Database Management System – 40 Transaction Processing (Characterizing Schedules Based on Serializability)

Ajay James
Asst. Prof in CSE
Government Engineering College Thrissur

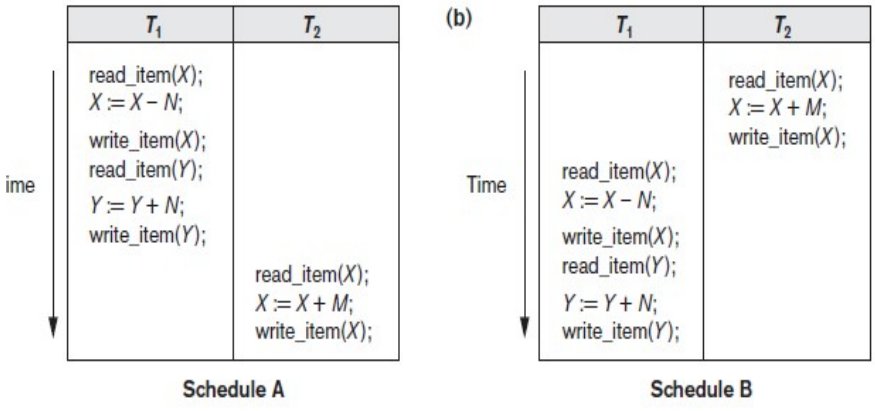
Outline

- Serializable schedules
- Serial and non serial schedules
- Result equivalent schedules
- Conflict equivalence

Characterizing Schedules Based on Serializability

- Serializable schedules
  - Always considered to be correct when concurrent transactions are executing
- Serial Schedules
  - Places simultaneous transactions in series
    - Transaction T1 before T2, or vice versa

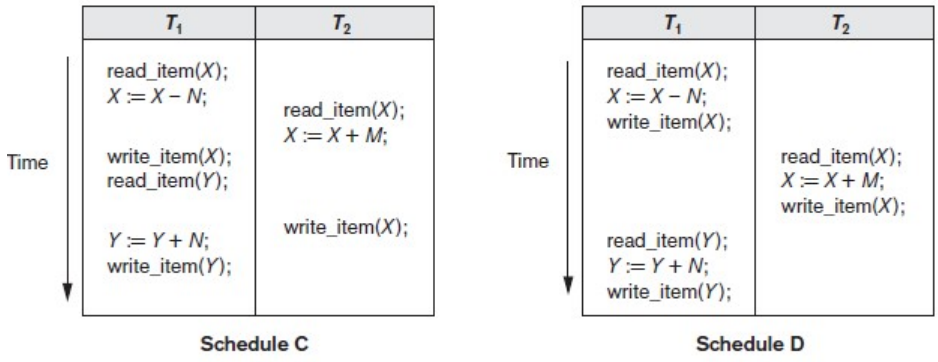
Serial Schedules



Schedule A

Schedule B

Non-serial schedules



Serial and Non Serial Schedule

- Serial
  - Schedule S is **serial** if, for every transaction T participating in the schedule, all the operations of T are executed consecutively in the schedule
- Otherwise **non serial**

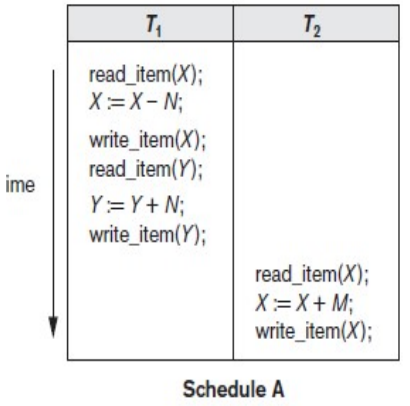
Problem with serial schedules

- Limit concurrency by prohibiting interleaving of operations
- Unacceptable in practice
- Solution: determine which schedules are equivalent to a serial schedule and allow those to occur

Serializable Schedule

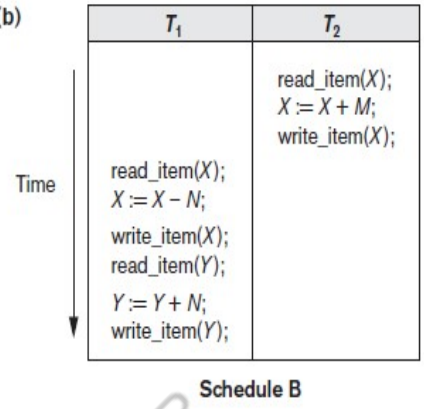
- A schedule S of n transactions is **serializable** if it is equivalent to some serial schedule of the same n transactions
- **n!** possible serial schedules of **n** transactions and many more possible nonserial schedules

Serial Schedules



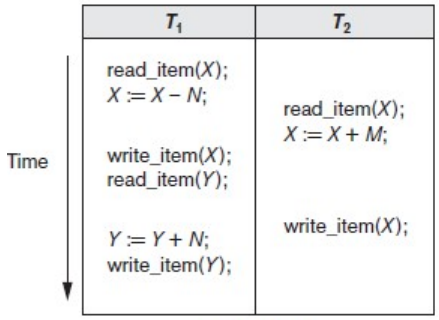
Schedule A

(b)

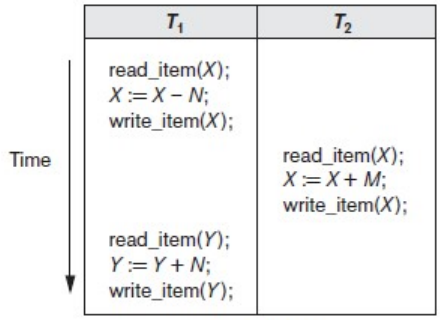


Schedule B

Non-serial schedules



Schedule C



Schedule D

Result equivalent schedules

- Produce the same final state of the database
- May be accidental
- Cannot be used alone to define equivalence of schedules

S_1

read_item(X);
$X := X + 10$;
write_item(X);

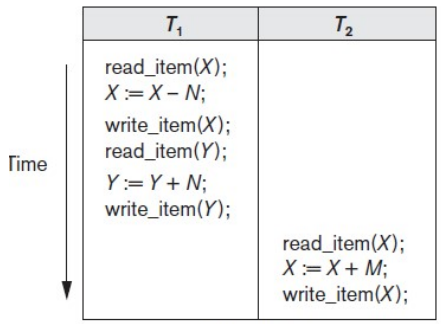
S_2
read_item(X); $X := X * 1.1$; write_item (X);

Conflict equivalence

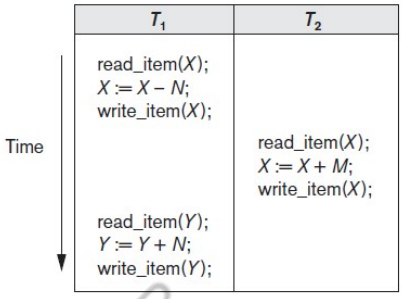
- Relative order of any two conflicting operations is the same in both schedules
- Two operations conflict
  - if they belong to different transactions
  - access the same database item
  - and either both are *write_item* operations or one is a *write_item* and the other a *read_item*

Serializable schedules

- Schedule S is serializable if it is conflict equivalent to some serial schedule S'



Schedule A



Schedule D

Reference

- Elmasri R. and S. Navathe, Database Systems: Models, Languages, Design and Application Programming, Pearson Education 6th edition and 7th edition

Thank you