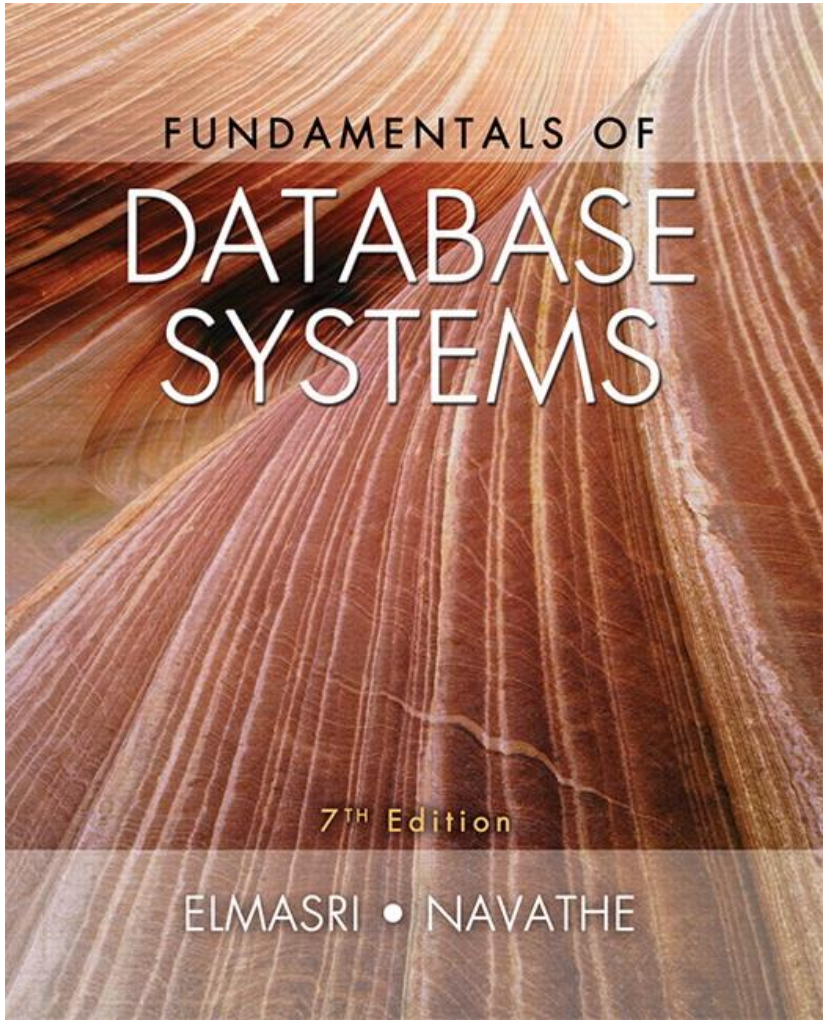

Copyright © 2016 Ramez Elmasri and Shamkant B. Navathe

CHAPTER 20

Introduction to Transaction Processing Concepts and Theory

Copyright © 2016 Ramez Elmasri and Shamkant B. Navathe

Introduction

- Transaction
  - Describes local unit of database processing
- Transaction processing systems
  - Systems with large databases and hundreds of concurrent users
  - Require high availability and fast response time

Copyright © 2016 Ramez Elmasri and Shamkant B. Navathe

20.1 Introduction to Transaction Processing

- Single-user DBMS
  - At most one user at a time can use the system
  - Example: home computer
- Multiuser DBMS
  - Many users can access the system (database) concurrently
  - Example: airline reservations system

Copyright © 2016 Ramez Elmasri and Shamkant B. Navathe

Introduction to Transaction Processing (cont'd.)

- Multiprogramming
  - Allows operating system to execute multiple processes concurrently
  - Executes commands from one process, then suspends that process and executes commands from another process, etc.

Copyright © 2016 Ramez Elmasri and Shamkant B. Navathe

Introduction to Transaction Processing (cont'd.)

- Interleaved processing
- Parallel processing
  - Processes C and D in figure below

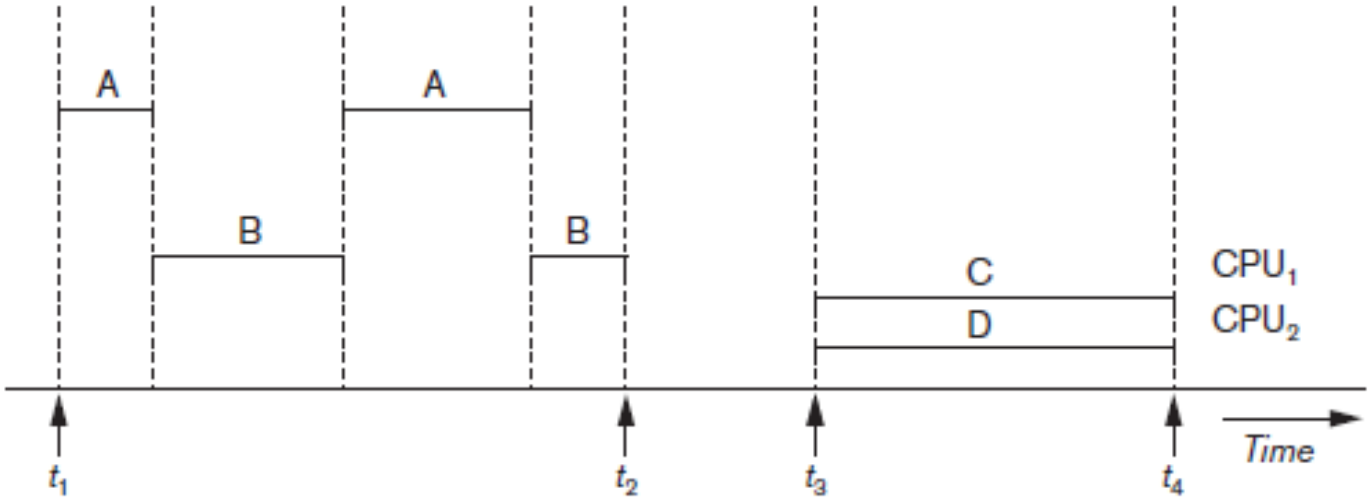

Figure 20.1 Interleaved processing versus parallel processing of concurrent transactions

Copyright © 2016 Ramez Elmasri and Shamkant B. Navathe

Transactions

- Transaction: an executing program
  - Forms logical unit of database processing
- Begin and end transaction statements
  - Specify transaction boundaries
- Read-only transaction
- Read-write transaction

Copyright © 2016 Ramez Elmasri and Shamkant B. Navathe

Database Items

- Database represented as collection of named data items
- Size of a data item called its granularity
- Data item
  - Record
  - Disk block
  - Attribute value of a record
- Transaction processing concepts independent of item granularity

Copyright © 2016 Ramez Elmasri and Shamkant B. Navathe

Read and Write Operations

- read_item(X)
  - Reads a database item named X into a program variable named X
  - Process includes finding the address of the disk block, and copying to and from a memory buffer
- write_item(X)
  - Writes the value of program variable X into the database item named X
  - Process includes finding the address of the disk block, copying to and from a memory buffer, and storing the updated disk block back to disk

Copyright © 2016 Ramez Elmasri and Shamkant B. Navathe

Read and Write Operations (cont'd.)

- Read set of a transaction
  - Set of all items read
- Write set of a transaction
  - Set of all items written

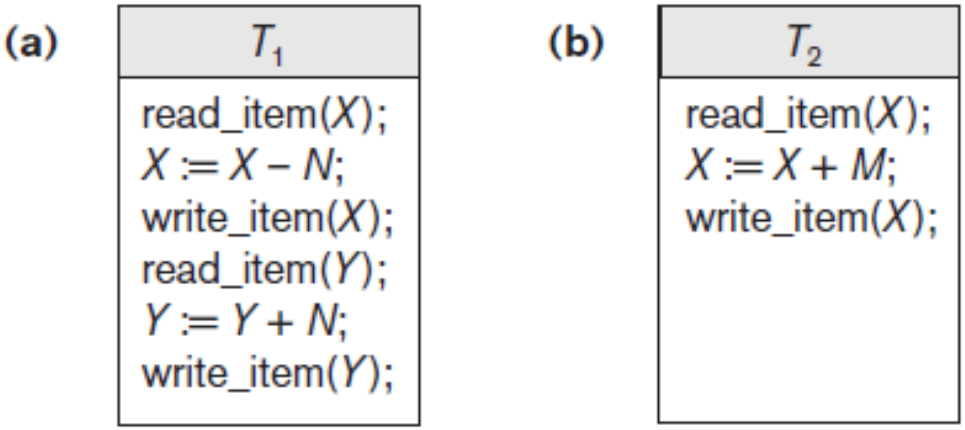

Figure 20.2 Two sample transactions (a) Transaction *T*1 (b) Transaction *T*2

Copyright © 2016 Ramez Elmasri and Shamkant B. Navathe

DBMS Buffers

- DBMS will maintain several main memory data buffers in the database cache
- When buffers are occupied, a buffer replacement policy is used to choose which buffer will be replaced
  - Example policy: least recently used

Copyright © 2016 Ramez Elmasri and Shamkant B. Navathe

Concurrency Control

- Transactions submitted by various users may execute concurrently
  - Access and update the same database items
  - Some form of concurrency control is needed
- The lost update problem
  - Occurs when two transactions that access the same database items have operations interleaved
  - Results in incorrect value of some database items

Copyright © 2016 Ramez Elmasri and Shamkant B. Navathe

The Lost Update Problem

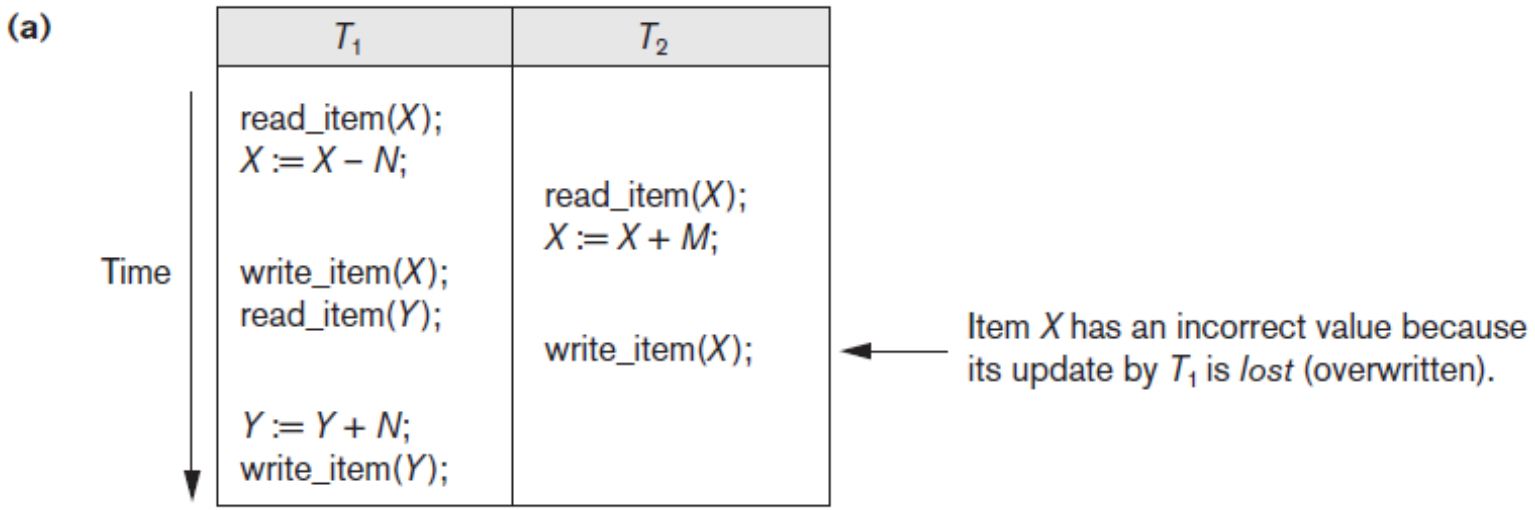

Figure 20.3 Some problems that occur when concurrent execution is uncontrolled (a) The lost update problem

Copyright © 2016 Ramez Elmasri and Shamkant B. Navathe

The Temporary Update Problem

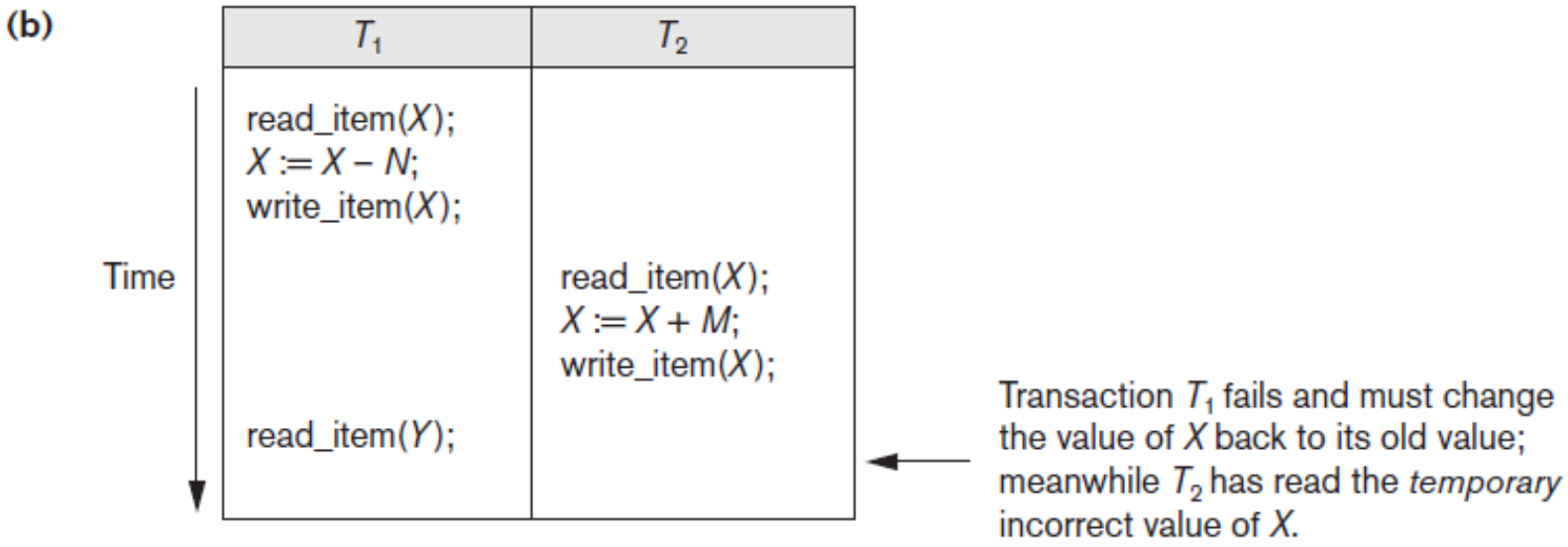

Figure 20.3 (cont'd.) Some problems that occur when concurrent execution is uncontrolled (b) The temporary update problem

Copyright © 2016 Ramez Elmasri and Shamkant B. Navathe

The Incorrect Summary Problem

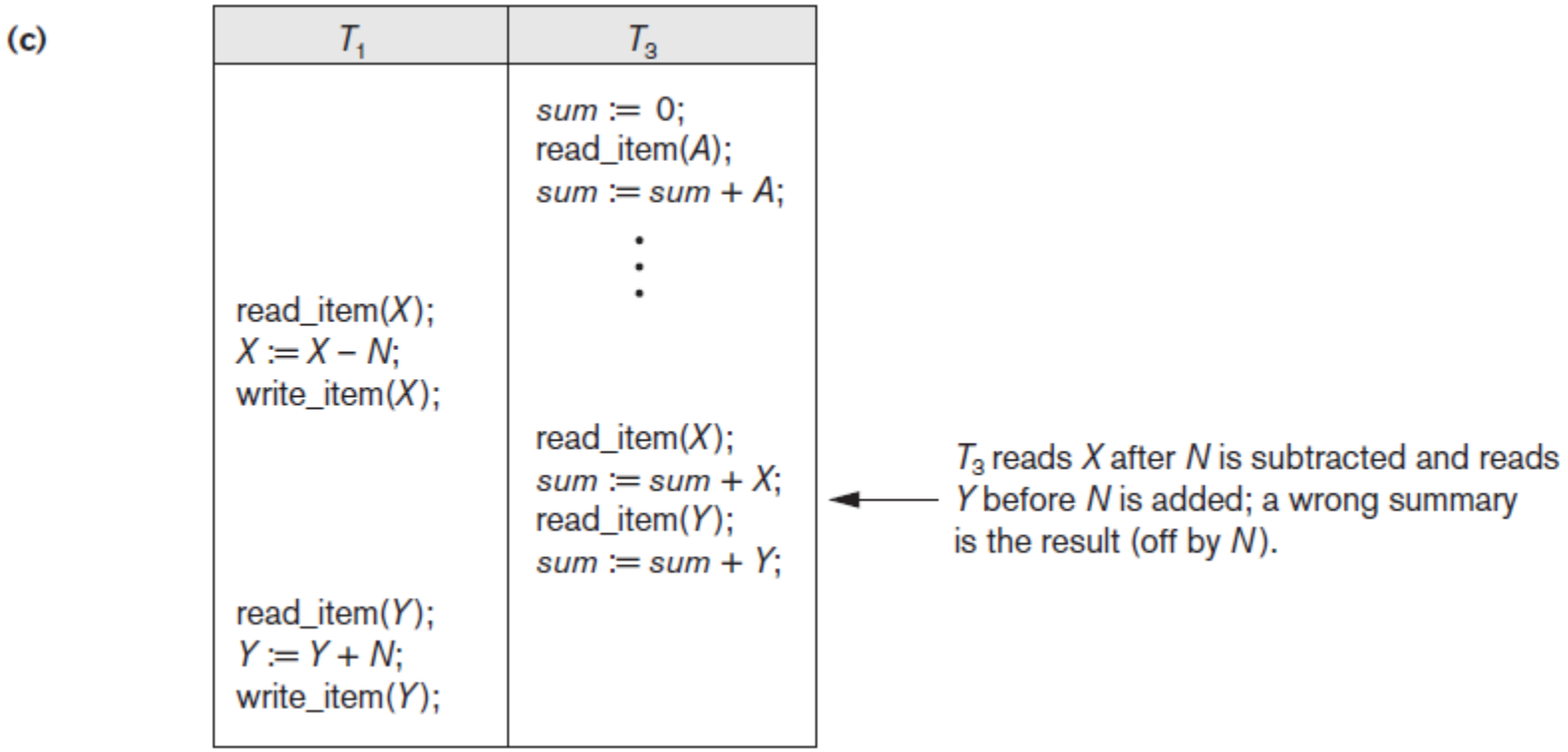

Figure 20.3 (cont'd.) Some problems that occur when concurrent execution is uncontrolled (c) The incorrect summary problem

Copyright © 2016 Ramez Elmasri and Shamkant B. Navathe

The Unrepeatable Read Problem

- Transaction T reads the same item twice
- Value is changed by another transaction T′ between the two reads
- T receives different values for the two reads of the same item

Copyright © 2016 Ramez Elmasri and Shamkant B. Navathe

Why Recovery is Needed

- Committed transaction
  - Effect recorded permanently in the database
- Aborted transaction
  - Does not affect the database
- Types of transaction failures
  - Computer failure (system crash)
  - Transaction or system error
  - Local errors or exception conditions detected by the transaction

Copyright © 2016 Ramez Elmasri and Shamkant B. Navathe

Why Recovery is Needed (cont'd.)

- Types of transaction failures (cont'd.)
  - Concurrency control enforcement
  - Disk failure
  - Physical problems or catastrophes
- System must keep sufficient information to recover quickly from the failure
  - Disk failure or other catastrophes have long recovery times

Copyright © 2016 Ramez Elmasri and Shamkant B. Navathe

20.2 Transaction and System Concepts

- System must keep track of when each transaction starts, terminates, commits, and/or aborts
  - BEGIN_TRANSACTION
  - READ or WRITE
  - END_TRANSACTION
  - COMMIT_TRANSACTION
  - ROLLBACK (or ABORT)

Copyright © 2016 Ramez Elmasri and Shamkant B. Navathe

Transaction and System Concepts (cont'd.)

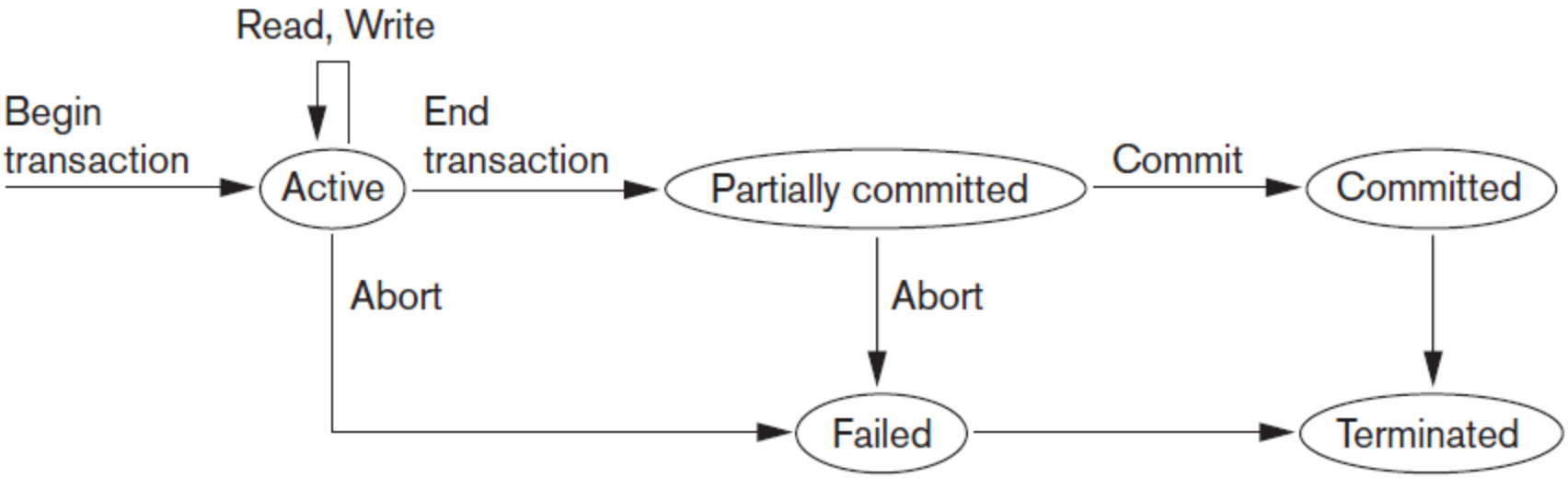

Figure 20.4 State transition diagram illustrating the states for transaction execution

Copyright © 2016 Ramez Elmasri and Shamkant B. Navathe

The System Log

- System log keeps track of transaction operations
- Sequential, append-only file
- Not affected by failure (except disk or catastrophic failure)
- Log buffer
  - Main memory buffer
  - When full, appended to end of log file on disk
- Log file is backed up periodically
- Undo and redo operations based on log possible

Copyright © 2016 Ramez Elmasri and Shamkant B. Navathe

Commit Point of a Transaction

- Occurs when all operations that access the database have completed successfully
  - And effect of operations recorded in the log
- Transaction writes a commit record into the log
  - If system failure occurs, can search for transactions with recorded start_transaction but no commit record
- Force-writing the log buffer to disk
  - Writing log buffer to disk before transaction reaches commit point

Copyright © 2016 Ramez Elmasri and Shamkant B. Navathe

20.3 Desirable Properties of Transactions

- ACID properties
  - Atomicity
    - Transaction performed in its entirety or not at all
  - Consistency preservation
    - Takes database from one consistent state to another
  - Isolation
    - Not interfered with by other transactions
  - Durability or permanency
    - Changes must persist in the database

Copyright © 2016 Ramez Elmasri and Shamkant B. Navathe