CSE 414 Homework 5: Transactions

Objectives: To evaluate the properties of transaction schedules, and write transaction statements that would be used in an application.

Assignment tools:

- Any word processing or drawing tools you prefer (e.g., Google docs to pdf, Word, draw.io).

Assigned date: Tuesday, November 10th

Due date: Wednesday, November 18th, 11pm

What to turn in: See submission instructions at the bottom.

Assignment Details

Part 1: Schedules With Anomalies (30 points)

Consider a database with objects X, Y, and Z and assume that there are two transactions T1 and T2 that attempt the following operations.

T1: R(X), R(Y), W(X)
T2: R(X), R(Y), W(Y), R(X), R(Y), W(X), R(Z), W(Z)

Write example schedules (i.e. interleavings of the operations) for the transactions T1 and T2 to illustrate each of the anomalies below:

1. Your schedule should contain a write-read conflict that causes one of the transactions to perform a dirty read.
2. Your schedule should contain a read-write conflict that causes one of the transactions to encounter an unrepeatable read.
3. Your schedule should contain a write-write conflict that causes a lost update.

In each case, your schedules may contain additional conflicts, but should contain at least one conflict of the type indicated. (If you wish, you may write a single schedule, which illustrates all three conflicts!) In each case, indicate the conflict of the type you are illustrating by pointing out which operations have caused it.

Part 2: Conflict Serializability (20 points)

Consider the following three transactions and schedule (time goes from top to bottom). Is this schedule conflict-serializable? Show why or why not.

T1	T2	T3
R(A)		
W(A)		
		R(A)
		W(A)
	R(A)	
R(B)		
		R(B)
W(B)		
		W(B)
	R(B)	
	commit	
commit		
		commit

Part 3: Two-Phase Locking (20 points)

A) Now modify the above schedule by adding locks, which may block some transactions from doing their operations until the lock is released. You'll need to **rewrite** the above schedule in a table form.

Use two-phase locking in your modified schedule to ensure a conflict-serializable schedule for the transactions above.

Use the notation L(A) to indicate that the transaction acquires the lock on element A and U(A) to indicate that the transaction releases its lock on.

B) If 2PL ensures conflict-serializability, why do we need strict 2PL? Explain briefly.

Submission Instructions

The files you will need to submit to Gradescope

- Part1.pdf
- Part2.pdf
- Part3.pdf

Points may be deducted for incorrect file names.

Submit your answers to Gradescope.