CSE 444: Homework 3

Linxing Preston Jiang   Winter 2018

February 14, 2018

1 Concurrency Control with Locking

1. (a) $R_2(X), R_2(Y), \underline{W_2(Y)}, R_1(X), \underline{R_1(Y)}, W_1(X), C_1, ...$(the rest of Transaction 2)

   (b) $\underline{R_1(X)}, R_1(Y), \underline{W_1(X)}, \underline{R_2(X)}, R_2(Y), W_2(Y), C_1, ...$(the rest of Transaction 2)

   (c) $R_1(X), R_1(Y), \underline{W_1(X)}, R_2(X), R_2(Y), W_2(Y), R_2(X), R_2(Y), \underline{W_2(X)}, \underline{C_1}, ...$(the rest of Transaction 2)

2. No. $R_0(A)$ is before $R_2(A)$ so Transaction T_0 needs to precede T_2. $R_2(B)$ is before $W_0(B)$ so Transaction T_2 needs to precede T_0, forming a cycle in the precedence graph.

3. See the following schedule

Transaction T_0	Transaction T_1	Transaction T_2
$L_0(A), L_0(B)$		
$R_0(A)$		
$W_0(A), U_0(A)$		
	$L_1(A)$ – DENIED	$L_2(A), R_2(A)$
		$W_2(A)$
	$L_1(A)$ – DENIED	
$R_0(B)$		$L_2(B)$ – DENIED
$W_0(B), U_0(B)$		
		$L_2(B), R_2(B), U_2(A)$
		$W_2(B), U_2(B)$
	$L_1(A), R_1(A)$	
	$L_2(B), R_1(B)$	
	$U_1(A), U_1(B)$	

4. By only releasing all locks when the transaction is completed, we have both conflict-serializable and recoverable schedules. Also we can avoid cascading aborts.

2 Optimistic Concurrency Control

1. See below

T_1	T_2	T_3	T_4	X	Y
1	2	3	4	RT = 0 WT = 0 C = true	RT = 0 WT = 0 C = true
	$R_2(X)$			RT = 2	
$R_1(X)$				RT = 2	
	$W_2(X)$			WT = 2 C = false	
			$W_4(X)$	WT = 4	
$W_1(X)$ ABORT					
		$W_3(X)$ DELAY			
			ABORT		
	$R_2(Y)$	$W_3(X)$		WT = 3	RT = 2
	$W_2(Y)$				WT = 2 C = false
		$R_3(X)$ DELAY			
	C_2	$R_3(X)$		RT =2 WT = 3 C = true	RT = 3
		$W_3(X)$			WT = 3
		C_3			RT =3 WT = 3 C = true

2. See below

T_1	T_2	T_3	T_4	X_0	X_3	X_4
1	2	3	4			
$R_1(X)$				RT = 1		
		$R_3(X)$		RT = 3		
		$W_3(X)$			CREATE	
	$R_2(X)$			RT = 3		
			$R_4(X)$		RT = 4	
	$W_2(X)$ ABORT					
			$W_4(X)$			CREATE