Exercises for *Transaction Systems*, summer term 2017
Linnea Passing (linnea.passing@tum.de)
http://www-db.in.tum.de/teaching/ss17/transactions/

Sheet No. 9

Info

- Due date: Friday, July 21, 3pm.
- Please send your solution via e-mail, and prefix the subject with [transactions].
- Please include your Matrikelnummer and your name.

Exercise 1 (6 points) Consider the following execution of operations on an initially empty queue $q$, where $a$, $b$, and $c$ are added:

\[\text{enq}_1(q, a) \ \text{enq}_2(q, b) \ \text{deq}_3(q) \ \text{enq}_1(q, c) \ \text{deq}_3(q)\]

For each of the following assumptions, state whether the execution is serializable, and give a short explanation (i.e., one sentence):

(a) general commutativity
(b) return value commutativity for queues
(c) return value commutativity for semi-queues

Exercise 2 (12 points) Apply the following hybrid protocols to the schedule found on slide 8 of chapter 7 in a selective way, such that only two levels are involved:

- forward-oriented optimistic concurrency control (FOCC) at the page level and strong 2PL at the record level
- FOCC at the page level, strong 2PL at the query level
- ROMV at the page level, strong 2PL at the record level
- ROMV at the page level, strong 2PL at the query level

Exercise 3 (optional, 6 bonus points) Show for the IDM model of transactions:

(a) Final state serializability is not monotone.
(b) Conflict serializability is monotone.

Exercise 4 (optional, 4 bonus points) Discuss how predicate locking can be extended to disjunctive conditions such as queries of the form

\[\text{select name from employees where position='Manager' or department='Research'};\]

Also discuss how join queries such as

\[\text{select e.name, e.department from employees e, department d where e.position='Manager' and d.city='Toronto' and e.department=d.department};\]

could be (conservatively) handled by predicate locking.
**Exercise 5 (optional, 10 bonus points)** Consider the following B+-tree index on the attribute `accountnumber` of an `accounts` table. Assume that all tree nodes have a space capacity for holding up to four entries. Write down all locking and unlocking operations necessary for the execution of the following transaction, assuming incremental key range locking at the access layer and lock coupling at the page layer.

```
begin transaction;
select count(*) from accounts where accountnumber between 11 and 25;
insert into accounts (accountnumber, ...) values (27, ...);
commit transaction;
```