

Tutorial Week 11: Transactions and Concurrency Control Continued.

Notes

Exercises

46. For each of the following input schedules, will deadlocking occur if 2PL is used? Provide the locking schedule. Note that $R(x, T_1)$, $W(x, 5, T_2)$ represents a read in transaction 1 on object x and writing 5 into the object x in transaction 2 respectively.

- (a) $R(x, T_1), W(y, 9, T_2), W(x, 4, T_2), W(y, 5, T_1)$
- (b) $W(y, 9, T_2), R(x, T_1), W(x, 8, T_1), R(y, T_2)$
- (c) $R(x, T_1), W(y, 0, T_2), R(x, T_2), W(x, 8, T_1)$

47. Consider a relaxation of two-phase locks in which read-only transactions can release read locks early.

- (a) Would a read-only transaction have consistent retrievals?
- (b) Would the objects become inconsistent?

48. Under 2PL protocol it is possible for a transaction to "starve" in the following sense: A transaction gets involved in a deadlock, is chosen as the victim and aborted. After a restart, it again gets involved in a deadlock, is chosen as the victim and aborted, and so on. Provide a concrete example for such a situation, and describe how 2PL could be extended in order to avoid starvation of transactions.

49. Can the scenario described in 48 occur when timestamp ordering is used? Provide a concrete example or show why not.

50. Conservative or static 2PL (C2PL) is a variant of 2PL under which each transaction sets all locks that it needs in the beginning, before executing its first read or write steps. This is also known as preclaiming all necessary locks up front.

- (a) Are deadlocks possible under this variant?
- (b) If the read and write sets (all read and write operations) are not known by the scheduler ahead of time, is C2PL possible? Explain.

