Exercise 6 Solution

1. If we use the following comments to lock and unlock access to objects then: Which transactions below are in deadlock if they start around the same time?

T1	T2	T3	T4
Begin	Begin	Begin	Begin
LOCK(C)	LOCK(A)	LOCK(C)	LOCK(B)
Write C	Write A	Write C	Write B
UNLOCK(C)	LOCK(B)	UNLOCK(C)	LOCK(A)
End	Write(B)	End	Write A
	UNLOCK(A)		UNLOCK(A)
	UNLOCK(B)		UNLOCK(B)
	End		End

Solution:

T2 and T4 are in a deadlock as each of them will wait for the other to release a lock while holding a lock that the other needs to acquire to complete.

2. What are the dependencies in the following history (a sequence of tuples in the form (Ti, Oi, Tj))? Draw the dependency graph mapping to this dependency set as well.

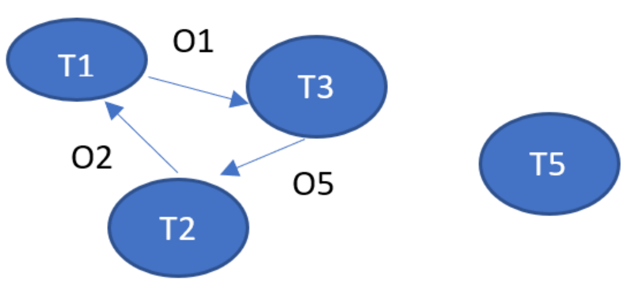
$$H =< (T1, R, O1), (T3, W, O5), (T3, W, O1), (T2, R, O5), (T2, W, O2), (T5, R, O4), (T1, R, O2), (T5, R, O3) >$$

Solution:

Dependencies are given as:

$$DEP(H) =< T1, O1, T3 >, < T3, O5, T2 >, < T2, O2, T1 >$$

We can also build the following dependency graph based on this like the following:



3. Given the solution above for question 2, can we say the history is equal to a serial history. If yes, show one such history. If not, show that there is a wormhole.

Solution:

There exists a cycle in the dependency graph, i.e., there are wormhole transactions (e.g., $T1$ is before and after of $T3$ at the same time). Therefore finding equivalent serial execution is not possible.

4. Repeat questions 2 and 3 with history:

$$H =< (T3, W, O5), (T3, W, O1), (T2, R, O5), (T2, W, O2), (T5, R, O4), (T1, R, O1), (T1, R, O2), (T5, R, O3) >$$

Solution:

Dependencies are given as:

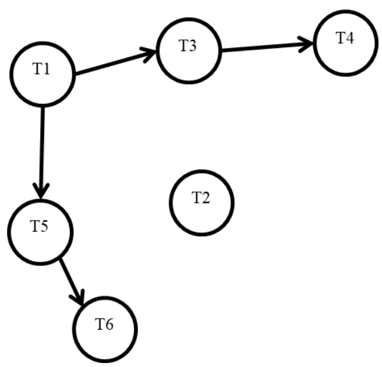
$$DEP(H) = < T3, O5, T2 >, < T3, O1, T1 >, < T2, O2, T1 >$$

This basically reverses the direction of dependency in the previous graph for O1. Thus there is no more cycles in the dependency graph, and hence we are equal to for example the following serial history:

$$H =< (T3, W, O5), (T3, W, O1), (T2, R, O5), (T2, W, O2), (T1, R, O1), (T1, R, O2), (T5, R, O4), (T5, R, O3) >$$

where $T3$ runs before $T2$ which runs before $T1$ and that is before $T5$. The DEP of this is the same as the DEP of the history given in the question.

5. Given the graph below for six transactions we can see the dependency between the transactions. If we add a link from T6 to T3, the graph becomes more complicated. Please in that case list After(T6) and Before(T6). Does this link cause a wormhole to appear, in otherwise a set of dependencies where there were no wormholes in the original version of the graph? Instead, if we have a link added from T4 to T1, please list After(T4) and Before(T4). Does this link cause a wormhole? Which one of the two modifications represents dependencies for a history that we can call isolated?



Solution:

Please review the wormhole theorem first. Then: If we add a link from T6 to T3 then After($T6$) = $T3, T4$ and Before ($T6$) = $T1, T5$. As After(T6) does not have any transactions in its intersection with Before(T6), by adding that link we did not create a cycle or basically, there is no wormhole transaction created. If a cycle was formed then T6 would have been a part of it and would create a wormhole transaction. As the previous state of the graph did not have a wormhole and this new version does not create a wormhole transaction as well, then we can say this modification's dependencies come from an isolated history. On the contrary with the second modification: with a link to T1 from T4, After($T4$) = $T1, T3$ and Before ($T4$) = $T1, T3$. The two sets overlap and thus their intersection is not an empty set and transaction T1 becomes a wormhole for example. Thus, the history that lead to that set of dependencies cannot be an isolated history.

6. Given the operations for a transaction T1 below, please list the lines that this transaction is executing that cannot happen with two-phase locking. Briefly explain.

```
1   Slock(A)
2   Read(A)
3   Unlock(A)
4   Slock(B)
5   Read(B)
6   Unlock(B)
7   Xlock(C)
8   Write(C)
9   Unlock(C)
10  Xlock(A)
11  Write(A)
12  Unlock(A)
```

Solution:

Lines 4, 7, and 10 cannot happen with two-phase locking because the first unlock operation is in Line 3. There cannot be a lock operation after the start of the shrinking phase, i.e., the transaction cannot get further locks after the first unlock operation.