Algorithm to determine conflict serializability

Input: Concurrent schedule S

Output: S is conflict serializable or not (conflict equivalent to serial schedules)

Step1: for each transaction Ti participating in schedule S, create a node labelled Ti in the precedence graph (a directed graph where the vertices are the transactions)

Step2: For each transaction in S, draw an edge Ti → Tj in the graph if one of the following three conditions holds:

(i) Ti ~~reads~~ executes read(Q) before Tj executes write (Q)

(ii) Ti executes write (Q) before Tj executes read (Q)

(iii) Ti executes write (Q) before Tj executed write (Q)

Step 3: If the graph contains no cycle, then schedule S is conflict serizable.

NOTE: A serial schedule or serializability order of the transactions can be obtained through topological sorting of the graph. (linear ordering of vertices of the graph such that for every directed edge Ti → Tj, vertex Ti comes before vertex Tj in the ordering) in the serial schedule S' equivalent to S.