S1: r1(X), w2(Y), w1(Y), w3(Z), c1, w2(Z), w2(X), c2, w3(X), c3.

S2: r1(Y), r1(Z), w1(Z), w2(Z), r2(Y), r3(X), w3(X), w1(X), c1, w2(Y), r3(Z), c2, c3.

Schedule

(S1) :

Q1 Basic 2PL (wait-die policy)

T_1	T_2	T_3
S1-lock(x)		
$r_1(x)$		
	x2-lock(Y)	
	$w_2(Y)$	
x1-lock(Y) (wait for T_2 on Y)		
		x3-lock(z)
		$w_3(z)$
	x2-lock(z) (wait for T_3 on z)	
		x3-lock(x) (abort due to T_1 & restart with T_3)
	$w_2(z)$, wake-up	
	x2-lock(x) (abort due to T_1 & restart with T_2)	
unlock(Y) wakeup		
$w_1(Y)$		
unlock(Y)		
c_1		
	T_2 can restart here	T_3 can restart here

Q2: Strict 2PL (wound-wait policy)

T_1	T_2	T_3
S_1-lock(x)		
r(x)		
	x_2-lock(y)	
	$w_2(y)$	
x_1-lock(y) (wound T_2)		
	Abort due to T_1 & restart with same TS	
$w_1(y)$		
unlock(x)		
		x_3-lock(z)
		$w_3(z)$
c_1		
unlock(y)		
		x_3-lock(x)
		$w_3(x)$
		c_3
		unlock(z)
		unlock(x)
	T_2 can now restart here	

Q#3 Rigorous 2PL (wait-die policy)

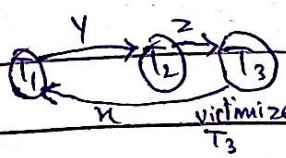
T_1	T_2	T_3
S_1-lock(X)		
$r_1(X)$		
	X_2-lock(Y)	
	$w_2(Y)$	
X_1-lock(Y) (wait for T_2 on Y)		
		X_3-lock(Z)
		$w_3(Z)$
	X_2-lock(Z) (wait for T_3 on Z)	
		X_3-lock(X) (abort due to T_1 & restart with same TS)
	$w_2(Z)$, wake up	
	X_2-lock(X) (abort due to T_1 & restart with same TS)	
$w_1(Y)$		
c_1		
unlock(X)		
unlock(Y)		
	T_2 can restart here	
		T_3 can restart here

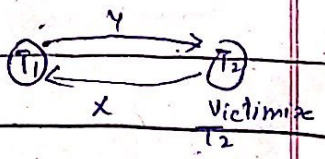
Q 3: Rigorous 2PL (wound-wait)

T_1	T_2	T_3
S_1-lock(X)		
$r_1(X)$		
	x_2-lock(Y)	
	$w_2(Y)$	
x_1-lock(Y) (wound T_2)	(Abort due to T_1 restart with same TS)	
$w_1(Y)$		
		x_3-lock(Z)
		$w_3(Z)$
C_1		
unlock(X)		
unlock(Y)		
		x_3-lock(X)
		$w_3(X)$
		C_3
		unlock(Z)
		unlock(X)
	T_2 can restart now	

Q#5: Rigorous 2PL : deadlock detection/wait for graph

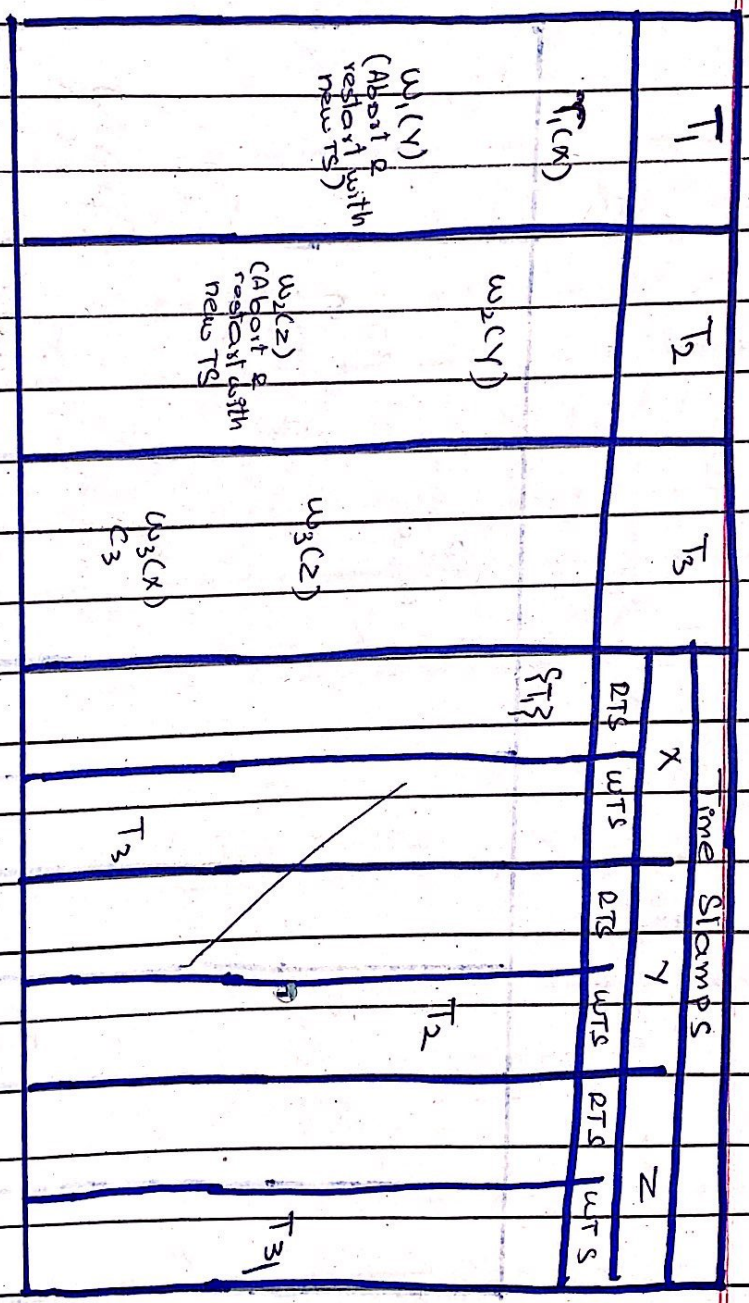
T_1	T_2	T_3
S_1-lock(x)		
$r_1(x)$		
	x_2-lock(Y)	
	$w_2(Y)$	
x_1-lock(Y) (wait for T_2 on Y)		
		x_3-lock(Z)
		$w_3(Z)$
	x_2-lock(Z) (wait for T_3 on Z)	
		x_3-lock(X) (Victimize T_3 and restart with same TS)
	$w_2(Z)$ wake-up	
	x_2-lock(X) (Victimize T_2 and restart with same TS)	
$w_1(Y)$, wake-up		
c_1		
unlock(X)		
unlock(Y)		
	T_2 can restart here	
		T_3 can restart here



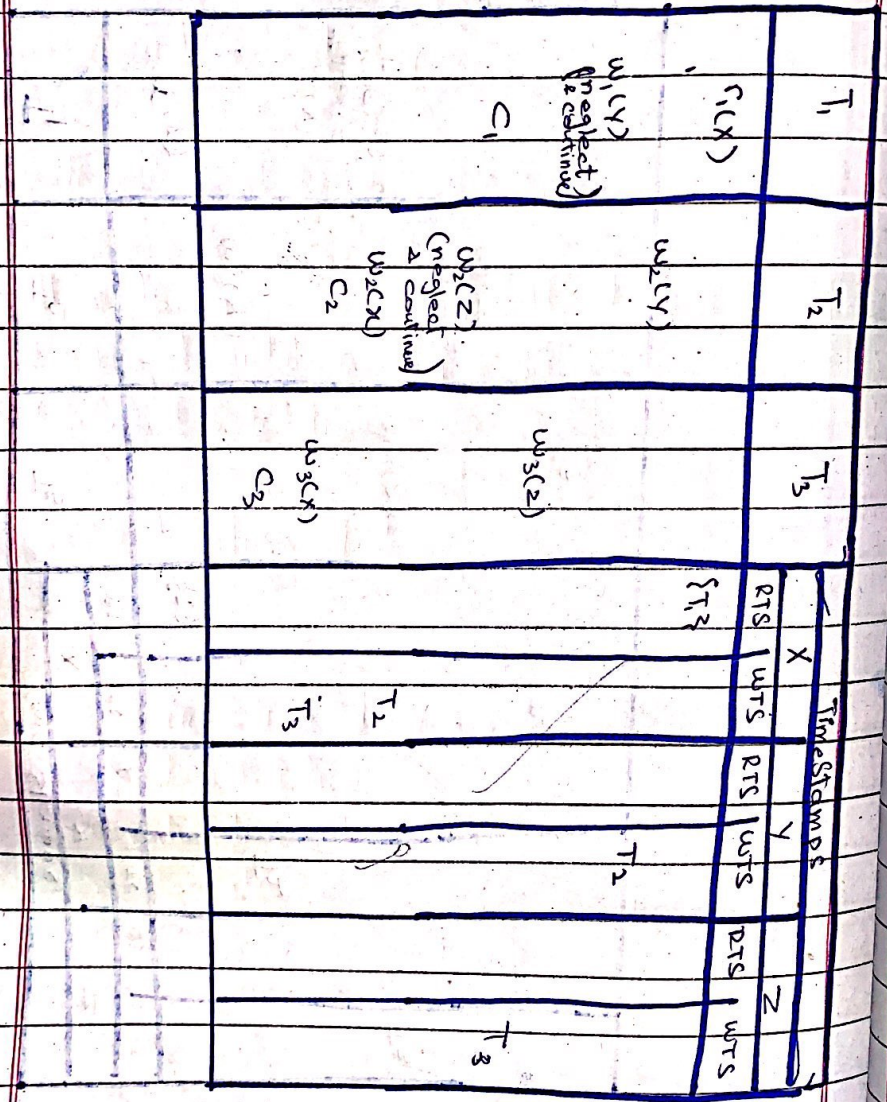


Q#6: Basic Timestamp Ordering

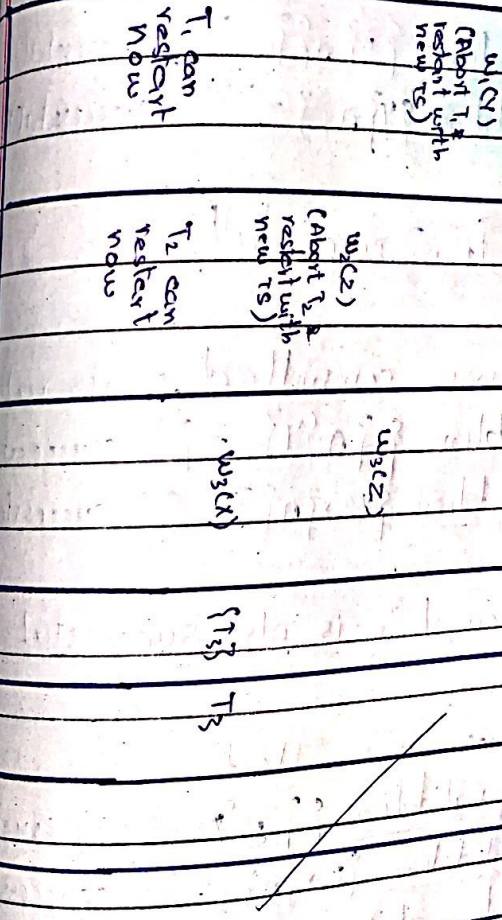
Q#7 Strict TimeStamp Ordering



Q#8: Time Stamp Ordering Thomas write rule



Q#9 Multi-version



Q#9 Multi-version Time stamp ordering

T_1	T_2	T_3	Time Stamps							
			X		Y		Z			
			RTS	WTS	RTS	WTS	RTS	WTS		
			{3}	T_0	{3}	T_0	{3}	T_0		
$r_1(x)$			{T1}							
	$w_2(y)$				{T2}	T_2				
$w_1(y)$ (Abort T_1 & restart with new TS)							{T3}	T_3		
T_1 can restart now				T_3						
	$w_2(z)$ (Abort T_2 & restart with new TS)									
	T_2 can restart now									
		$w_3(z)$								
		$w_3(x)$	{T3}							

Q10: Validation

- T_1 committed first. So it's validation starts first
  Backward validation: $\{x\} \cap \{\} = \{\}$ (successful)
  Forward validation: $\{y\} \cap \{\} = \{\}$ (successful)

  validation of T_1 is successful!

  Afterwards T_2 was committed. so it's validation
  Backward validation: $\{\} \cap \{y\} = \{\}$ (successful)
  Forward validation: $\{y,z,x\} \cap \{\} = \{\}$ (successful)

  So validation of T_2 is also successful

  Than T_3 was committed
  Backward validation: $\{\} \cap \{y,z,x\} = \{\}$ (successful)
  Forward validation: $\{z,x\} \cap \{\} = \{\}$ (successful)
  So validation of T_3 is also successful

(S_2)

Q#1: Basic 2PL (wait-die)

T_1	T_2	T_3
S_1-lock(Y)		
$r_1(Y)$		
S_1-lock(z)		
$r_1(z)$		
X_1-lock(z)		
$w_1(z)$		
	X_2-lock(z) (Abort due to T_1 & restart with same TS)	
		S_3-lock(x)
		$r_3(x)$
		X_3-lock(x)
		$w_3(x)$
X_1-lock(x) (wait for T_3 on x)		
		S_3-lock(z) (Abort due to T_1 & restart with same TS)
unlock(Y), wake up		
unlock(z)		
$w_1(x)$		
unlock(x)		
C_1		
	T_2 can restart now	
		T_3 can restart now

Q#2: Strict 2PL (wound-wait)

T_1	T_2	T_3
S_1-lock(Y)		
$r_1(Y)$		
S_1-lock(Z)		
$r_1(Z)$		
x_1-lock(Z)		
$w_1(Z)$		
	x_2-lock(Z) (wait for T_1 on Z)	
		S_3-lock(X)
		$r_3(X)$
		x_3-lock(X)
		$w_3(X)$
x_1-lock(X) (wound T_3)		(Abort due to T_1 & restart with same T_3)
unlock(Y)		
$w_3(X)$		
C_1		
unlock(Z)		
unlock(X)		
	$w_2(Z)$, wakeup	
	S_2-lock(Y)	
	$r_2(Y)$	
	x_2-lock(Y)	
	$w_2(Y)$	

T_1	T_2	T_3
	C.S	
	unlock(x)	
	unlock(y)	
		T_3 can restart now

Q#3: Rigorous 2PL (wait-die)

T_1	T_2	T_3
S_1-lock(Y)		
$r_1(Y)$		
S_1-lock(Z)		
$r_1(Z)$		
X_1-lock(Z)		
$w_1(Z)$		
	X_2-lock(Z) (Abort due to restart with same TS)	
		S_3-lock(X)
		$r_3(X)$
		X_3-lock(Y)
		$w_3(X)$
X_1-lock(X) (wait for T_3 on X)		
		S_3-lock(Z) (Abort due to T_1 & restart with same TS)
$w_1(X)$		
C_1		
unlock(X)		
unlock(Y)		
Unlock(Z)		
	T_2 can restart now	
		T_3 can restart now

Q#4 Rigorous 2PL (wound-wait)

T_1	T_2	T_3
S_1-lock(Y)		
$r_1(Y)$		
S_1-lock(Z)		
$r_1(Z)$		
X_1-lock(Z)		
$w_1(Z)$		
	X_2-lock(Z) (wait for T_1 on Z)	
		S_3-lock(X)
		$r_3(X)$
		X_3-lock(Y)
		$w_3(X)$
X_1-lock(X) (wound T_3)		
		(Abort due to T_1 & restart with same TS)
$w_1(X)$		
C_1		
unlock(X)		
unlock(Y)		
Unlock(Z)		
	$w_2(Z)$ wakeup	
	S_2-lock(Y)	
	$r_2(Y)$	
	X_2-lock(Y)	

Q#4 Rigorous 2PL (wound-wait)

T_1	T_2	T_3
S_1-lock(Y)		
$r_1(Y)$		
S_1-lock(z)		
$r_1(z)$		
x_1-lock(z)		
$w_1(z)$		
	x_2-lock(z); (wait for T_1 on z)	
		S_3-lock(x)
		$r_3(x)$
		x_3-lock(x)
		$w_3(x)$
x_1-lock(x) (wound T_3)		(Abort due to T_1 & restart with same TS)
$w_1(x)$		
c_1		
unlock(x)		
unlock(Y)		
unlock(z)		
	$w_2(z)$ wakeup	
	S_2-lock(Y)	
	$r_2(Y)$	
	x_2-lock(Y)	

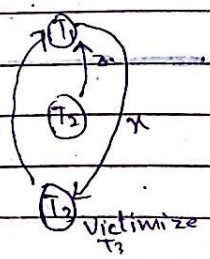
	$w_2(Y)$	
	c_2	
	unlock(z)	
	unlock(Y)	
		(T_3 can be restarted here)

Q#5: Rigorous 2PL - deadlock detection (wait for graph)

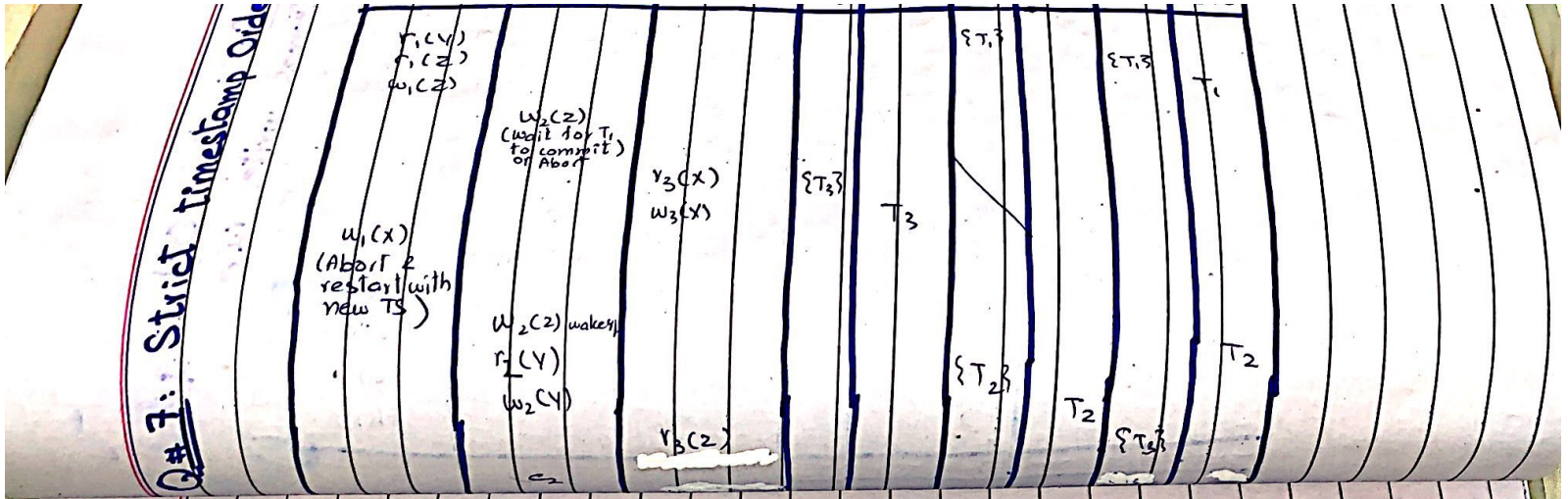
T_1	T_2	T_3
S1-lock(Y)		
$r_1(Y)$		
S1-lock(z)		
$r_1(z)$		
X2-lock(z)		
$w_1(z)$		
	X2-lock(z) (wait for T_1 on z)	
		S3-lock(x)
		$r_3(x)$
		X3-lock(x)
		$u_3(x)$

X1-lock(X)		
(wait for T_3 on X)		
		X3-lock(Z)
		(victimize T_3 & restart with new TS)
$W_1(X)$, wake up		
C_1		
unlock(Y)		
unlock(Z)		
unlock(X)		
	$W_2(Z)$	
	S2-lock(Y)	
	$R_2(Y)$	
	X2-lock(Y)	
	$W_2(Y)$	
	C_2	
	~~unlock(Z)~~	
	unlock(Y)	

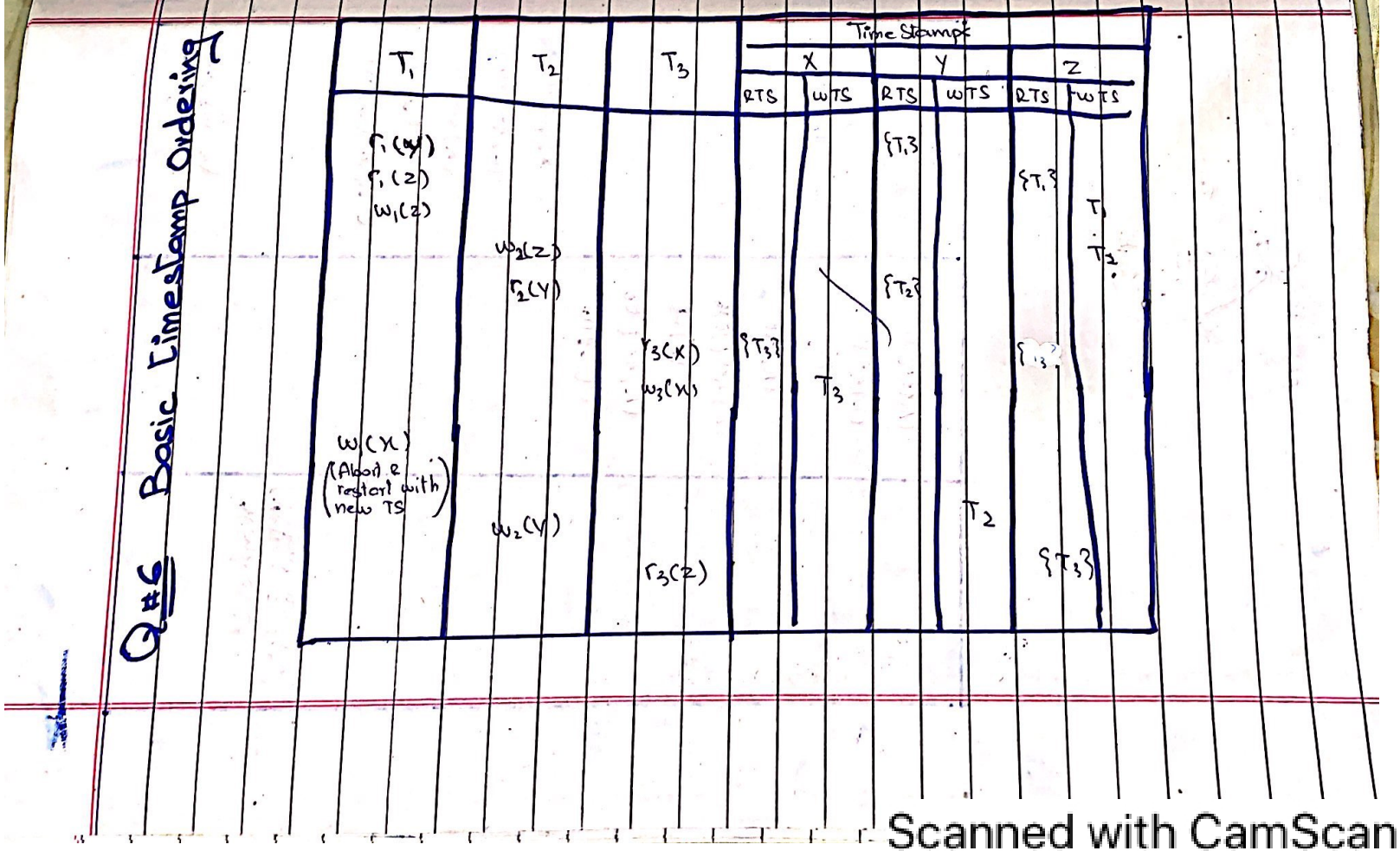
Graph:

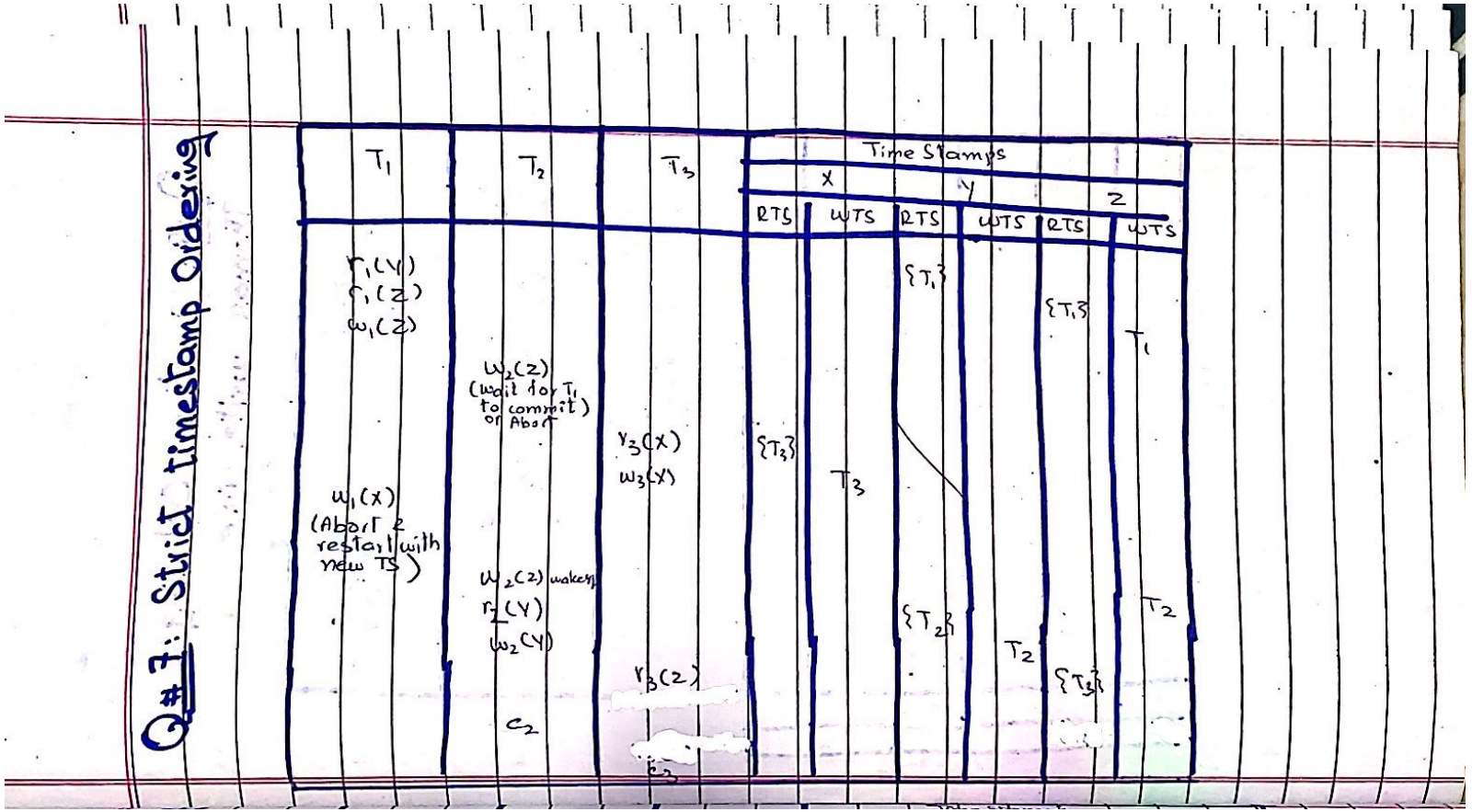


Q#7: Strict timestamp Ord



Q#6 Basic Timestamp Ordering





Q#8. Timestamp Ordering:
Thomas write rule

T_1	T_2	T_3	Timestamps					
			X		Y		Z	
			RTS	WTS	RTS	WTS	RTS	WTS
$r_1(Y)$					$\{T_1\}$			
$r_1(Z)$							$\{T_1\}$	
$w_1(Z)$								T_1
								T_2
	$w_2(Z)$							
	$r_2(Y)$				$\{T_2\}$			
		$r_3(X)$	$\{T_3\}$					
		$w_3(X)$		T_3				
$w_1(X)$ (ignore and continue)								
C_1							T_2	
	$w_2(Y)$							
		$r_3(Z)$						$\{T_3\}$

Q#9: Multi-Version Time Stamp Ordering

T_1	T_2	T_3	Timestamps					
			X		Y		Z	
			RTS	WTS	RTS	WTS	RTS	WTS
			{}	T_0	{}	T_0	{}	T_0
$r_1(X)$					$\{T_1\}$			
$r_1(Z)$							$\{T_1\}$	
$w_1(Z)$							$\{T_1\}$	T_1
	$w_2(Z)$						$\{T_1, T_2\}$	T_1, T_2
	$r_2(Y)$				$\{T_2\}$			
		$r_3(X)$	$\{T_3\}$					
		$w_3(X)$	$\{T_3\}$	T_3				
$w_1(X)$ (Abort & restart with new TS)					$\{T_2\}$	T_2		
	$w_2(Y)$							
		$r_3(Z)$					$\{T_2\}$	
	c_2							
		c_3						

Q#10: Validation

T_1 commits first so it's validation starts first

Backward validation:

$\{Y,Z\} \cap \{\} = \{\}$ (successful)

Forward validation:

$\{Z\} \cap \{X,Y,Z\} = \{X,Z\}$ (failed)

since forward validation failed so delaying validation of T_1

Now validation of T_2 starts

Backward validation:

$\{Y\} \cap \{\} = \{\}$ (successfull)

Forward validation:

$\{Z,Y\} \cap \{Y,Z,X\} = \{Z,Y\}$ (failed)

forward validation of T_2 also failed so also delaying it.

Now validation of T_3 starts

Backward validation:

$\{X,Z\} \cap \{\} = \{\}$ (successfull)

Forward validation:

$\{X\} \cap \{Y,Z\} = \{\}$ (successfull)

so validation of T_3 was successfull

validation of T_1 restarts

Backward validation:

$\{Y,Z\} \cap \{X\} = \{\}$ (successfull)

Forward validation:

$\{X,Z\} \cap \{Y\} = \{\}$ (successfull)

so validation of T_1 is succeed

validation of T_2 restarts

Backward validation:

$\{Y\} \cap \{Z,X\} = \{\}$ (successfull)

Forward validation:

$\{Y,Z\} \cap \{\} = \{\}$ (successfull)

so validation of T_2 is also successfull.