Exercise 3 "Implementation of Databases"
Due until 12 Nov, 2008 (before exercises)  

3.1  Recoverability  (10 pt.)

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

2. Test for every following schedule its membership in RC, ACA, and ST
   a) $s_1 = r_3(y) r_1(x) w_1(x) r_2(x) c_1 w_3(y) w_2(x) c_3 c_2$
   b) $s_2 = r_2(x) r_3(y) w_2(x) c_2 r_3(x) w_3(x) c_3 r_1(x) w_1(x) c_1$
   c) $s_3 = r_1(x) w_1(x) r_3(x) w_3(x) c_3 r_2(x) w_2(x) c_2 a_1$

3.2  2PL / S2PL / Deadlock Handling  (10 pt.)

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

2. Given the following schedules:
   $s_1 = w_1(x) r_2(y) r_1(x) c_1 r_2(x) w_2(y) c_2$
   $s_2 = w_1(x) r_2(y) r_1(x) c_1 r_2(x) w_2(y) a_2$
   $s_3 = r_1(x) r_2(x) w_3(x) w_4(x) w_1(x) c_1 w_2(x) c_2 c_3 c_4$
   a) Compute the conflict graphs of the schedules above.
   b) For each input schedule 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. 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)