Name: Nisha Patel

CSCI 410

Assignment 5

1.

T1, T2, and T3 they all start will the balance of $400:

T1:

Purchase $150

Previous Bal: $400

Total: $-550

T2:

Payment :$-400

Previous Bal: $400

Total : $0

T3:

Return $150

Previous Bal : $-400

Total: $-250

Final Balance = $250

Final balance will be of $250 because there is no concurrency control therefore it wipes out whatever result we got from T1 and T2. The final result will be from T3.

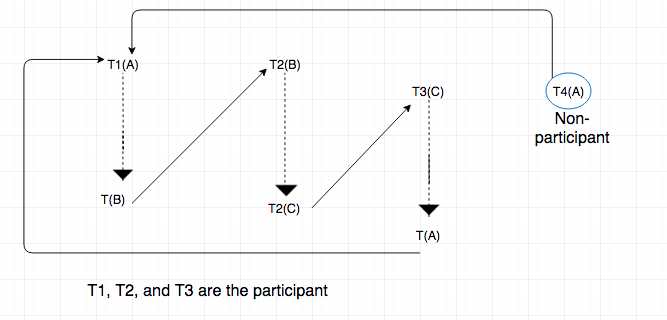
2.

|  |  |  |  |
| --- | --- | --- | --- |
| **T1** | **T2** | **T3** | **T4** |
| R(A) //S(A) acquires a shared lock |  |  |  |
|  |  | R(C) //S(C) acquires a shared lock |  |
|  | W(B) //X(B) acquires an exclusive lock. |  |  |
|  |  |  | W(A) // waiting to acquire a lock, it does not have any lock right now. Not in a deadlock state, it’s waiting… |
|  | W(C) // X(C) waits for T3 to release the lock in order to apply the exclusive lock -deadlock |  |  |
| R(B) // S(B) waits for T2 to release the lock in order to apply the shared lock-deadlock |  |  |  |
|  |  | W(A) //X(A) waits for T1 to release the lock in order to apply the exclusive lock -deadlock. |  |

T1, T2, and T3 are in a deadlock state because they already have a lock and they are requesting for more locks. In two phase locking, first they acquire all required locks to complete the process and then release it, which is why we run into a deadlock state.

However T4 is not in a deadlock state because it is waiting to acquire the lock.

Below is the graph that represents the deadlock state:



Cycles:

