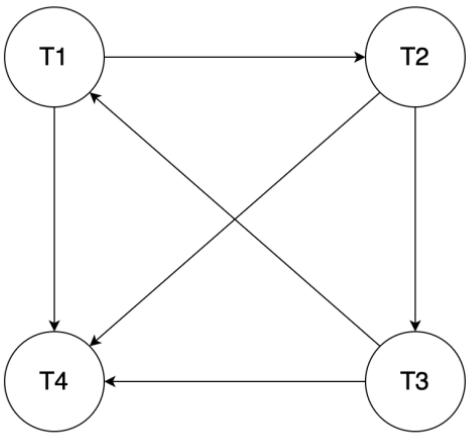
课后自由练习 #6 参考解答

1. 判断下面的并发调度是否冲突可串？如果是，请给出冲突等价的串行调度事务顺序；如果不是，请解释理由。

> w3(D); r1(A); w2(A); r4(A); r1(C); w2(B); r3(B); r3(A); w1(D); w3(B); r4(B); r4(C); w4(C); w4(B)

解答：

根据并发调度作出其对应的优先图：



可以看到该优先图中存在环路，因此该并发调度**不是冲突可串。**

---

2. 证明：如果一个并发调度 S 中的所有事务都遵循 2PL，则该调度必定是冲突可串调度。

证明：

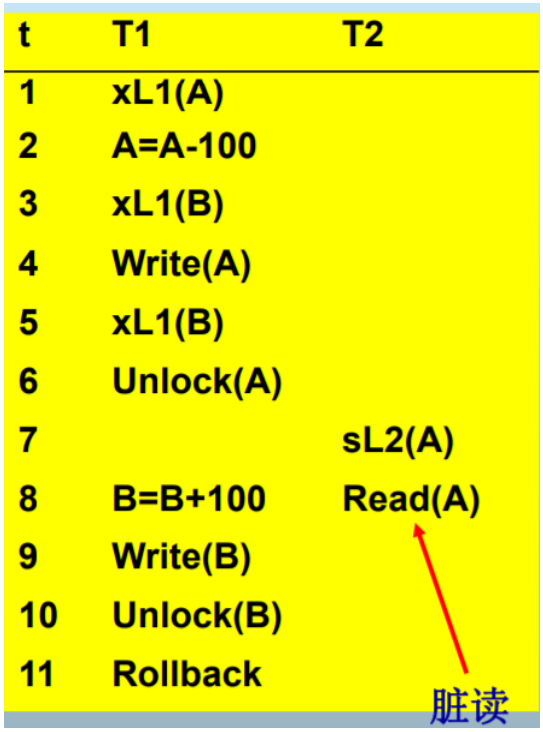
反证法。如果发生了不可串调度情况，则该调度不满足冲突可串性，那么它的优先图必存在环。设其上事务为$T_1, T_2 \ldots T_n$，操作的数据为$D_1, D_2 \ldots D_n$，则有

- $uL_1(D_1) \rightarrow L_2(D_1)$
- $L_2(D_1) \rightarrow uL_2(D_2)$
- $uL_2(D_2) \rightarrow L_3(D_2)$
- **...**
- $uL_n(D_n) \rightarrow L_1(D_n)$
- $L_1(D_n) \rightarrow uL_1(D_1)$

事务T_1在释放锁之后又申请了锁，违背了**2PL**，故不成立。

---

3. 如果一个并发调度中的所有事务都遵循两阶段锁协议，该并发调度还会出现脏读问题吗？如果不会，请解释理由；如果会，请给出一个例子。



---

4. 考虑下面两个事务：

> T1 = r1(A) w1(A) r1(B) w1(B)
> T2 = r2(B) r2(A) w2(A) w2(B)

假设调度器插入了 X 锁动作，如下所示：

> T1 = xL1(A) r1(A) w1(A) xL1(B) r1(B) w1(B), AFTER COMMIT: U1(A) U1(B)
> T2 = xL2(B) r2(B) xL2(A) r2(A) w2(A) w2(B), AFTER COMMIT: U2(A) U2(B)

现在 T1 和 T2 并发执行，问：

1）T1、T2 并发执行能保证数据库的一致性吗？为什么？

可以。 因为这是两阶段锁，两阶段锁的调度一定是可串化调度，因此可以保证一致性。（死锁由系统处理）

2）如果不使用上面的加锁方式，而是约定事务按先 A 后 B 的顺序请求锁，T1、T2 并发执行会产生死锁吗？为什么？

不会。 约定顺序之后，T1 和 T2 都需要先申请 A 上的锁，无论哪个事务申请成功，另一个事务都只能等待 A 锁的释放，而不能申请其他锁。等申请成功的事务提交释放 A 锁后，另一个事务才能继续执行，因此不会产生死锁。

3）如果使用等待-死亡 (wait-die) 死锁预防方案，假设 T1 先开始，那么如果产生死锁时系统将采取什么样的动作？

如果采用 wait-die 方案，最初 T1 获得 A 上的锁，T2 获得 B 上的锁，T1 申请 B 上的锁发现被 T2 占用，因为 T1 先开始，所以 T1 进入等待状态，当 T2 申请 A 上的锁发现被 T1 占用，开始回滚，此时 T1 获得 B 上的锁顺利运行，T2 在 T1 提交以后获得B上的锁继续执行。