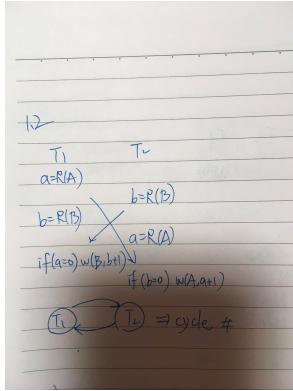


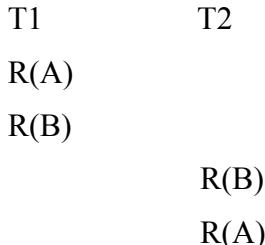
1.1. No matter the T1 is interleaved with T2 or not. If T1 finishes before T2, it would execute if($a==0$) W(B,1) then if($b=0$) W(A,0) => B=1, A=0. If T2 finishes before T1, it would be B=0 A=1 (same reason as above). Both case matches the requirement.

1.2. If not serializable => precedence graph has cycle



A=1 and B=1 eventually.

1.3. Assume

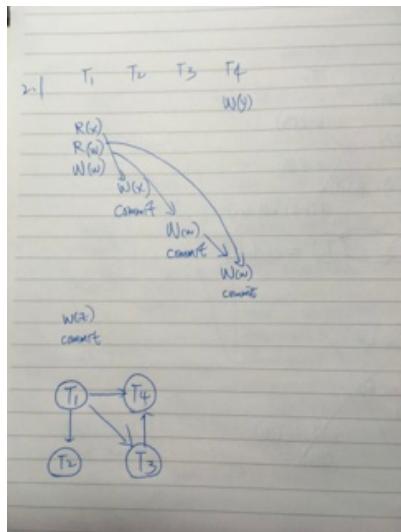


If($a=0$) W(B,b+1)

If($b=0$) W(A,a+1)

We may have the read lock for A and B in T1 then release them. Then have another read lock for A and B in T2. But we could not acquire new lock once T1 or T2 release any lock, so there is no writing lock for "If($a=0$) W(B,b+1)" => deadlock occurs => not obey strict 2PL.

2.1.



2.2. No cycle => it is serializable

2.3. topological sorting => remove no incoming edges's node

$T_1 \Rightarrow T_2 \Rightarrow T_3 \Rightarrow T_4$

$T_1 \Rightarrow T_3 \Rightarrow T_2 \Rightarrow T_4$

$T_1 \Rightarrow T_3 \Rightarrow T_4 \Rightarrow T_2$