U O W

Introduction to Transaction Processing

(Serialization graph testing, two-phase locking, timestamp ordering)

CSIT882: Data Management Systems

UNIVERSITY
OF WOLLONGONG
AUSTRALIA

Outline

- Serialization graph testing protocol
- Two-phase locking protocol
- Timestamp ordering protocol

Serialization graph testing protocol (SGT)

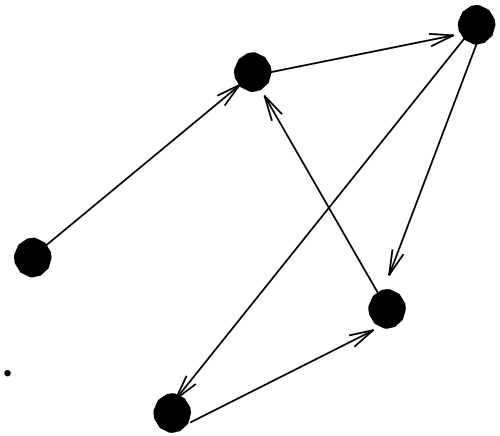
Principles

Scheduler maintains and tests serialization graph If an operation issued by a transaction violates conflict serializability (i.e. it creates a cycle in serialization graph) then such transaction is aborted

Preliminaries

A directed graph G = (V, A) consists of

- a vertex set V
- an edge set A such that each edge connects two vertices.



Cyclic Graph

Cyclic: G is cyclic if G contains a directed cycle.

Serialization graph

Consider some schedule of a set of transactions $T_1, T_2, ..., T_n$

Serialization graph — a directed graph $G = (V, E)$

- where the vertices (V) are the transactions.

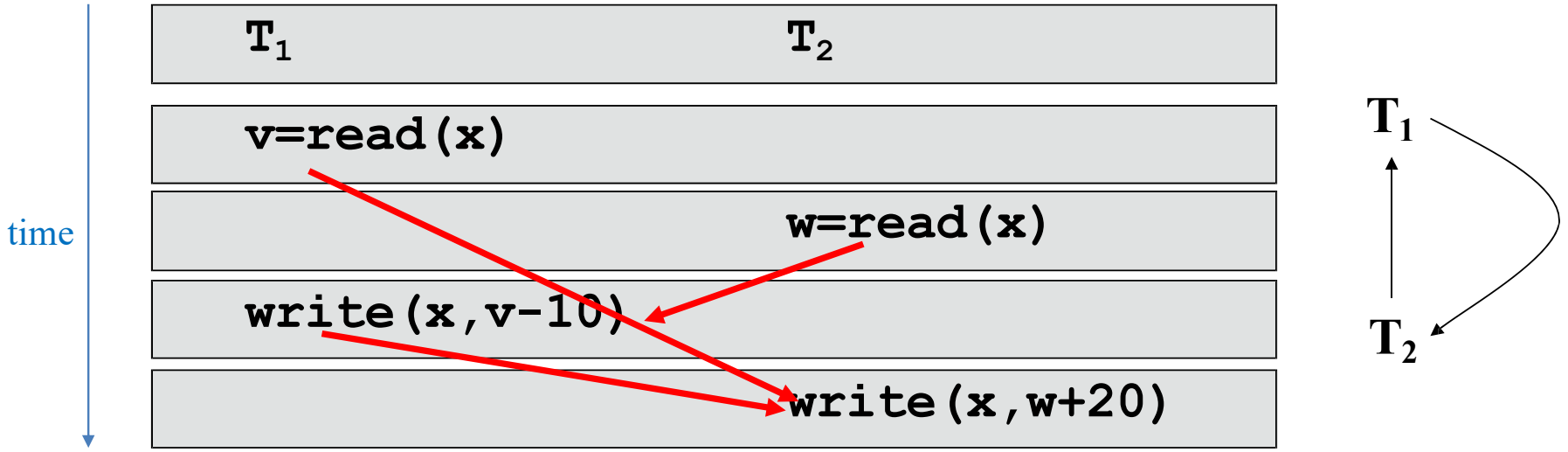
We draw an edge from T_i to T_j: $T_i \rightarrow T_j$

- if the two transactions are conflict,
- and T_i accessed the data item earlier.

A schedule is conflict serializable

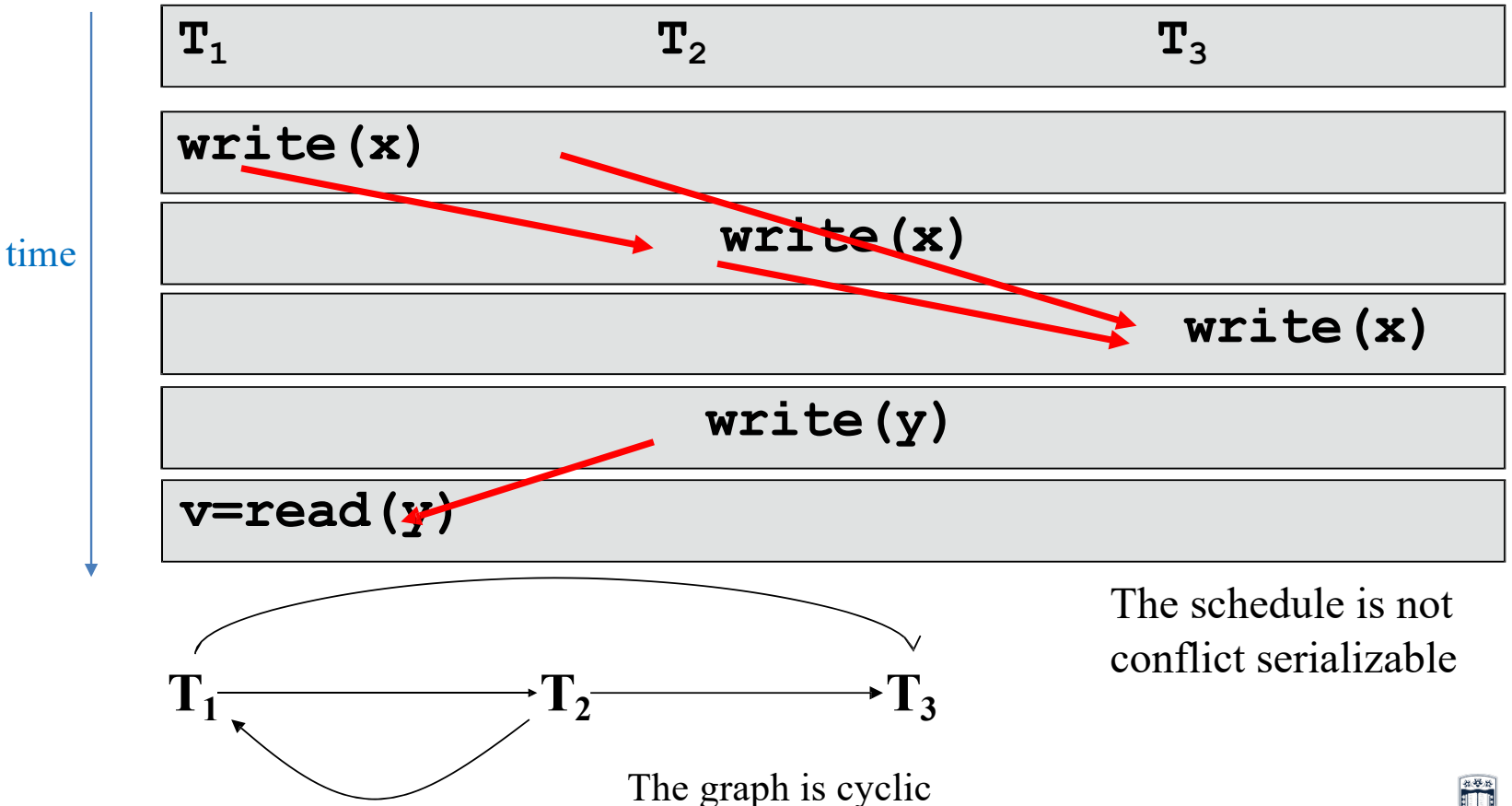
- if and only if its serialization graph is acyclic (cycle free).

Serialization graph



The serialization graph is cyclic (not cycle free)
Above schedule is not conflict serializable

Serialization graph



The schedule is not conflict serializable

Outline

- Serialization graph testing protocol
- Two-phase locking protocol
- Timestamp ordering protocol

Lock

A lock is a mechanism to control concurrent access to a data item

Data items can be locked by requesting locks

To use a data item, a transaction must acquire the relevant locks

A transaction may be granted a lock on an item if the requested lock is compatible with locks already held on the item by other transactions

A locking protocol is a set of rules followed by all transactions while requesting and releasing locks

Goal: Locking protocols enforce serializability by restricting the set of possible schedules

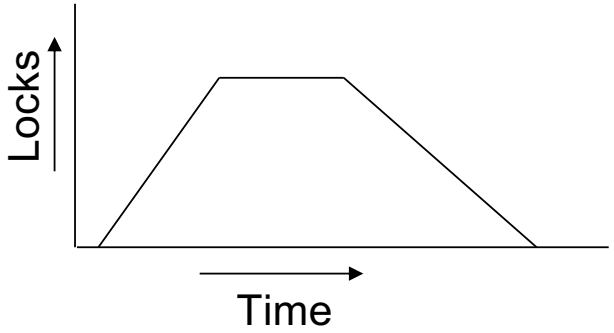
Two-phase locking (2PL) protocol

Principle

Each transaction must acquire all locks before releasing any lock

Phase 1: Growing Phase

- Transaction obtains locks
- Transaction does not release any locks



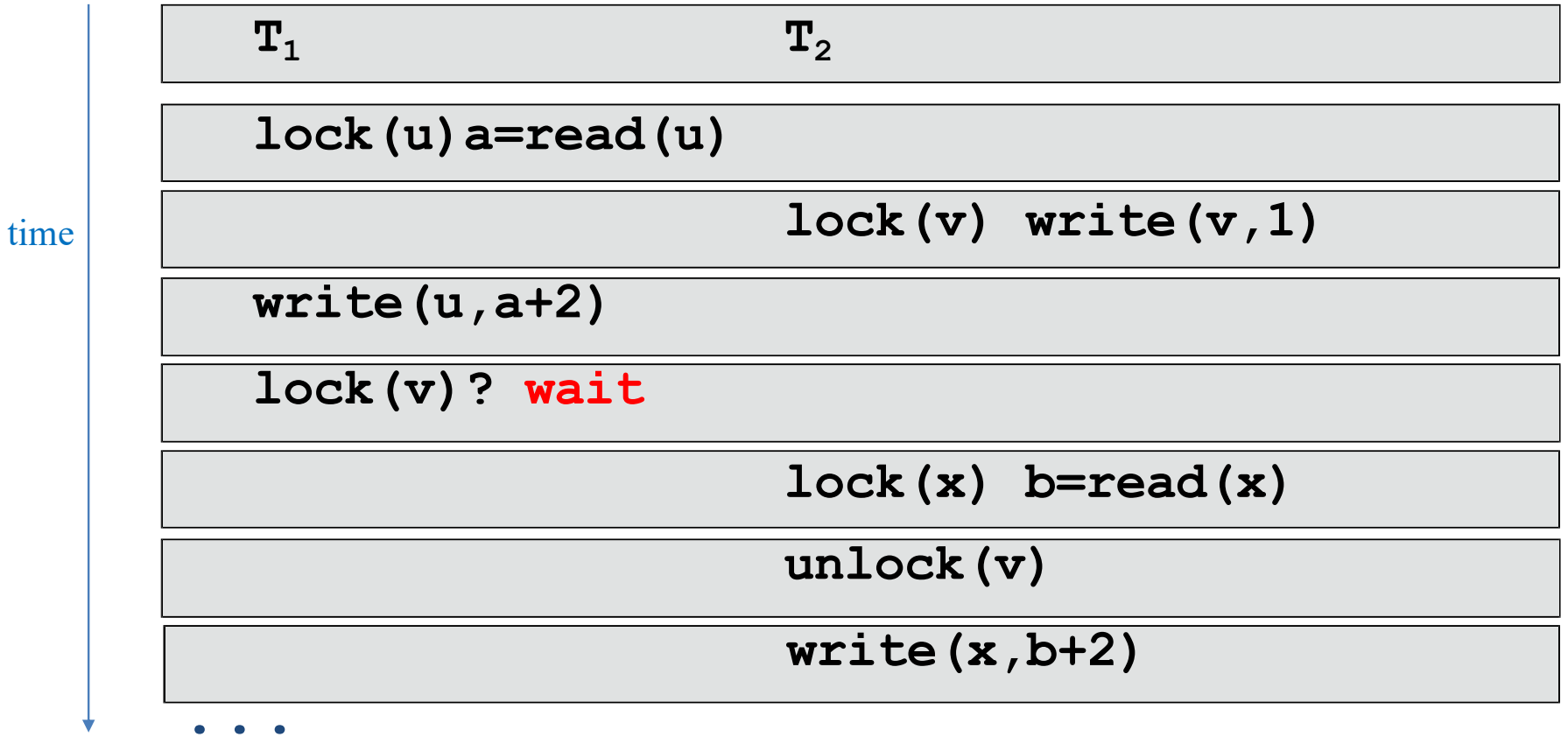
Phase 2: Shrinking Phase

- Transaction releases locks
- Transaction does not obtain new locks

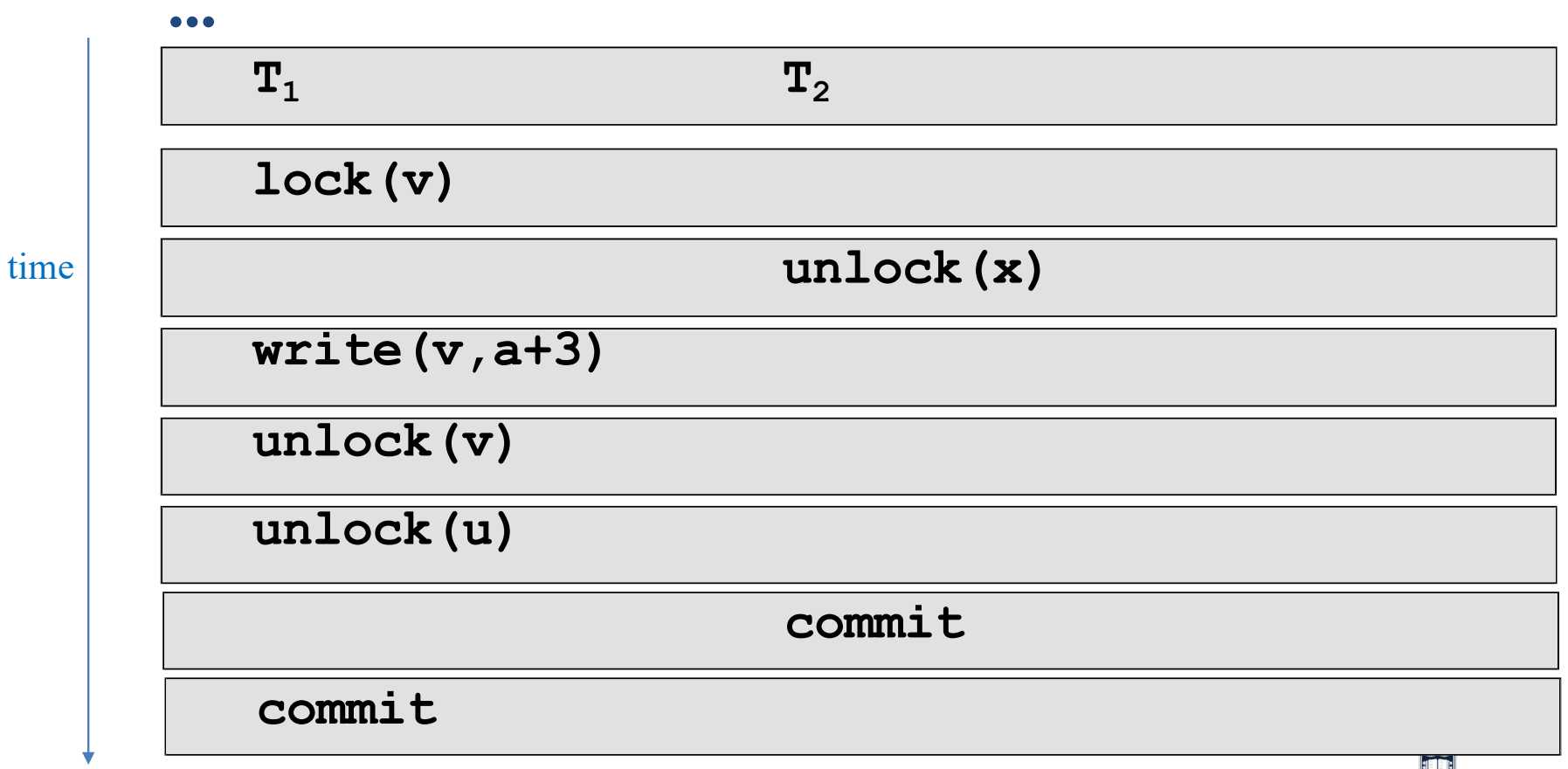
This protocol ensures serializability

- produces conflict-serializable schedules

Two-phase locking (2PL) protocol



Two-phase locking (2PL) protocol

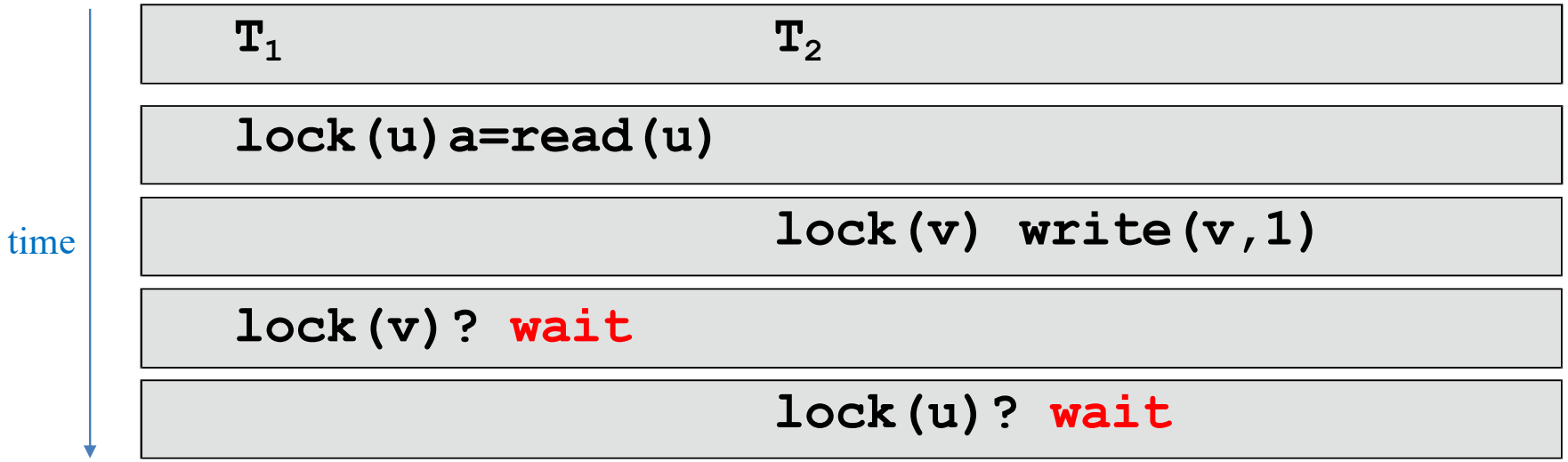


Deadlock

Deadlock occurs when each transaction T in a set of two or more transactions is waiting for some item that is locked by some other transaction T in the set.

In most locking protocols, a deadlock can exist.

Deadlock



Above situation is called a **deadlock**.

Issue: neither T1 nor T2 can make progress

How to handle deadlocks

There are several techniques for handling deadlocks

- **Timeouts**: If a transaction waits for a period longer than a system-defined timeout period, the system assumes that the transaction may be deadlocked and aborts it.

- **Deadlock detection**: Use transaction dependencies to construct a wait-for graph:
  create a vertex for each transaction; and
  an edge from T_i to T_j if T_i is waiting for an item locked by T_j.

  If the wait-for graph has a cycle, then a deadlock has occurred.

- **Deadlock prevention**: Use transaction timestamps to order transactions.

Outline

- Serialization graph testing protocol
- Two-phase locking protocol
- Timestamp ordering protocol

Timestamp ordering (TO) protocol

Principles

Each transaction obtains a timestamp at the start point

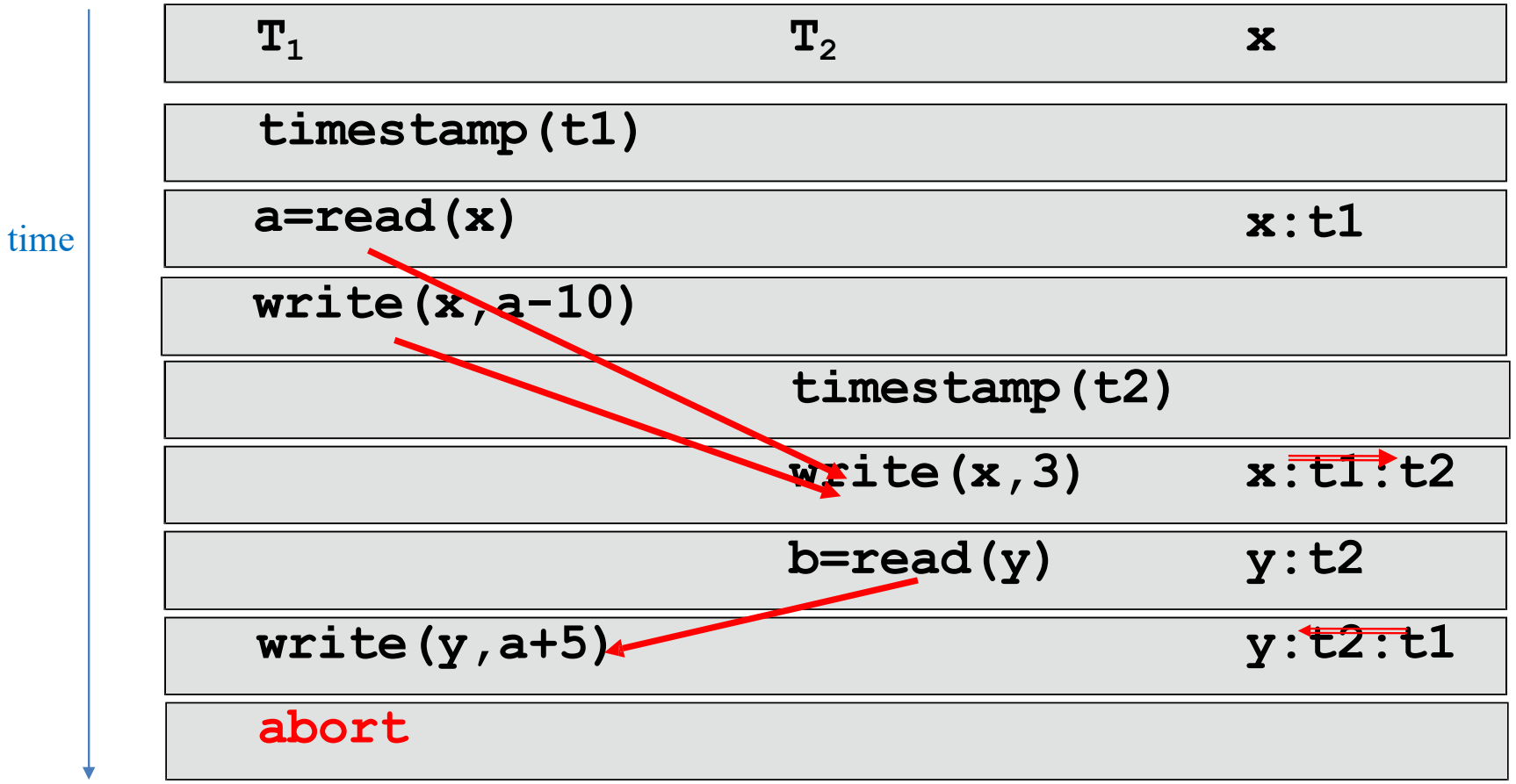
Data items are stamped each time a transaction accesses data items in a read or write mode

Access to data items is permitted in increasing order of timestamps

This protocol ensures serializability

- produces conflict-serializable schedules

Timestamp ordering (TO) protocol



References

Elmasri R., Navathe S., Fundamentals of Database Systems, 6th edition, chapter 21 Introduction to Transaction Processing Concepts and Theory, pp. 747-779

Elmasri R., Navathe S., Fundamentals of Database Systems, 6th edition, chapters 22.1, 22.3 Concurrency Control Techniques, pp. 780-794