

1. 证明两阶段锁协议能够保证冲突可串行化，且事务可以根据它们的锁点
进行可串行化。

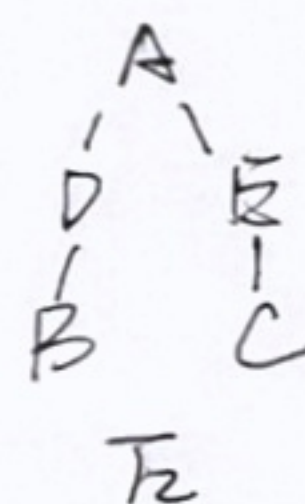
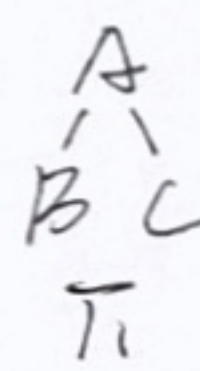
证明：

两阶段锁协议有增长阶段（只能请求锁，不能释放锁）和缩减阶段（只能释放锁，不能请求锁），这就保证了一个事务对数据的修改在一段时间内完成且此阶段内数据被上锁，这就保证了事务并发中对数据修改是串行的，所以事务可以根据它们的锁点进行可串行化，所以同一两阶段协议能够保证冲突可串行化。

2. a. 应该是对关系 r 获得 IS 锁，对元组获得 S 锁。
- b. 应该是对关系 r 的 IX 锁，对元组获得 X 锁。

c. 问题：必须给这些元组加锁，这十分费时
措施：利用多粒度锁协议来解决。

3. 在每对顶点之间插入一个虚拟结点能获得更好的并行性。
因为增加了层数，降低了并发事务的竞争。如 T_2 是下增加虚拟结点，若 T_1 和 T_2 需要对 B、C 加锁会造成对 A 的竞争，而在 T_2 上却不会。



4. 多粒度锁协议中，隐式锁和显式锁的区别：

显式锁：通过明确的锁请求和操作来管理锁，是细粒度的控制。

隐式锁：由数据库自动管理的锁，根据数据自动确定锁，减轻了锁管理的负担。

5. T_{34}
Lock-S(A)
~~read~~ read(A)
Lock-S(B)
read(B)
if A=0 then B:=B+1
Lock-X(B)
write(B)
unlock(A)
unlock(B)

T_{35}
Lock-S(B) (read(B))
Lock-S(A)
read(A)
if B=0 then A:=A+1
Lock-X(A)
write(A)
unlock(B)
unlock(A)

会发生死锁如下

T_{34}
L-S(A)
R(B)

T_{35}
L-S(B)
R(B)
L-S(A)
R(A)
if B=0 then A:=A+1
Lock-X(A)
write(A)
unlock(B)
unlock(A)

if A=0 then B:=B+1
L-X(B)
W(B)
unlock(A)
unlock(B)

L-S(B)
R(B)