



Problem 14.5: Since every conflict-serializable schedule is view serializable, why do we emphasize conflict serializability rather than view serializability?

Most of the concurrency control protocols (protocols for ensuring that only serializable schedules are generated) used in practice is based on conflict serializability they actually permit only a subset of actually permit only a subset of conflict serializable schedules. The general form of view serializability is very expensive to test, and only a very restricted form of it is used for concurrency control.

Problem 14.6: Consider the precedence graph of Figure 14.16. Is the corresponding schedule conflict serializable? Explain your answer.

There is a serializable schedule corresponding to the precedence graph below, since the graph is acyclic. A possible schedule is obtained by doing a topological sort, that is, T_1 , T_2 , T_3 , T_4 , and T_5 .

Problem 14.7: What is a cascadeless schedule? Why is cascadelessness of schedules desirable? Are there any circumstances under which it would be desirable to allow noncascadeless schedules? Explain your answer.

A cascadeless schedule is one where, for each pair of transactions T_i and T_j such that T_j reads data items previously written by T_i , the commit operation of T_i appears before the read operation of T_j . Cascadeless schedules are desirable because the failure of a transaction does not lead to the aborting of any other transaction. Of course this comes at the cost of less concurrency. If failures occur rarely, so that we can pay the price of cascading aborts for the increased concurrency, noncascadeless schedules might be desirable.

Problem 14.14: Explain the distinction between the terms serial schedule and serializable schedule.

A schedule in which all the instructions belonging to one single transaction appear together is called a serial schedule. A serializable schedule has a weaker restriction that it should be equivalent to some serial schedule. There are two definitions of schedule equivalence – conflict equivalence and equivalence.

Problem 14.17: What is a recoverable schedule? Why is recoverability of schedules desirable? Are there any circumstances under which it would be desirable to allow nonrecoverable schedules? Explain your answer

A recoverable schedule is one where, for each pair of transactions T_i and T_j such that T_j reads data items previously written by T_i , the Commit operation of T_i appears before the commit operation of T_j . recoverable schedule are desirable because failure of a transaction might otherwise bring the system into an irreversibly inconsistent state. Nonrecoverable schedules may sometimes be needed when updates must be made visible early due to time constraints,

even if they have not yet been committed, which may be required for very long duration transactions.