

**定理:** 由2PL事务所构成的任意合法调度  $S$  都是冲突可串行化的。

**证明:**

1. 当调度  $S$  仅由一个事务组成时，调度  $S$  是冲突可串行化的。
2. 假设：由  $(n-1)$  个2PL事务所构成的任意一个合法调度都是冲突可串行化的。
3. 设调度  $S$  涉及  $n$  个2PL事务： $T_1, T_2, \dots, T_n$ ，并且  $T_i$  是调度  $S$  中第一个有解锁动作的事务，则我们可以得到以下结论：
  - \* 可以将  $T_i$  的所有动作不经过任何冲突而移动到调度  $S$  的最前面。

□ 设在  $T_i$  中有一个动作  $w_i(A)$  (或  $r_i(A)$ )，如果调度  $S$  在该动作的前面有一个与之冲突的动作  $w_j(A)$  ( $i \neq j$ )，那么调度  $S$  的情况必是：

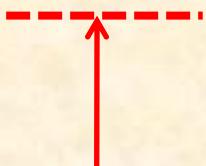
...;  $w_j(A)$ ; ...;  $u_j(A)$ ; ...;  $l_i(A)$ ; ...;  $w_i(A)$ ; ...

$\because T_i$  是调度  $S$  中第一个有解锁动作的事务

$\therefore$  在  $u_j(A)$  之前必存在  $T_i$  中的一个解锁动作 (如  $u_i(B)$ )，

则调度  $S$  变为： ...;  $w_j(A)$ ; ...;  $u_i(B)$ ; ...;  $u_j(A)$ ; ...;

$l_i(A)$ ; ...;  $w_i(A)$ ; ...



在上述的调度  $S$  中，仅考虑与事务  $T_i$  有关的动作序列

□ 在上述的调度  $S$  中，仅考虑与事务  $T_i$  有关的动作序列： $\dots; u_i(B); \dots; l_i(A); \dots; w_i(A); \dots$

■ 这不符合 2PL 事务的定义，与  $T_i$  是 2PL 事务相矛盾。

$\therefore$  在  $T_i$  的每个动作  $w_i(A)$  (或  $r_i(A)$ ) 之前，都不存在与其产生冲突且属于其它事务的动作

$\therefore$  结论 \* 成立

- 根据得到的结论 \*，我们可以将调度S转换为另一个冲突等价的调度 $S'$ ：  
**( $T_i$ 的所有动作);(其它( $n-1$ )个2PL事务的动作)**
- 并且维持其后半部分动作在调度S中的原有顺序不变；
- 其中 $T_i$ 的封锁/解锁动作在转换后可以恢复到调度 $S'$ 中去。

□ 调度  $S'$ :

定理证明（续）

( $T_i$ 的所有动作);(其它( $n-1$ )个2PL事务的动作)

- ∴ 调度  $S$  是一个合法调度
- ∴ 调度  $S'$  的后半部分是其它( $n-1$ )个2PL事务的一个合法调度
- ∴ 根据步骤2的归纳假设的前提可得：调度  $S'$  的后半部分是其它( $n-1$ )个2PL事务的一个‘冲突可串行化’的调度，即冲突等价于这( $n-1$ )个2PL事务的某个串行调度。
- ∴ 调度  $S'$  冲突等价于这  $n$  个2PL事务的某个串行调度
- ∴ 调度  $S'$  是冲突可串行化的
- ∴ 调度  $S$  也是冲突可串行化的

证毕

	Read 操作		Write 操作	
	锁类型	封锁时间	锁类型	封锁时间
<b>READUNCOMMITTED</b> 未提交读	No Lock	—	不允许执行 <b>Write</b> 操作	
<b>READCOMMITTED</b> 提交读	共享锁	读操作		
<b>READREPEATABLE</b> 可重复读	共享锁	事务	排它锁	事务
<b>SERIALIZABLE</b> 可串行化	共享锁	事务		

事务的隔离级别与封锁策略之间的关系

