

15.10

a

- If transactions T_i and T_j have an I mode lock on item X , which increments X by C_i and C_j respectively, the order of the 2 increment operations has no impact, since the result is always $X + C_i + C_j$.
- If transaction T_i has an I mode lock and T_j has an S mode lock on item X , they are not compatible since the value T_j obtains may be the value before incrementation or the value after incrementation. It's the same for an I mode lock and an X mode lock.
- By contradiction:
Assume the precedence graph produced by 2PL of a series of conflicting transactions is $T_0 \rightarrow T_1 \rightarrow \dots \rightarrow T_i \rightarrow T_0$, which is cyclic and the schedule is non-serializable. Let α_j denotes the lock point of transaction T_j , then we have $\alpha_0 < \alpha_1 < \dots < \alpha_i < \alpha_0$, which is impossible.
Conclusion: two-phase locking ensures serializability

b

The I mode lock increases concurrency because incrementation is a simple operation and an I mode lock is compatible with itself, which allows multiple transactions running incrementations at the same time. If there is no I mode lock, a transaction has to implement an incrementation by an X mode lock, which is not compatible with itself and reduces the concurrency.

15.21

- Ensures recoverability and avoids cascading roll-backs
- Relatively simple code
- Considerable concurrency