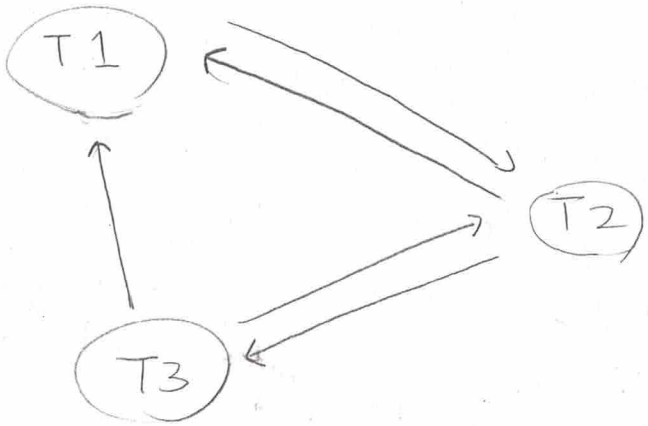
CS 143 Hw6 Junhong Wang (5049411137)

problem A

1. Dependency graph




NOT conflict-serializable because there's a cycle

2.

| T1 | T2 | T3 |
|---|---|---|
| lock-X(D) | | |
| write D | | |
| | lock-X(C) | |
| | write C | |
| waiting to get lock-S(C) | | |
| | | waiting to get lock-S(C) |
| | lock-S(A) | |
| | read A | |
| | unlock-S(A) | |
| | unlock-X(C) | |
| lock-S(C) | | |
| read C | | |
| | | lock-S(C) |
| | | read C |
| | | lock-X(A) |
| | | write A |
| | | unlock-X(A) |
| | | unlock-S(C) |
| lock-X(A) | | |
| write A | | |
| unlock-X(A) | | |
| unlock-S(C) | | |
| unlock-X(D) | | |

(i) Complete schedule

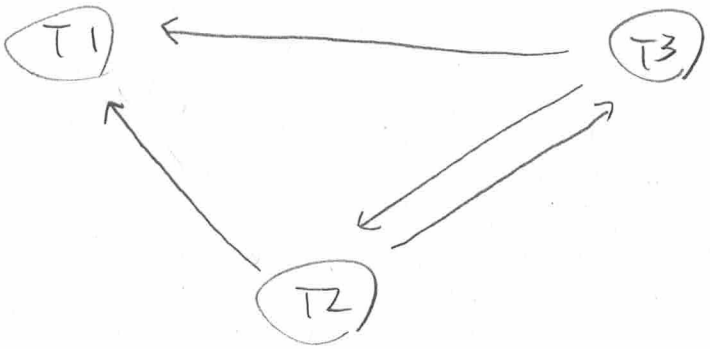
3.

wait-die prevents deadlock

| T1 | T2 | T3 |
|---|---|---|
| lock-X(D) | | |
| write D | | |
| | lock-X(C) | |
| | write C | |
| waiting to get lock-S(C) | | |
| | | abort and restart later |
| | lock-S(A) | |
| | read A | |
| | unlock-S(A) | |
| | unlock-X(C) | |
| lock-S(C) | | |
| read C | | |
| lock-X(A) | | |
| write A | | |
| unlock-X(A) | | |
| unlock-S(C) | | |

(i) complete schedule

4.




NOT conflict-serializable

5.

| T1 | T2 | T3 |
|---|---|---|
| lock-x(D) | | |
| write D | | |
| | lock-x(C) | |
| | write C | |
| waiting to set lock-S(C) | | |
| | | lock-x(A) |
| | | write A |
| | | waiting to set lock-S(C) |
| | waiting to set lock-S(A) | |

T2 waits for T3 to release its exclusive lock on A to set shared lock on A.

T3 waits for T2 to release its exclusive lock on C to set shared lock on C

cycle ! ⇒ (ii) deadlock

6

| T1 | T2 | T3 |
|---|---|---|
| lock-x(D) | | |
| write D | | |
| | lock-x(C) | |
| | write C | |
| | abort and restart later | |
| lock-S(C) | | |
| read C | | |
| | | lock-x(A) |
| | | write A |
| | | lock-S(C) |
| | | read C |
| | | unlock-S(C) |
| | | unlock-x(A) |
| lock-x(A) | | |
| write A | | |
| unlock-x(A) | | |
| unlock-S(C) | | |
| unlock-x(D) | | |

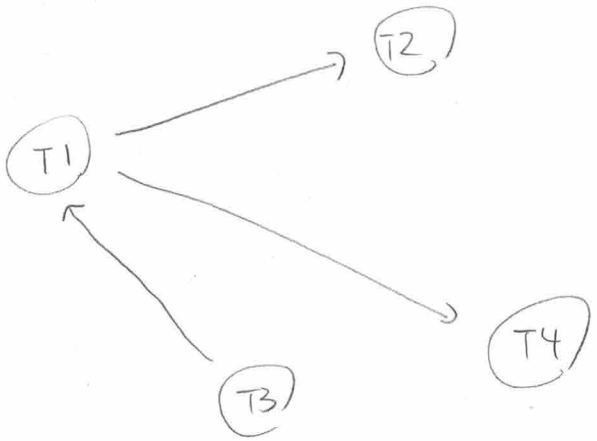
(i) Complete schedule

Problem B

| T1 | T2 | T3 | T4 |
|---|---|---|---|
| | | w3(A) | |
| r1(A) | | | |
| | | c3 | |
| w1(B) | | | |
| c1 | | | |
| | r2(B) | | |
| | | w2(C) | |
| | | | r4(B) |
| | c2 | | |
| | | | c4 |

(a) No

(b)




Yes.

w3(A) c3 w2(C) r1(A) w1(B) c1 r2(B) c2 r4(B) c4

(c) Yes

$w3(A) \rightarrow r1(A)$ and $C3 \rightarrow C1$

$w1(B) \rightarrow r2(B)$ and $C1 \rightarrow C2$

$w1(B) \rightarrow r4(B)$ and $C1 \rightarrow C4$

(d) No, because r1(A) is reading data uncommitted (i.e. w3(A))

Yes, we can make it cascadeless by moving c3 before r1(A).