Ex 8 and 9, Data Management

Matteo Marini

16/3/2021

Ex 8

Every schedule following the strict 2PL protocol is strict.

Given a legal schedule $S \in S2PL$ composed of n transactions $(t_1, ..., t_n)$ a transaction j can only read a value a written by a transaction t_i (or write on t_i) if transaction t_i has unlocked value a (because of the legal schedule property). Since $S \in S2PL$, the unlock of the resource a (which was written by t_i) is performed after the commit, or the rollback, of transaction t_i (because of the definition of strict two-phase locking). This implies that transaction t_j can only read a value a written by a transaction t_i (or write on t_i) if t_i has committed (which is the definition of strict schedules).

Every schedule that is strict and follows the 2PL protocol also follows the strict 2PL protocol

This statement is false, the following schedule:

$$xl_1(a), l_1(b), w_1(a), r_1(b), u_1(a), u_1(b), c_1, l_2(a), r_2(a), u_2(a)$$

is strict and follows the 2PL protocol, but does not follows the strict 2PL protocol. This happen because nothing of the first two properties implies that the unlock of the exclusive locks have to be done after the commit or the rollback.

Ex 9

Every schedule following the strong strict 2PL protocol is rigorous.

The proof is similar to the one of Ex 8 part 1. The only difference is that this time we have strong strict 2PL that implies that all the unlocking operation (not only those related to exclusive locks) are done after the commit. This implies that every further operation that requires to lock that resource will come after the unlock and thus after the commit, or the rollback, of the transaction from which it is reading or writing.

Every schedule that is rigorous and follows the 2PL protocol also follows the strong strict 2PL protocol

Again, the proof is similar to the one of Ex 8. The schedule given in Ex 8 part 2 is rigorous and follows the 2PL protocol, but it does not follow the strong strict 2PL protocol.