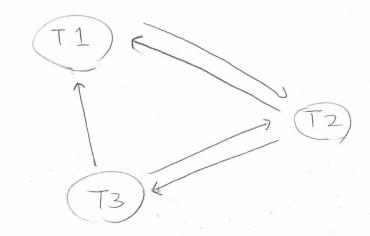
problem A

Dependency Sraph



NOT conflict-serializable because there's a cycle

E garden	agree fact of the 25	1) 111 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
TI	T 2	† 3
lock-x(D) write D	lock-x (C) write C	
waiting to set lock-S(C)		which to cot
71072		waiting to set lock-S(C)
	read A unlock-S(A) unlock-S(A) unlock-x(C)	
rend c		(ock-SCC) read C lock-+(A)
to to get	where trades at A	unlock - x(A) unlock - s(c)
lock $- \times (A)$ unite A unlock $- \times (A)$ unlock $- \times (C)$		perior promit houses
unlock - XU		

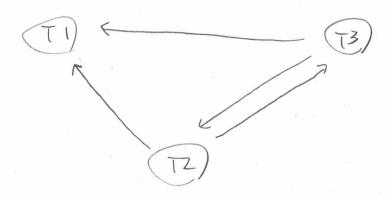
(1) complete schedule

wait- die prevents deadlock

T Z	T 3
(ock-X(c) wite (C)	
	abort and vestart later
lock-S(A) read A unlock-S(A) unlock-X(C)	
	lock-X(c) write (C) lock-S(A) read A unlock-S(A) unlock-X(C)

(i) complete schedule

4.



NOT conflict socializable

5

TI	TZ	73
(cck-x(D)		
write D		
	lock -x(c).	7
	write C	
waiting to set	Lance of the second	
(ock-5(C)		
		(OCK-X(A)
	*	write A
		waiting to 92+
Б		lock-S(C)
	waiting to set	
	(ock-S(A)	

TZ warts for T3 to release its exclusive lock on A to set shared lock on A.

73 warts for TZ to release its exclusive look on C

to get shared look on C

TI	12	73
(ock-x(D)		
uvite D		
Q	(00K- x (C)	
	write C	
. 2	about and vestant	; E 30
	later	
lock-SCO		
Vead C		(cck - X (A)
04.		write A
r.		(cck-5(c)
		read C
		unlock-Scc)
		unlock - xCA)
(ock-X(A)		
urite A	,	
unlock-x(A)		
un lock -5 (C)		
un (ock-x(D)		

ii) complete schedule

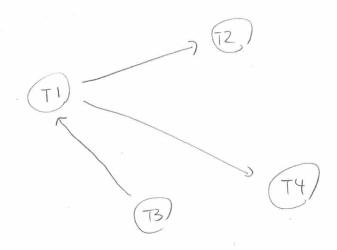
6

problem B

1	TZ	73	74
. '		w3(A)	
Y1(A)			
e	, ,	c3	
WI (B)			
C I		· .	
	Y2(B)	WZ(C)	
			Y4 (B)
,	C 2		
			C4

a No

(6)



Yes.

w3(A) C3 w2(C) r1(A) W1(B) C1 r2(B) CZ r4(B) C4

(c) Yes

 $W3(A) \rightarrow V1(A)$ and $C3 \rightarrow C1$ $W1(B) \rightarrow Y2(B)$ and $C1 \rightarrow C2$ $W1(B) \rightarrow V4(B)$ and $C1 \rightarrow C4$ un committed (i.e. W3(A))

Yes, we can make it cascadeless by moving c3 before VI(A).