Exercise 11 “Implementation of Databases”
Due until 23 Jan, 2008 (before exercise)  WS 07/08

Exercise 11.1 [Recoverability] :

1. Prove the following:
   (a) Every schedule belonging to avoid cascading abort (ACA) is also recoverable (RC). (2 pt.)
   (b) Each schedule that is strict is also in ACA. (2 pt.)

2. Test for every following schedule its membership in RC, ACA, and ST. (6 pt.)

\[
\begin{align*}
  s_1 &= r_3(y)r_1(x)w_1(x)r_2(x)c_1w_3(y)w_2(x)c_3c_2 \\
  s_2 &= r_2(x)r_3(y)w_2(x)c_2r_3(x)w_3(x)c_3r_1(x)w_1(x)c_1 \\
  s_3 &= r_1(x)w_1(x)r_3(x)w_3(x)c_3r_2(x)w_2(x)c_2a_1
\end{align*}
\]

Exercise 11.2 [2PL] :

1. Prove that schedules produced by 2PL are conflict serializable. (2 pt.)

2. For each of the following input schedules write down the corresponding schedule indicating the necessary locking (rl/wl) and unlocking (ru/wu) operations. Write down the resulting output schedules for 2PL (transactions must unlock resources as soon as possible) and S2PL. (4 pt.)

\[
\begin{align*}
  s_1 &= w_1(x)r_2(y)r_1(x)c_1r_2(x)w_2(y)c_2 \\
  s_2 &= w_1(x)r_2(y)r_1(x)c_1r_2(x)w_2(y)a_2
\end{align*}
\]

Exercise 11.3 [Deadlock Handling] :

Given the following schedule
\[
s = r_1(x)r_2(x)w_3(x)w_4(x)w_1(x)c_1w_2(x)c_2c_3c_4
\]

1. Draw the wait-for graph for the schedule. (1 pt.)

2. Write down the corresponding schedules produced by 2PL and S2PL, using a deadlock detection (blocking) strategy. In case of a deadlock the transaction with the lowest index is aborted. Once aborted, transactions are restarted anew at the end of the original schedule (abort-restart). (3 pt.)