CS1555 - Serialization

Zachary M. Mattis

April 13, 2019

I. History 1

$H_1 = RL_1(x), R_1(x), WL_1(x), W_1(x), RL_2(y), R_2(y), RU_2(y), WL_1(y), C_2, W_1(y), WU_1(x), WU_1(y), C_1$

	T_1	T_2
1	RL(x)	
2	R(x)	
3	WL(x)	
4	W(x)	
5		RL(y)
6		R(y)
7		RU(y)
8	WL(y)	
9		C
10	W(y)	
11	WU(x)	
12	WU(y)	
13	C	

Table 1: Concurrent History 1

	T_1	T_2
1		RL(y)
2		R(y)
3		RU(y)
4		C
5	RL(x)	
6	R(x)	
7	WL(x)	
8	W(x)	
9	WL(y)	
10	W(y)	
11	WU(x)	
12	WU(y)	
13	C	

Table 2: Serial History 1

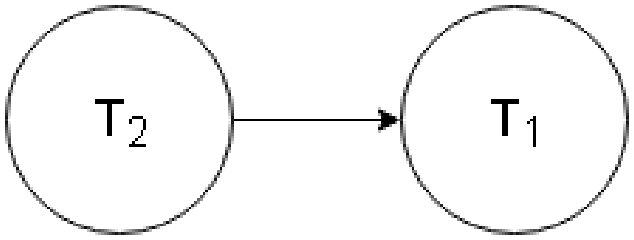

Figure 1: SG(H_1)

2PL:

History 1 above follows 2PL as each transaction adheres to the concurrency protocol by following both the *growing* and *shrinking* phases.

II. History 2

$$H_2 = WL_1(x), W_1(x), WU_1(x), RL_3(y), R_3(y), WL_2(x), W_2(x), WU_2(x), C_2, C_1, RU_3(y), C_3$$

	T_1	T_2	T_3
1	WL(x)		
2	W(x)		
3	WU(x)		
4			RL(y)
5			R(y)
6		WL(x)	
7		W(x)	
8		WU(x)	
9		C	
10	C		
11			RU(y)
12			C

Table 3: Concurrent History 2

	T_1	T_2	T_3
1	WL(x)		
2	W(x)		
3	WU(x)		
4	C		
5		WL(x)	
6		W(x)	
7		WU(x)	
8		C	
9			RL(y)
10			R(y)
11			RU(y)
12			C

Table 4: Serial History 2

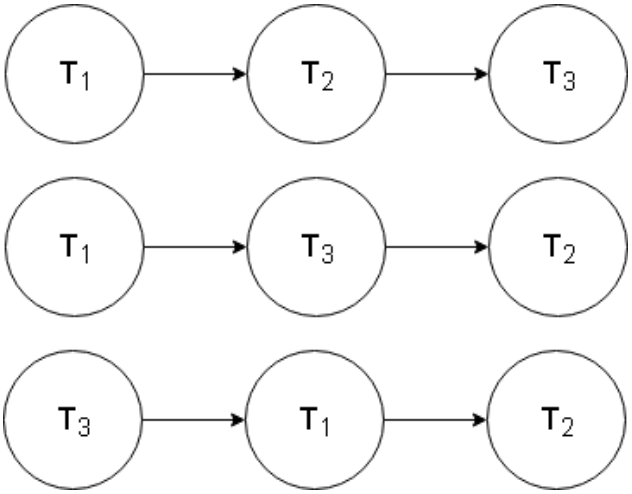

Figure 2: SG(H_2)

All 3 graphs are possible serialization graphs, with the requirement that T_2 follows T_1.

2PL:

History 2 above follows 2PL as each transaction adheres to the concurrency protocol by following both the *growing* and *shrinking* phases. However, this history does not follow strict 2PL, as it releases its write locks before the transaction is committed.