Exercise 12

Date of issue: 23rd May 2017 Deadline: 30th May 2017

1 Properties of Schedules

1. Consider schedule S_1:
   $S_1 = r_3(A), r_2(B), w_2(B), r_1(B), w_3(C), c_3, r_2(D), c_2, r_1(D), c_1$
   (a) Is S_1 cascadeless? Explain briefly.
   (b) Is S_1 recoverable? Explain briefly.
   (c) Draw the precedence graph of schedule S_1.
   (d) Is S_1 conflict-serializable? Explain briefly.
   (e) If S_1 is conflict-serializable, write down all equivalent serial schedules.
   (f) If S_1 is not conflict-serializable, make the schedule conflict-serializable. You may perform exactly one exchange, i.e. you can swap exactly two operations. The order of operations within a transaction must be preserved. Show that your modified schedule is conflict-serializable.
2. Consider schedule S_2:
   $S_2 = w_3(D), r_1(A), w_2(D), w_2(A), w_3(A), c_3, w_1(C), r_2(D), c_2, c_1$
   (a) Is S_2 cascadeless? Explain briefly.
   (b) Is S_2 recoverable? Explain briefly.
   (c) Draw the precedence graph of schedule S_2.
   (d) Is S_2 conflict-serializable? Explain briefly.
   (e) If S_2 is conflict-serializable, write down all equivalent serial schedules.
   (f) If S_2 is not conflict-serializable, make the schedule conflict-serializable. You may perform exactly one exchange, i.e. you can swap exactly two operations. The order of operations within a transaction must be preserved. Show that your modified schedule is conflict-serializable.

2 Two-Phase Locking

1. Consider schedule S:
   $S = r_1(A), r_2(D), r_1(C), w_1(C), w_2(A), c_2, r_1(B), c_1$
   (a) Extend schedule S with lock and unlock statements such that it adheres to the S2PL protocol and is deadlock-free. If this is impossible, explain why.
   (b) Extend schedule S with lock and unlock statements such that it adheres to the SS2PL protocol and is deadlock-free. If this is impossible, explain why.
2. Prove or disprove the following statement: The Two-Phase Locking (2PL) protocol permits all possible conflict-serializable schedules.