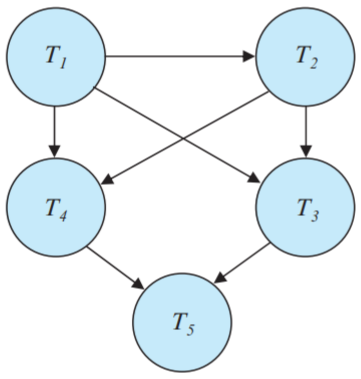
数据库week11

17.6

> Consider the precedence graph of Figure 17.16. Is the corresponding schedule conflict serializable? Explain your answer.
>
> 
>
>
> **Figure 17.16** Precedence graph for Practice Exercise 17.6.

Yes.

Since of that a schedule is conflict serializable if and only if its precedence graph is acyclic. By doing a topological sort{ T1, T2, T3, T4, T5},we can get a correct schedule.

17.7

> 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.

Cascadeless schedule:

- Cascading rollbacks cannot occur;
- For each pair of transactions T_i and T_j such that T_j reads a data item 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.

If failures occur rarely, so that we an pay the price of cascading aborts for the increased concurrency. In this case , noncascadeless schedules are desirable comparing other schedules.