Sheet 6 sol

18.22 Prove that strict 2PL guarntees strict schedule!

* Since No other transc can read or write an item written by a transct until T has committed so, strict   Q.E.D

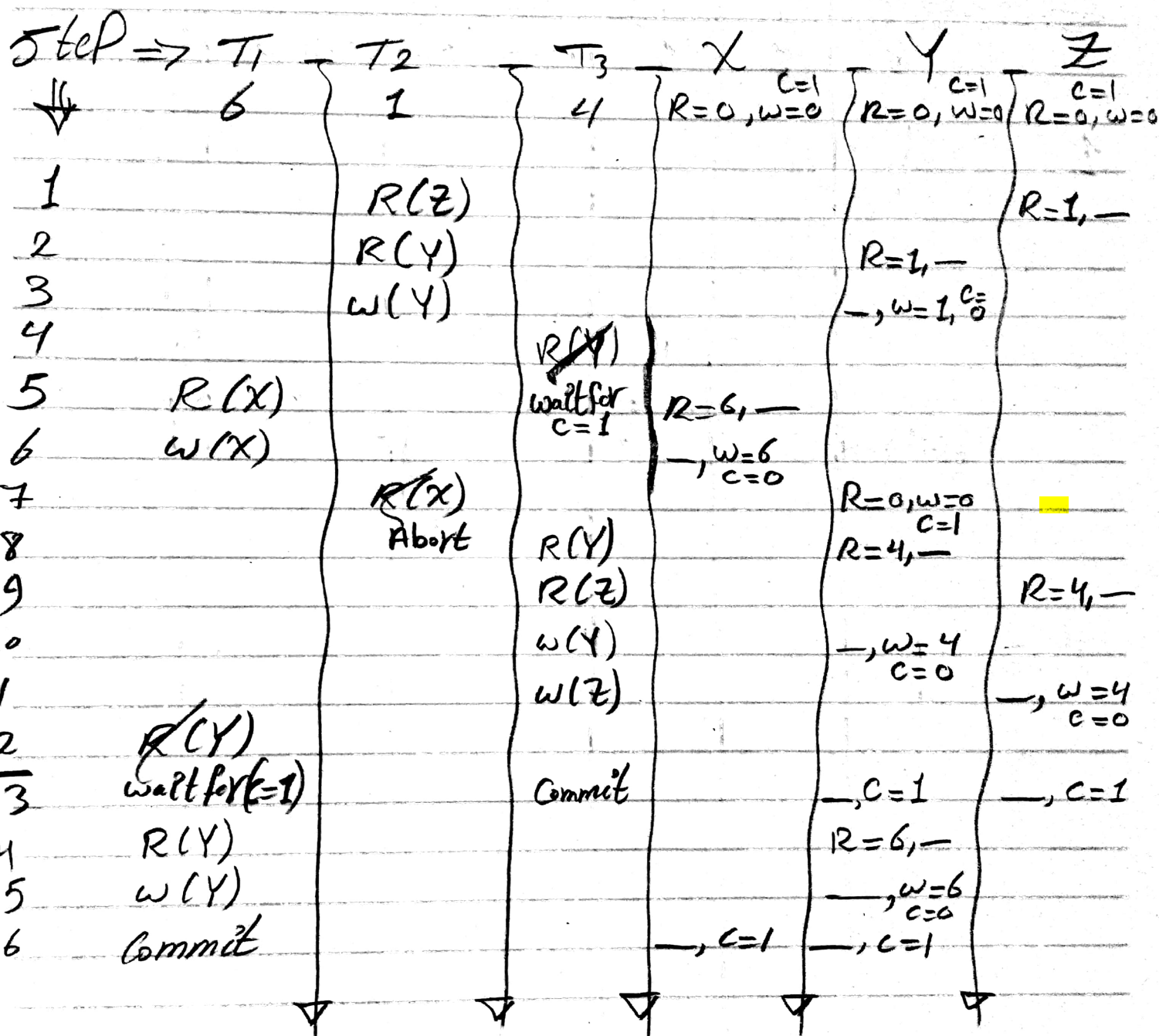**EXtra** Prove that strict 2PL guarntees serializable schedule!

* Since the write lock is held by only one transc → No cycles are allowed → Acyclic Precedence graph → serializable!

18.24 Prove that cautious waiting avoids deadlock!

* Because it avoids waiting; if transaction cannot obtain lock, aborted immediately and restarted aftr time t. So, No cycles in its wait-for graph so, No deadlocks!

Timestamp

18.25 using schedule (b) ⚡ $T_S(T_1)=6$, $T_S(T_2)=1$, $T_S(T_3)=4$

Step ⇒	T_1	T_2	T_3	X (c=1)	Y (c=1)	Z (c=1)
⇓	6	1	4	R=0, w=0	R=0, w=0	R=0, w=0
1		R(Z)				R=1, —
2		R(Y)			R=1, —	
3		W(Y)			—, w=1, c=0	
4			~~R(Y)~~			
5	R(X)		waitfor c=1	R=6, —		
6	W(X)			—, w=6, c=0		
7		~~R(X)~~			R=0, w=0, c=1	
8		Abort	R(Y)		R=4, —	
9			R(Z)			R=4, —
10			W(Y)		—, w=4, c=0	
11			W(Z)			—, w=4, c=0
12	~~R(Y)~~					
13	waitfor c=1		Commit		—, c=1	—, c=1
14	R(Y)				R=6, —	
15	W(Y)				—, w=6, c=0	
16	Commit			—, c=1	—, c=1	

* Assume that Abort or delay actions, take 1-timestamp each
* Starting from step 13 → Added Commit op to show u the Complete Sol, u can just stop there!
* Notice, at step 7, we undo(ne) the effect of T_2
* we were filling the Data one by one!

18.20

Step →	T1 (6)	T2 (1)	T3 (4)	Z0	Y0	Y1	X0	X6	Y4	Z4	Y6
1		R(Z)		R=1							
2		R(Y)			R=1						
3		w(Y)				R=1 w=1					
4			R(Y)			R=4					
5			R(Z)	R=4							
6	R(X)						R=6	R=6 w=6			
7	w(X)										
8			w(Y) w(Z)						R=4 w=4	R=4 w=4	
9								read			
10		R(X)						R=6			
11		R(Y)									
12		w(Y)									R=6 w=6
13		w(X) Abort									