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).

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

Assignment Details

Part 1: Schedules and Anomalies

Consider a database with objects (also called “elements”), X, Y, and Z and assume that there are two transactions T1 and T2 that attempt the following operations.

T1: $R_1(X)$, $R_1(Y)$, $W_1(X)$

T2: $R_2(X)$, $R_2(Y)$, $W_2(Y)$, $R_2(X)$, $R_2(Y)$, $W_2(X)$, $R_2(Z)$, $W_2(Z)$

A) [20 points] Write an example schedule that interleaves operations between T1 and T2, that is NOT conflict serializable.

B) [30 points] Consider the three schedules below. Here X and Y are database elements, while u,v,s,t, are local variables of the transactions. For each of the three schedules indicate whether the schedule is serializable. Notice that the question is whether the schedule is *serializable*; the question is not whether the schedule is *conflict serializable*.

Schedule 1:

T1	T2
$R_1(X,u)$	
	$R_2(X,s)$
	$R_2(Y,t)$
	$W_2(X,\min(s,t))$
	$W_2(Y,\max(s,t))$
$R_1(Y,v)$	
$W_1(X,\min(u,v))$	
$W_1(Y,\max(u,v))$	

Schedule 2:

T1	T2
$R_1(X,u)$	
$R_1(Y,v)$	
	$R_2(X,s)$
	$R_2(Y,t)$
	$W_2(X,\min(s,t))$
	$W_2(Y,\max(s,t))$
$W_1(X,\min(u,v))$	
$W_1(Y,\max(u,v))$	

Schedule 3:

T1	T2

$R_1(X,u)$	
$R_1(Y,v)$	
$W_1(X,\min(u,v))$	
	$R_2(X,s)$
	$R_2(Y,t)$
	$W_2(X,\min(s,t))$
	$W_2(Y,\max(s,t))$
$W_1(Y,\max(u,v))$	

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
$R_1(A)$		
$W_1(A)$		
		$R_3(A)$
		$W_3(A)$
	$R_2(A)$	
$R_1(B)$		
		$R_3(B)$
$W_1(B)$		
		$W_3(B)$
	$R_2(B)$	
	commit_2	

commit$_1$		
		commit$_3$

Part 3: Two-Phase Locking

A) [20 points] 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. (The lecture slides show how to represent blocking in your schedules.)

Use two-phase locking (doesn't need to be "strict") 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) [10 points] 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.