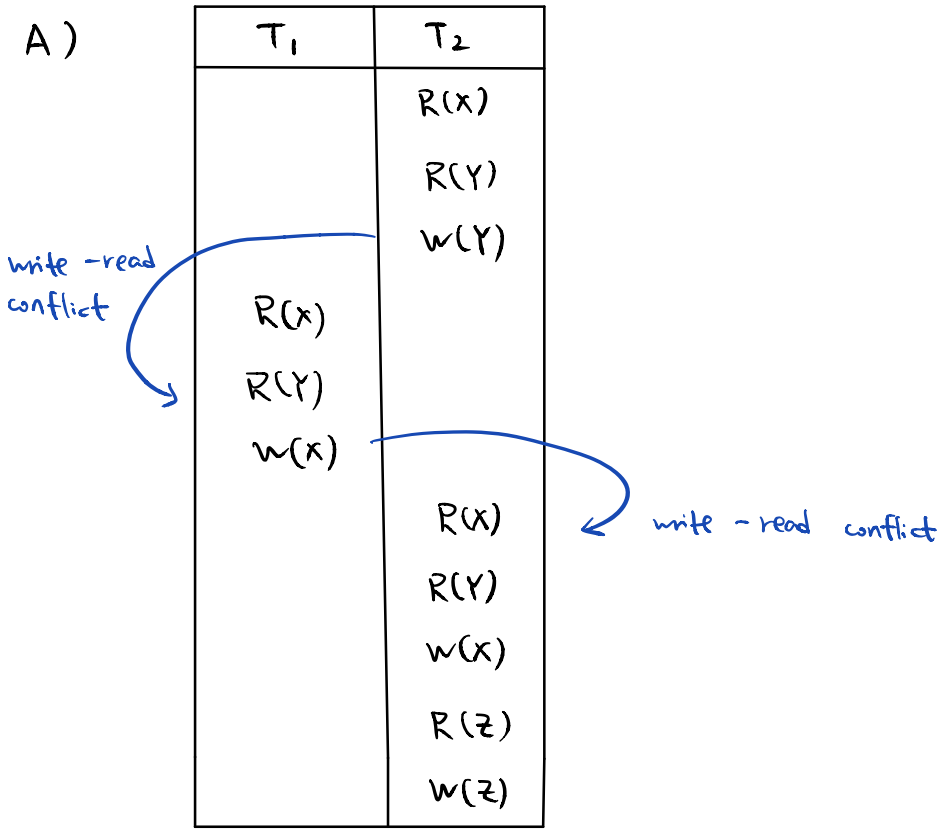
Schedules and anomalies

T1: R(X), R(Y), W(X)

T2: R(X), R(Y), W(Y), R(X), R(Y), W(X), R(Z), W(Z)

A)



This is not conflict serializable since we have a cycle between T1 and T2 with 2 write - read conflicts on object Y and X

B) We need a transaction for a single query like R(X), consider T2 has a W(X) before the single query R(X), and after this R(X) being executed, T2 abort the transaction. In this case, R(X) reads a wrong value, which is caused by a dirty read. Thus, we need a transaction for this single query to avoid conflicts, and need a lock to block this R(X) and unlock it after the transaction T2 being committed or aborted.