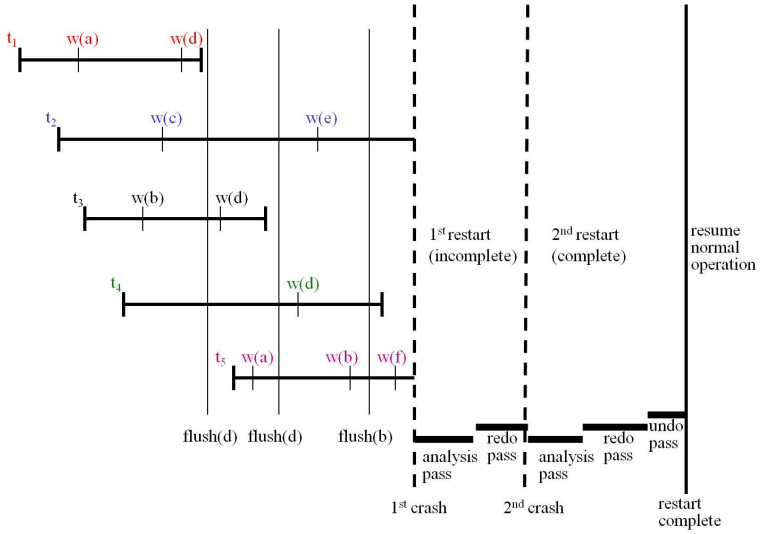
Universität Freiburg
Institut für Informatik
Prof. Dr. G. Lausen

Georges-Köhler Allee, Geb. 51
D-79110 Freiburg
lausen@informatik.uni-freiburg.de

Exercises
Distributed Systemes: Part 2
Summerterm 2012
13.7.2012

6. Exercise sheet: Recovery and Petri nets

Exercise 1
Consider the following diagram, which demonstrates normal processing, crash, repeated crash and restart of some transactions.



(1) Assume only full writes. For each action found in the figure fill the corresponding row of the table, as it is sketched:

sequence number:action	cached database PageNo:SeqNo	stable database PageNo:SeqNo	log entry added to log buffer LogSeqNo:action	log entry added to stable log LogSeqNo's
1:begin(t_1)				
2:begin(t_1)				
3:write(a,t_1)	a:3		3:write(a,t_1)	
⋮	⋮	⋮	⋮	⋮

(2) Now consider physiological writes and discuss the changes to your solution (1).

Exercise 2
Consider a distributed system with nodes a, b, c, d and assume data is replicated at each node.

(1) When applying the basic quorum protocol, what do you get for N_R and N_W?

(2) When applying quorum consensus, how many different solutions for N_R and N_W are possible? Discuss advantages and drawbacks for each possible selection.

(3) Now assume that node a is considerable more reliable than the others. Develop a generalization of the protocol, which is able to take this into account to achieve higher performance and reliability.

(4) Discuss how the different versions of the quorum protocol react on network partitions.

Exercise 3
Model a traffic light by a Petri-Net.

(1) You can use any number of places, however only multiplicity 1 is allowd.

(2) Now only 3 places (one for each color) may be used, but there are no restrictions on the multiplicities.

Exercise 4
Prove or give a counterexample: $m[q\rangle m' \Leftarrow m' = m + \Delta q$.

Exercise 5

(1) Model the following Handshaking protocol by a Petri-Net:
Two processes P1 and P2 mutually exchange messages. P1 is the sender and P2 the receiver. P1 starts in state *Ready-to-Send*. When it has sent a message to P2, it moves into the state *Ready-to-Receive* and waits for an acknowledgement ACK sent by P2. Once the acknowledgement has been arrived, P1 can send more messages. P2 starts in state *Waiting-for-Messages*. If it receives a message, it confirms by sending an acknowledgement ACK to P1 and waits for more messages.

(2) Give the reachability tree.