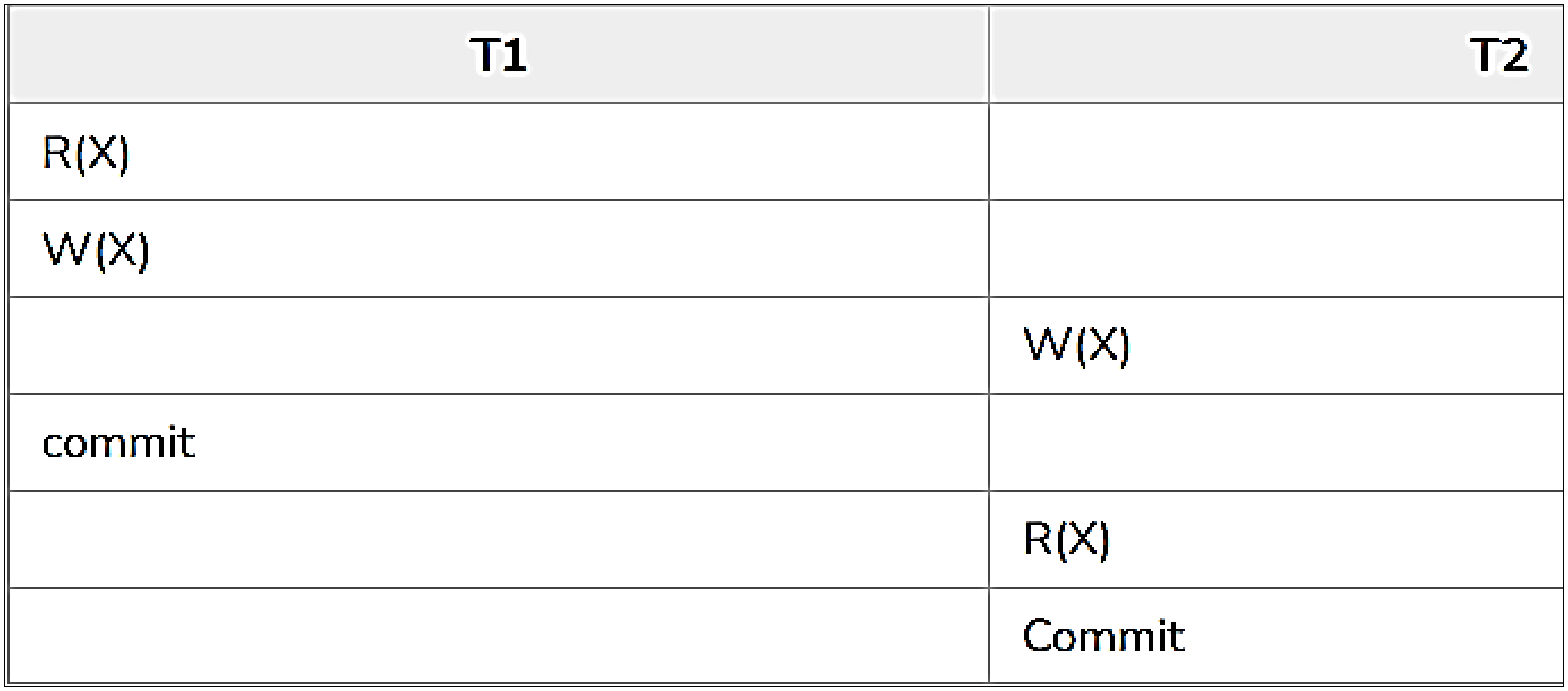


T1	T2	T3	T4
Read(A)			
Write(A)			
	Read (A)		
	Write(A)		
		Read(A)	
		Write(A)	
			Read(A)
			Write(A)
Failure			

Cascading rollback - because of T1 failure, T2 is rollback and rollback of T2 causes T3 to rollback and rollback T3 causes the T4 to rollback.

RECOVERABLE SCHEDULE

- **Cascadeless schedule**
  - When a transaction is not allowed to read data until the last transaction which has written it is committed or aborted, these types of schedules are called cascadeless schedules.



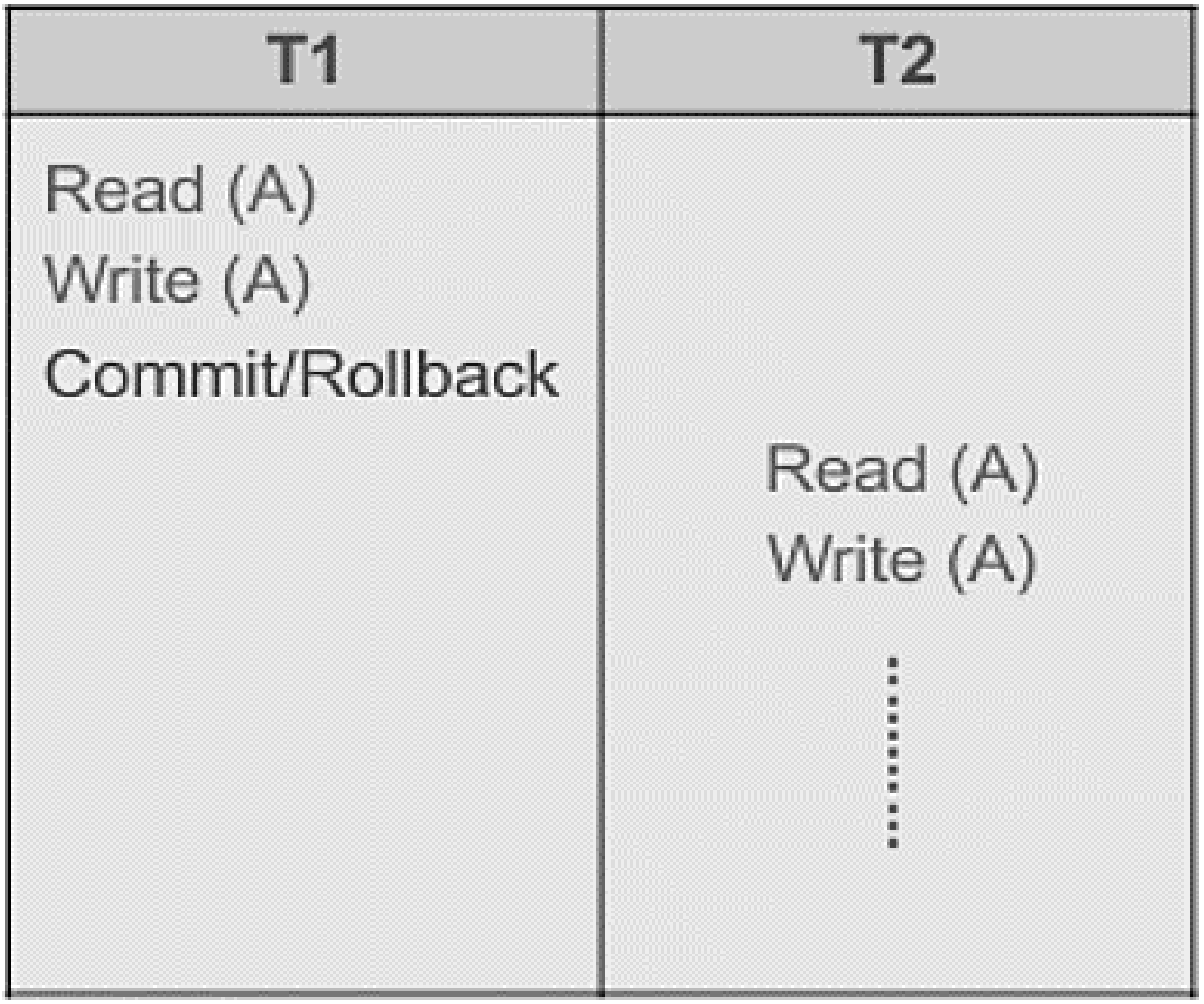
T1	T2
R(X)	
W(X)	
	W(X)
commit	
	R(X)
	Commit

The updated value of **X** is read by transaction T2 only after the commit of transaction T1. Hence, the schedule is cascadeless schedule

RECOVERABLE SCHEDULE

Strict schedule

- If the schedule contains no **read** or **write** before commit then it is known as strict schedule.
- Schedule with no read-write or write-write conflict is strict schedule

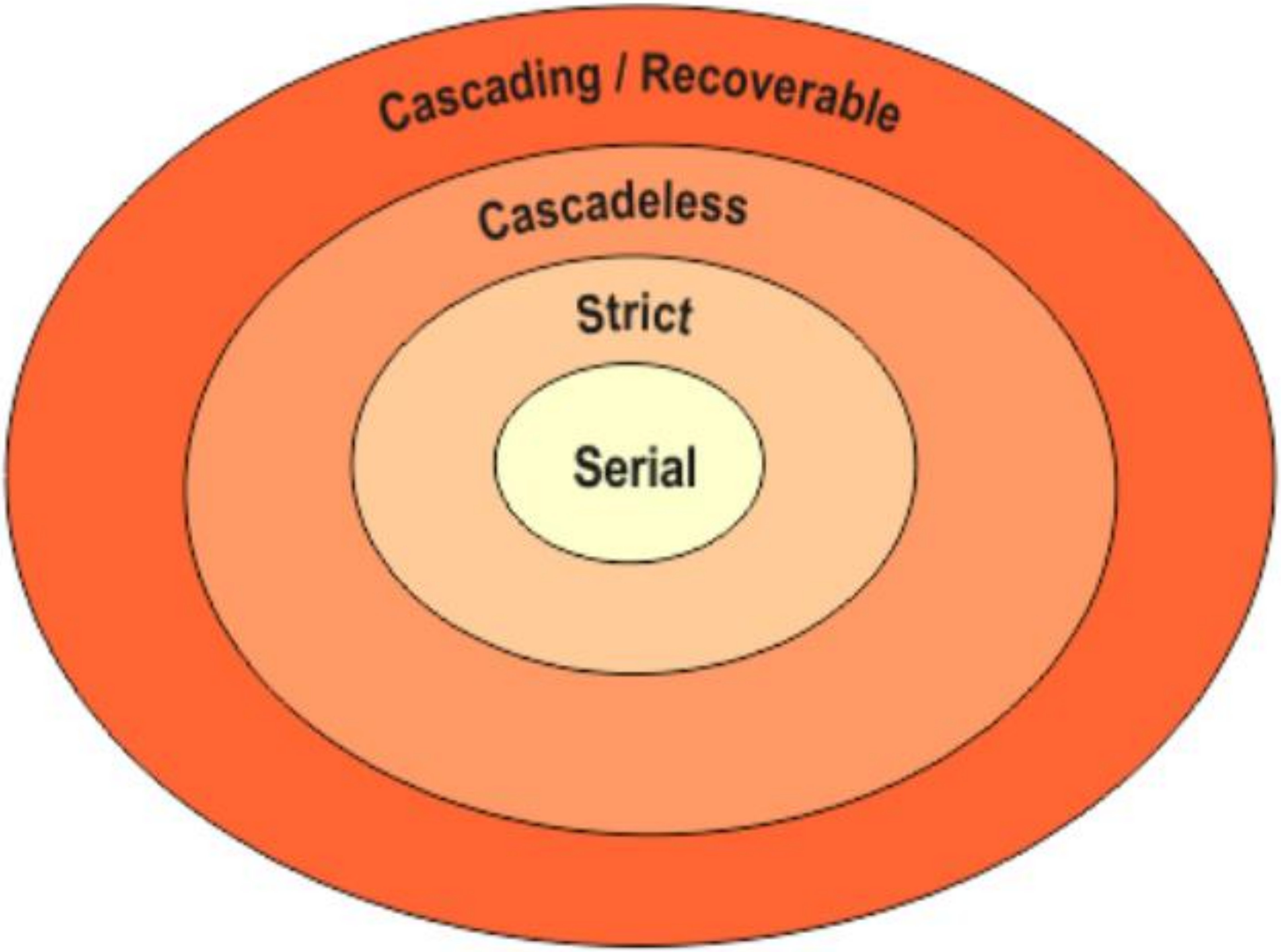


T1	T2
Read (A)	
Write (A)	
Commit/Rollback	
	Read (A)
	Write (A)
	⋮

Transaction T2 reads and writes the updated or written value of transaction T1 only after the transaction T1 commits. Hence, the schedule is strict schedule

Difference between cascading, cascadeless and strict schedule

Cascading Schedule	Casecadless Schedule	Strict Schedule
Cascading allows READ or WRITE operation for dependent Transaction (T2) before to T1 committed or abort.	Casecadless Don't allow READ to dependent Transaction (T2) untill T1 committed or abort. But it allows WRITE operation for dependent Transaction (T2) before to T1 committed.	Strict don't allow READ or WRITE operation to dependent Transaction (T2) until T1 committed or abort.
Note: Rollback of one Transaction leads to rollback of all other dependent Transactions Note: This case is problematic.	Note: This case is also problematic.	Note: This case if 100% safe.



All Serial schedules are Strict, Cascadeless, and recoverable but
All Recoverable Schedules may or may not be a Cascadeless, Strict and Serial