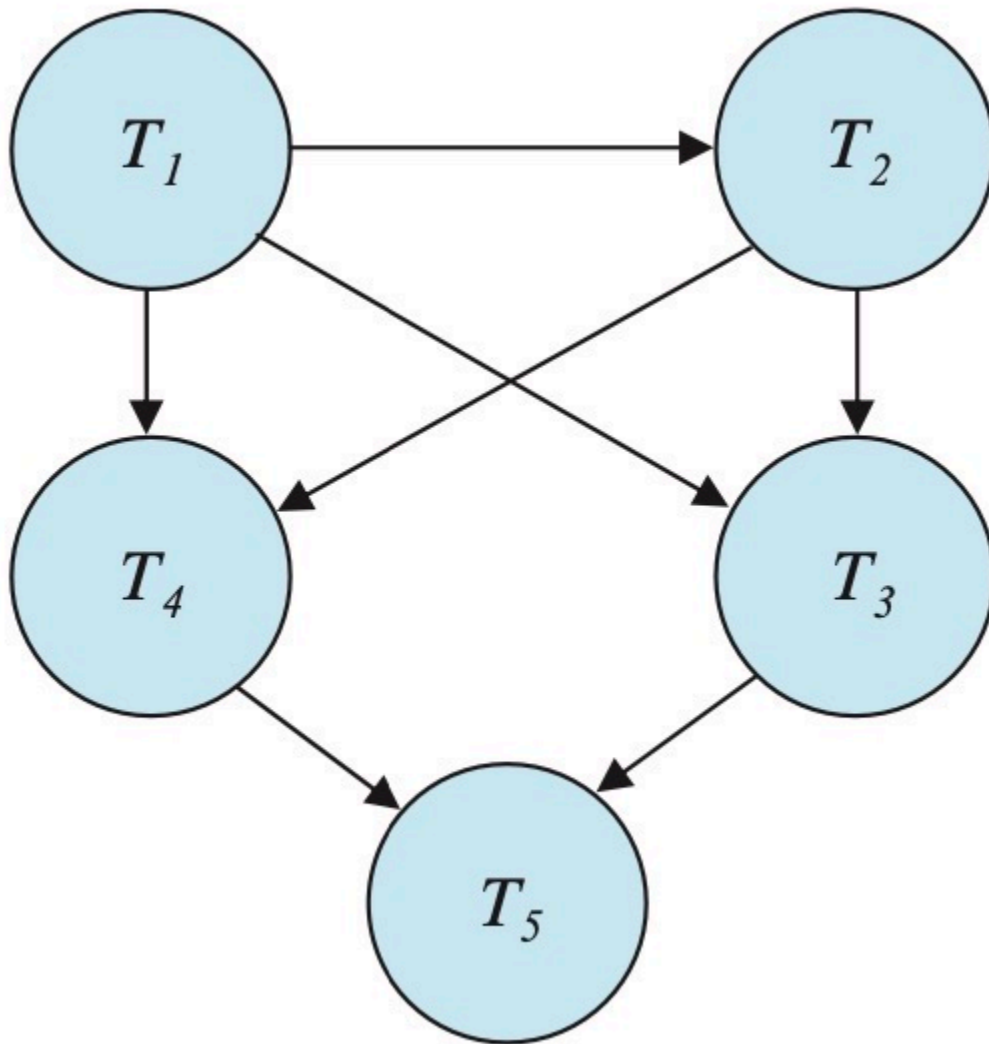


17.6 Consider the precedence graph of Figure 17.16. Is the corresponding schedule conflict serializable? Explain your answer.



是 conflict serializable 的，因为图中没有环，可以按照 $T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow T_4 \rightarrow T_5$ 或者 $T_1 \rightarrow T_2 \rightarrow T_4 \rightarrow T_3 \rightarrow T_5$ 的顺序执行，都可以和一个串行执行等价。

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 是指如果一个事务 T_j 读取了一个事务 T_i 的写入的数据，那么 T_i 必须在 T_j 之前提交。

这样可以避免 cascade rollback 的问题。cascade rollback 是指如果一个事务 T_j 读取了一个事

务 T_i 的写入的数据，然后 T_i 回滚了，那么 T_j 也必须回滚。

在某些情况下，可能会允许 **noncascadeless schedules**，比如在某些事务中，可能会有一些事务是只读的，不会对数据进行修改，这样即使发生了 **cascade rollback** 也不会有问题。