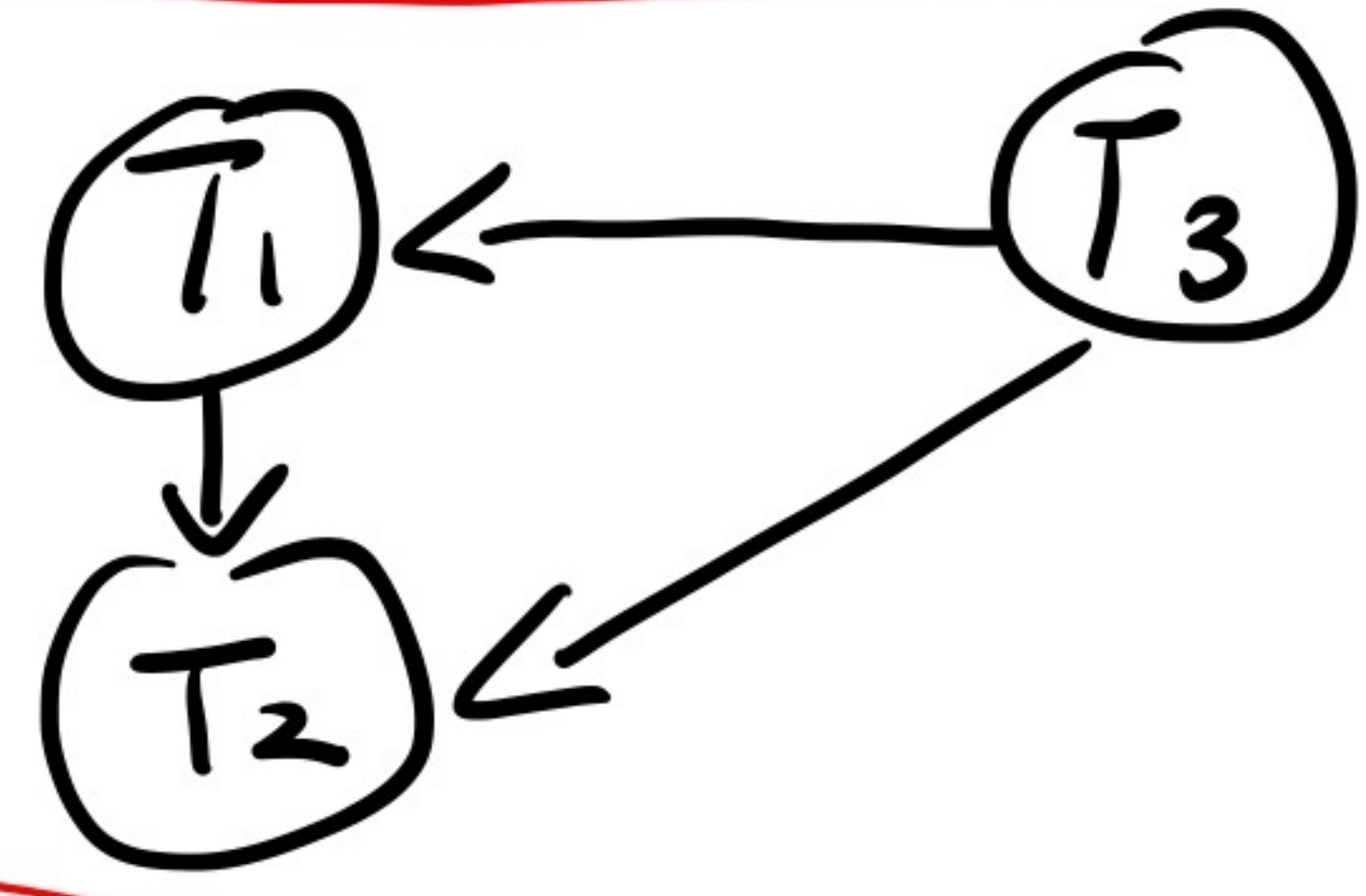


Q2

S1: r1(x) r2(z) r1(y) w2(w) w2(z) r3(z) w3(x) r1(w) w1(y) w3(z)

a) Conflict Graph:



∴ Serializable

b)

S1: r1(x) r2(z) r1(y) w2(w) w2(z) r3(z) w3(x) r1(w) w1(y) w3(z)

SL<sub>1</sub>(x) SL<sub>2</sub>(z) SL<sub>1</sub>(y) XL<sub>2</sub>(w) XL<sub>2</sub>(z) SL<sub>3</sub>(z) [UL<sub>1</sub>(x), XL<sub>3</sub>(x)]

*T<sub>1</sub>: unable to acquire lock.*  
*T<sub>1</sub> shrinking*

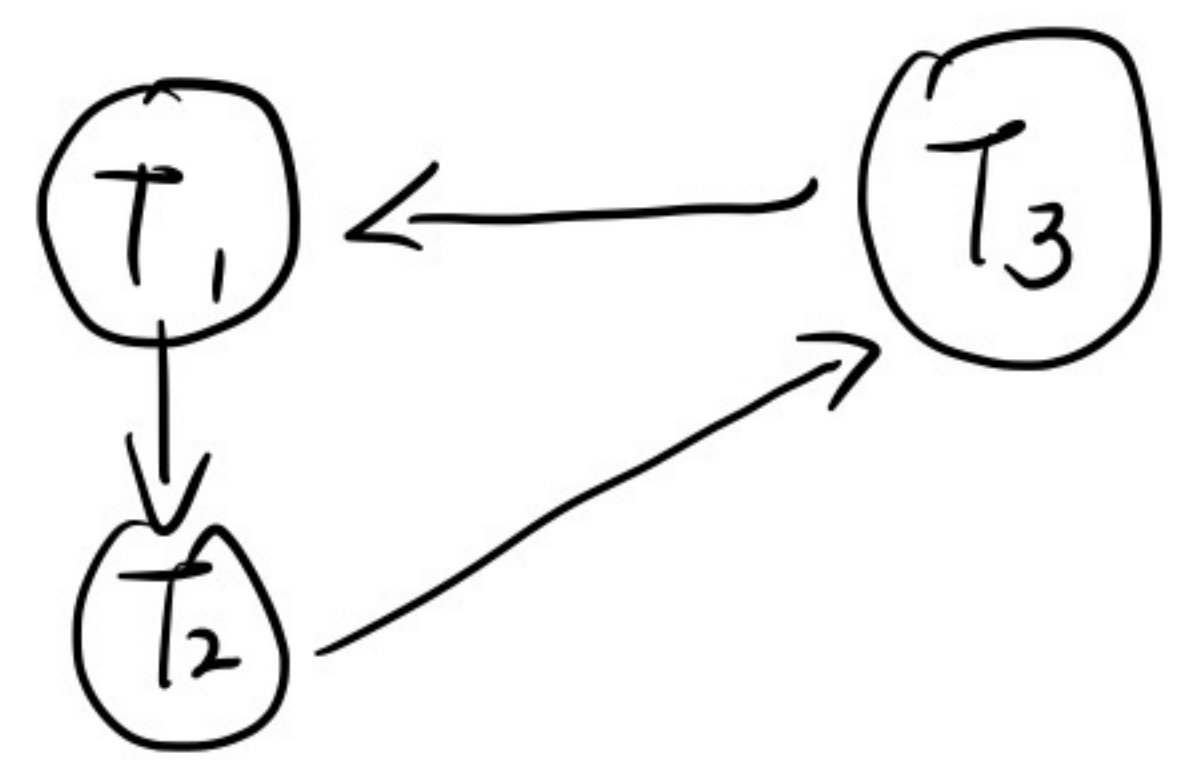
∴ Not possible to obtain schedule, T<sub>1</sub> will try acquiring SL on w in shrinking phase.

(2)

S2: r1(x) r1(y) r2(z) r3(z) r3(x) w3(x) w2(w) w2(z) r1(w) w1(y)

a)

Conflict Graph:



∴ NOT serializable



b) Not possible, since this schedule is not serializable, there will always a cycle prevent valid schedule.

Q 3

LOG:	LSN	Entry
	100	T1 update P2
	101	T2 update P1
	102	T2 commit
	103	T3 update P1
	104	T3 update P3
	105	T1 update P4
	106	T3 commit

Data Page	LSN of Last recorded log entry
P1	101
P2	100
P3	104
P4	-

a)

P<sub>2</sub> 100

P<sub>1</sub> ~~101~~ 103

P<sub>3</sub> 104

P<sub>4</sub> 105 →

First, redo : 103, 105

Then, undo: 100, 105

b) No Force, since T<sub>3</sub> is partially written to data page after commit, which won't happen if force used

c) STEAL used, as T<sub>1</sub> is an uncommitted transaction but its changes reflected to the DATA page.