**Deadlock Detection –**

When a transaction waits indefinitely to obtain a lock, the database management system should detect whether the transaction is involved in a deadlock or not.

**Wait-for graph**

Wait-for-graph is one of the methods for detecting the deadlock situation. In this method a graph is drawn based on the transaction and their lock on the resource.

If the graph created has a closed loop or a cycle, then there is a deadlock

Algorithm To draw wait-for graph

1. Create a node T in the graph for each participating transaction in the schedule.
2. Whenever a transaction Ti is waiting to lock an item X that is currently locked by a transaction Tj, it creates a directed edge Ti to Tj
3. When Tj releases the lock(s) on the items that Ti was waiting for, the directed edge is dropped from the waiting-for graph.
4. If the wait-for graph has a cycle then the system is in deadlock state
5. We can find out list of transactions involved in deadlock using above created by graph by examining nodes involved in cycle present in graph graph

Conflict pairs of lock:

**Examples on wait-for Graph**

1. The following is a list of events in an interleaved execution of set of transactions T1, T2 with two phase locking protocol. Time Transaction Code

T1 Lock (B, S)

T2 Lock (A, X)

T1 Lock (A, X)

T2 Lock (B, X)

Check if there is a deadlock using wait-for graph. If yes, which transactions are

involved in deadlock?

**Creating wait-for graph:**

1. Make two nodes corresponding to Transaction T1 and T2.
2. T1 make request to acquire shared lock on B ,which is not locked by any transaction till ,request is fulfilled
3. T2 make request to acquire Exclusive lock on A ,which is not locked by any transaction till ,request is fulfilled
4. T1 make request to acquire Exclusive lock on B ,which is locked by T1 using exclusive lock ,request is not completed,T1 has to wait for T2 to release lock .Hence there should be edge from T1 to T2

1. T2 make request to acquire Exclusive lock on B ,which is locked by T1 using shared lock ,request is not completed,T2 has to wait for T2 to release lock .Hence there should be edge from T2 to T1

1. Since the graph is cyclic(as there is cycle present in graph), we can conclude that system in deadlock.
2. As node T1 and T2 are involved in cycle ,Transactions involved in deadlock are T1 and T2
3. The following is a list of events in an interleaved execution of set of transactions T1, T2, T3, T4 with two phase locking protocol. Time Transaction Code

t1 T1 Lock (B, X)

t2 T2 Lock (A, X)

t3 T3 Lock (C, S)

t4 T4 Lock (B, X)

t5 T1 Lock (D, S)

t6 T2 Lock (C, X)

t7 T3 Lock (A, X)

t8 T4 Lock (C, S)

Check if there is a deadlock using wait-for graph. If yes, which transactions are

involved in deadlock?

1. The following is a list of events in an interleaved execution of set of transactions T1, T2, T3, T4 with two phase locking protocol. Time Transaction Code.

Check if there is a deadlock using wait-for graph. If yes, which transactions are involved in deadlock?

t1 T1 Lock (A, X)

t2 T2 Lock (B, S)

t3 T3 Lock (A, S)

t4 T4 Lock (B, S)

t5 T1 Lock (B, X)

t6 T2 Lock (C, X)

t7 T3 Lock (D, S)

t8 T4 Lock (D, X)

T2 COMMIT

As there is cycle present in graph system is in deadlock. Transactions involved in deadlock are T1,T3 and T4