

## Assignment #5: Recovery & Serializability

Release: April 11, 2017

Due: 11:59 PM, April 23, 2017.

---

Answer the following questions [for a total of 100 points]:

1. [15 points] Develop appropriate data structures and a recovery algorithm based on before images for rolling back aborted transactions whose execution is not strict and yet the lost update problem is eliminated. Assume no system or media failures.

2. [50 points] Consider the following Histories:

$$H_1 = W_1(x)W_1(y)W_3(x)R_2(y)R_1(z)R_3(x)R_2(z)R_2(x)$$

$$H_2 = R_2(y)R_1(z)W_1(y)W_3(x)R_2(x)W_1(x)R_2(z)R_3(x)$$

$$H_3 = R_2(z)R_1(z)W_1(x)W_1(y)R_2(x)R_3(x)R_2(y)W_3(x)$$

$$H_4 = W_1(y)W_1(x)W_3(x)R_3(x)R_2(x)R_2(z)R_2(y)R_1(z)$$

- (a) Which of them are conflict equivalent?
  - (b) Which of them are conflict serializable? Give the equivalent serial history.
  - (c) State which scheduler (2PL or TO or SSG) would have generated each history.
  - (d) For each history insert Commit operations to make them ACA, if possible.
  - (e) For each history insert Commit operations to make them Strict, if possible.
3. [20 points] Consider the execution of three transactions  $T_1$ ,  $T_2$  and  $T_3$ .

$T_1$ :	$T_2$ :	$T_3$ :
read(z)	write(x)	write(x)
read(y)	read(z)	read(z)
write(x)	write(y)	read(y)

- (a) Give an example of histories of  $T_1$ ,  $T_2$  and  $T_3$  that are not serial but serializable in the order of  $T_3, T_1, T_2$ .
- (b) Give another example of histories of  $T_1$ ,  $T_2$  and  $T_3$  that are not serial but serializable in any order of  $T_1, T_2$  and  $T_3$ .

4. [15 points] Specify the compatibility table for the following set of database operations defined on a stack-queue:
  - Pop(s): Removes and returns the top element on the stack-queue  $s$ .
  - Push(s,X): Pushes  $X$  on top of the stack-queue  $s$  and returns OK.
  - Top(s): Returns the top element on the stack-queue  $s$ .
  - Enqueue(s,X): Enqueues  $X$  at the bottom of the stack-queue  $s$  and returns OK.

## What to submit

- You are required to submit **exactly one** PDF file under your **pitt\_user\_name** (e.g., abc95.pdf). In addition to providing the answers, you are expected to: **include your name and pitt user name at the top of the PDF file.**
- Submit your assignments through the Web-based submission interface (at the class web page <http://db.cs.pitt.edu/courses/cs2550/17-2/>). **It is your responsibility to make sure the assignment was properly submitted.**
- Submit your assignment by the due date (**11:59 PM, April 23, 2017**). There is **no late submission.**

## Academic Honesty

The work in this assignment is to be done *independently*. Discussions with other students on the assignment should be limited to understanding the statement of the problem. Cheating in any way, including giving your work to someone else will result in an F for the course and a report to the appropriate University authority.