2PL ensures conflict-serializability

Overview of this video

We will show that 2PL schedules are conflict-serializability

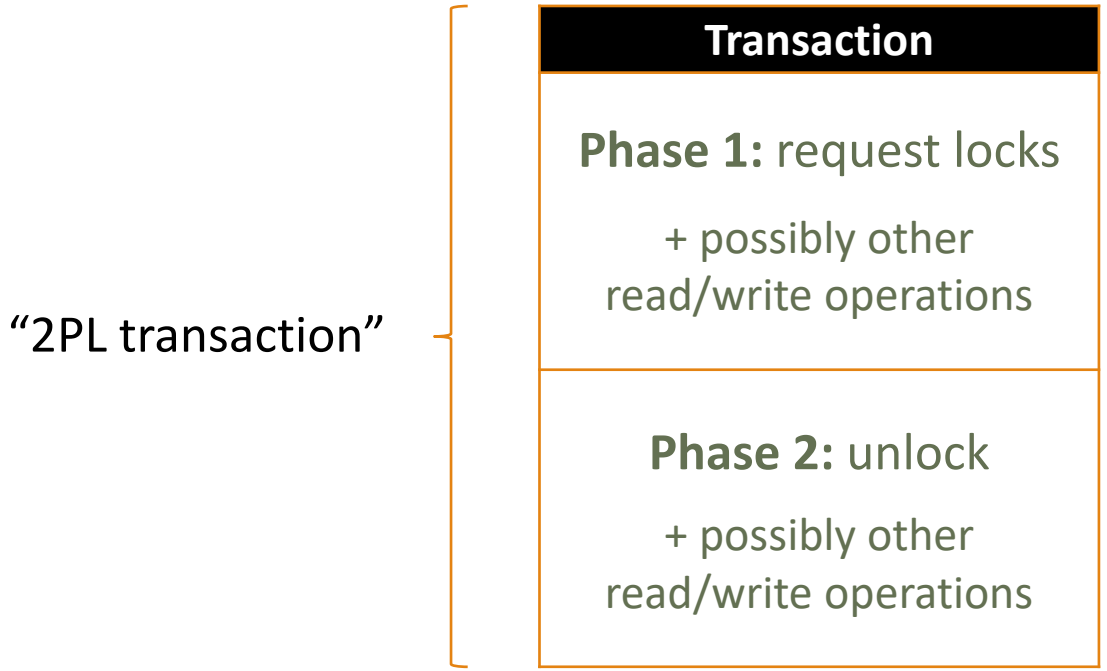
Two-Phase Locking (2PL)

(From video on Locking conflict-serializable schedules)

Simple modification of the simple locking mechanism that *guarantees conflict-serializability*

This video!

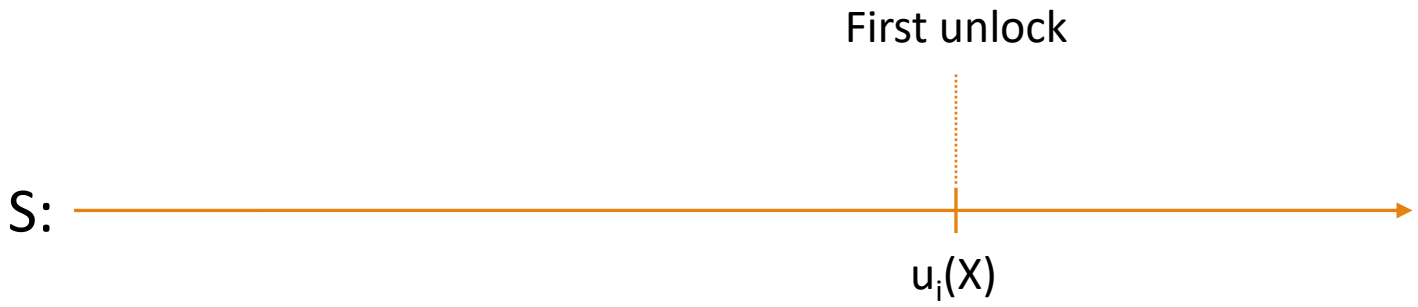
Two-phase locking (2PL) condition:
In each transaction, all lock operations precede all unlocks.



2PL Ensures Conflict-Serializability

If S is a schedule containing only 2PL transactions, then S is conflict-serializable.

Proof idea:

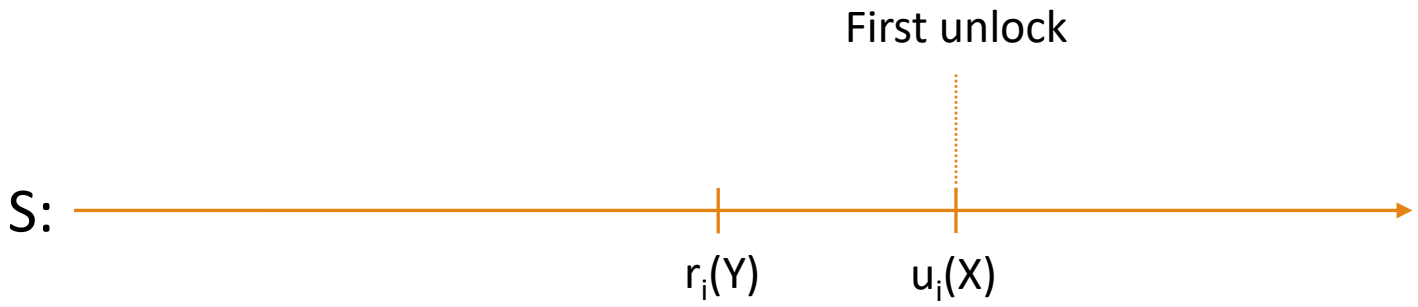


Claim: We can move all operations of T_i to beginning of schedule (using swaps of consecutive non-conflicting operations).

2PL Ensures Conflict-Serializability

If S is a schedule containing only 2PL transactions, then S is conflict-serializable.

Proof idea:

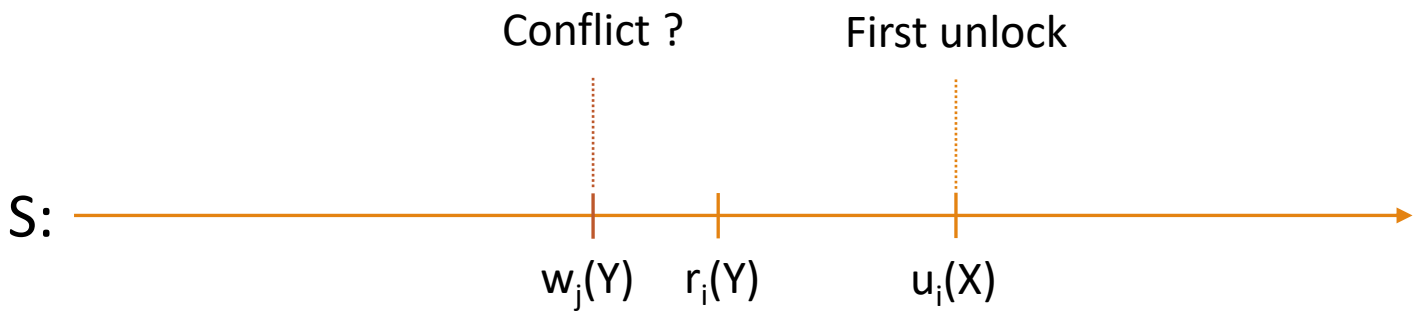


Claim: We can move all operations of T_i to beginning of schedule (using swaps of consecutive non-conflicting operations).

2PL Ensures Conflict-Serializability

If S is a schedule containing only 2PL transactions, then S is conflict-serializable.

Proof idea:

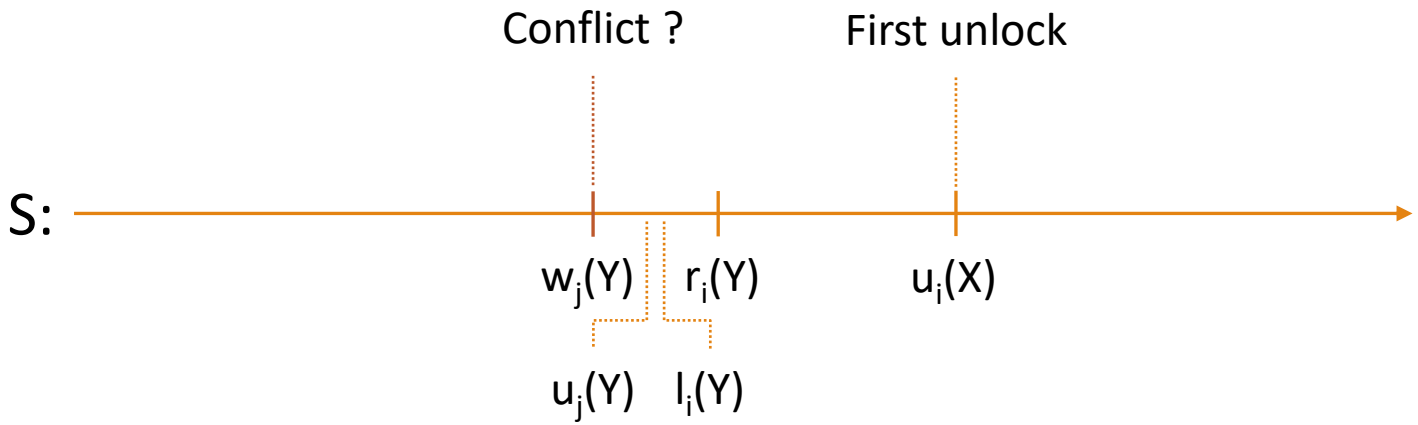


Claim: We can move all operations of T_i to beginning of schedule (using swaps of consecutive non-conflicting operations).

2PL Ensures Conflict-Serializability

If S is a schedule containing only 2PL transactions, then S is conflict-serializable.

Proof idea:

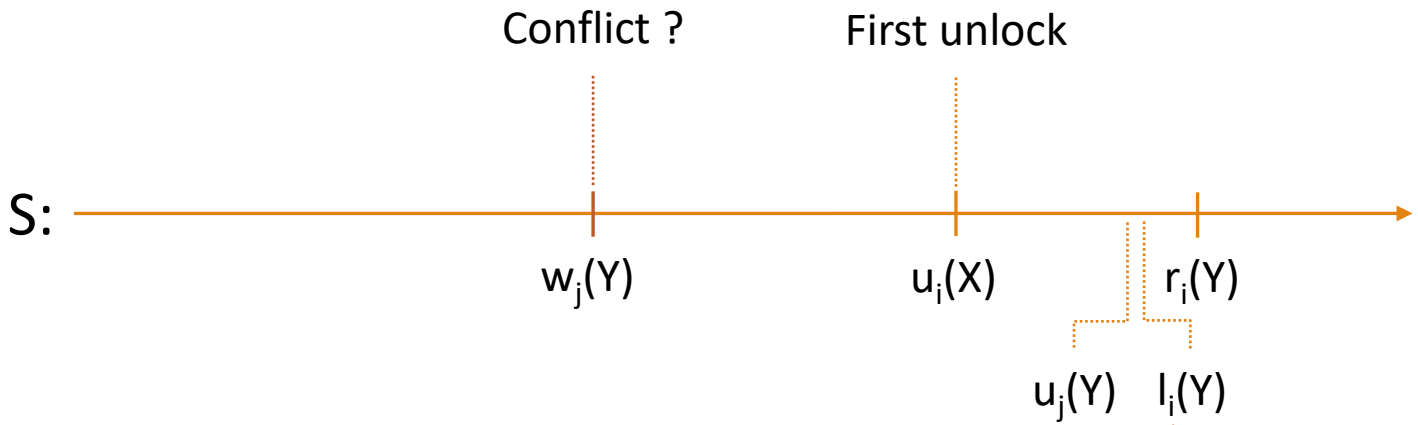


Claim: We can move all operations of T_i to beginning of schedule (using swaps of consecutive non-conflicting operations).

2PL Ensures Conflict-Serializability

If S is a schedule containing only 2PL transactions, then S is conflict-serializable.

Proof idea:

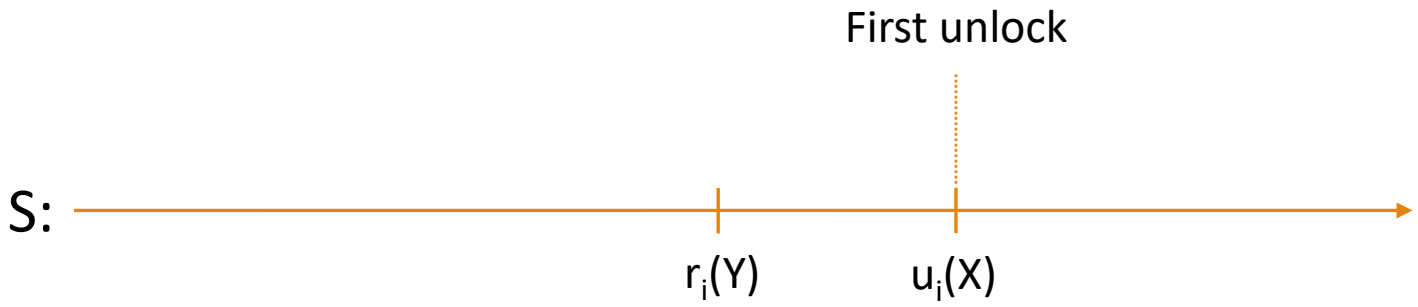


Claim: We can move all operations of T_i to beg e
(using swaps of consecutive non-conflicting operations).

2PL Ensures Conflict-Serializability

If S is a schedule containing only 2PL transactions, then S is conflict-serializable.

Proof idea:

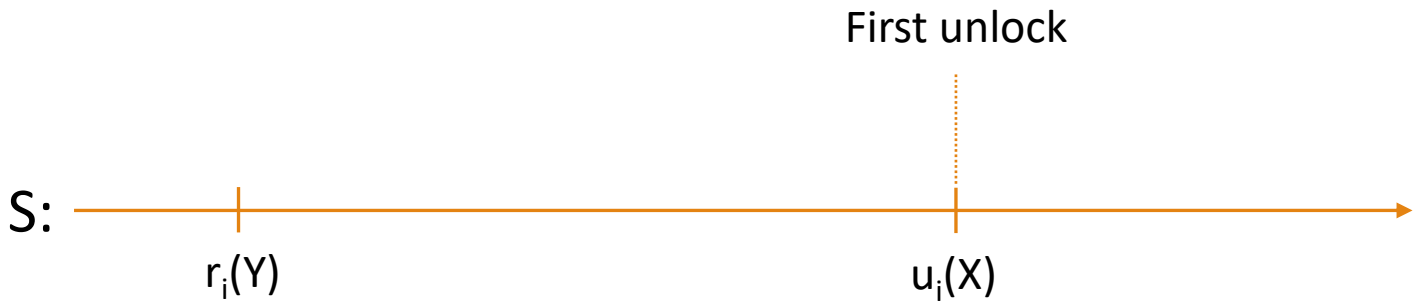


Claim: We can move all operations of T_i to beginning of schedule (using swaps of consecutive non-conflicting operations).

2PL Ensures Conflict-Serializability

If S is a schedule containing only 2PL transactions, then S is conflict-serializable.

Proof idea:



Claim: We can move all operations of T_i to beginning of schedule (using swaps of consecutive non-conflicting operations).

2PL Ensures Conflict-Serializability

If S is a schedule containing only 2PL transactions, then S is conflict-serializable.

Proof idea:

The tail is a new schedule that contains only 2PL transactions.

Claim: We can move all operations of T_i to beginning of schedule (using swaps of consecutive non-conflicting operations).

Summary

2PL schedules are conflict-serializable as well and are a simple way to ensure such