***Conflict Serializability***

• Instructions Ii and Ij, of transactions Ti and Tj respectively, conflict if and only if there exists some item Q accessed by both Ii and Ij, and at least one of these instructions wrote Q.

• 1. Ii = read(Q), Ij = read(Q). Ii and Ij don’t conflict.

2. Ii = read(Q), Ij = write(Q). They conflict.

3. Ii = write(Q), Ij = read(Q). They conflict.

4. Ii = write(Q), Ij = write(Q). They conflict.

\* Intuitively, a conflict between Ii and Ij forces a (logical) temporal order between them. If Ii and Ij are consecutive in a schedule and they do not conflict, their results would remain the same even if they had been interchanged in the schedule

----------

• If a schedule S can be transformed into a schedule S0 by a series of swaps of non-conflicting instructions, we say that S and S0 are conflict equivalent.

• We say that a schedule S is conflict serializable if it is conflict equivalent to a serial schedule.

• Example of a schedule that is not conflict serializable :

T3 read(Q)

T4 write(Q)

T3 write(Q)

We are unable to swap instructions in the above schedule to obtain either the serial schedule < T3, T4 >, or the serial schedule < T4, T3 >

------------------------

Schedule 3 below can be transformed into Schedule 1, a serial schedule where T2 follows T1, by a series of swaps of non-conflicting instructions. Therefore Schedule 3 is conflict serializable.

T1 read(A)

T1 write(A)

T2 read(A)

T2 write(A)

T1 read(B)

T1 write(B)

T2read(B)

T2 write(B)

***View Serializability***

• Let S and S' be two schedules with the same set of transactions. S and S' are view equivalent if the following three conditions are met:

1. For each data item Q, if transaction Ti reads the initial value of Q in schedule S, then transaction Ti must, in schedule S', also read the initial value of Q.

2. For each data item Q, if transaction Ti executes read(Q) in schedule S, and that value was produced by transaction Tj (if any), then transaction Ti must in schedule S0 also read the value of Q that was produced by transaction Tj.

3. For each data item Q, the transaction (if any) that performs the final write(Q) operation in schedule S must perform the final write(Q) operation in schedule S'.

• As can be seen, view equivalence is also based purely on reads and writes alone.

-----

A schedule S is view serializable if it is view equivalent to a serial schedule.

• Every conflict serializable schedule is also view serializable.

• Schedule 9 (from text) — a schedule which is view-serializable but not conflict serializable.

T3 read(Q)

T4 write(Q)

T3 write(Q)

T6 write(Q)

• Every view serializable schedule which is not conflict serializable has blind writes.

***CONCURRENCY CONTROL***

Lock-Based Protocols

• Timestamp-Based Protocols

• Validation-Based Protocols

• Multiple Granularity

• Multiversion Schemes

• Deadlock Handling

• Insert and Delete Operations

• Concurrency in Index Structures