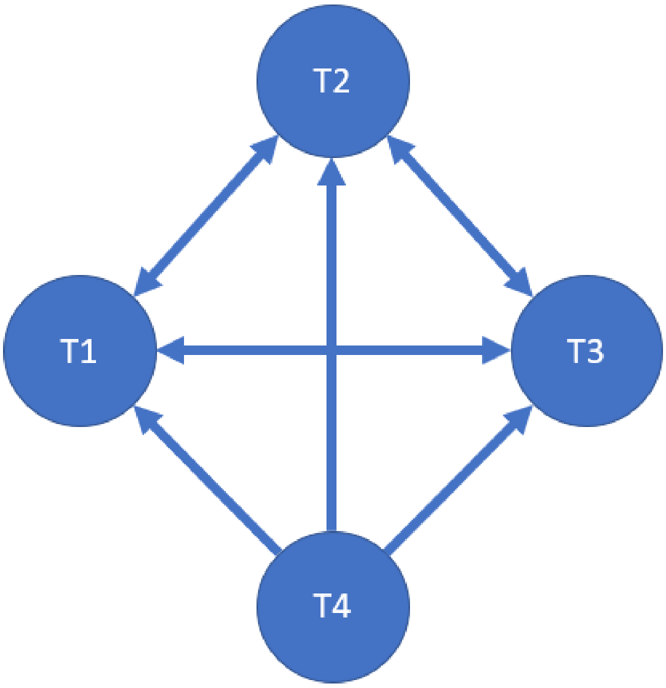
3 Transaction Management

3.1 Precedence graph



3.2 Serializability

A schedule is conflict serializable if and only if its precedence graph is acyclic. If a graph is not conflict serializable, it implies that there are some anomalies with the interleaved schedule, which is undesirable. Conflict serializability also implies serializability. If something is serializable it means that the interleaved transactions could be run after each other with the same result. This property alone does not imply conflict serializability.

The given schedule is not conflict serializable because its precedence graph is not acyclic. It is also not serializable, because T4 writes B first and then T2 reads and writes B. T2 clearly cannot be the first transaction from T2 and T4 because T2 writes B the last. However T4 also cannot be before T2 because then T2 would read the changed data by T4, while according to the schedule it reads the original unchanged B. Therefore, the schedule is not serializable (which also implies that it is not conflict serializable).

3.3 Strict 2PL

T1	X(A)Wait T2 X(A)Wait T2
T2	S(A) R(A) S(B)Wait T4
T3	X(C)Wait T4 X(C)W(C)
T4	S(C)R(C) S(A)R(A) X(B)W(B) cmt.

T1	
T2	S(B)R(B) X(B)W(B) S(C)Wait T3
T3	X(A)Wait T2
T4	

3.4 Waits-for-graph