Quiz Sheet No. 3 for *Architecture and Implementation of Database Systems*
Prof. Rudolf Bayer, Ph. D.
Institut für Informatik
SS 2003

Exercises for **Chapters 4.2 – 4.7: Serializability and Synchronization**

1. Let H: $r_1(y), r_3(n), r_2(y), w_1(y), r_1(x), w_1(x), r_2(x), r_2(z), w_2(z), r_3(x), w_3(x)$ be a schedule of three transactions t_1, t_2 and t_3 that are accessing objects n, x, y and z.

   a) Write the sequence of actions for each transaction t_1, t_2 and t_3.

Answer:

t_1 = (bot, r(y), w(y), r(x), w(x), eot)
t_2 = (bot, r(y), r(x), r(z), w(z), eot)
t_3 = (bot, r(n), r(x), w(x), eot)

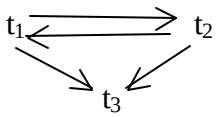
   b) Determine the conflict relation $conf_H$.

Answer:

$conf_H$ = { $(r_2(y), w_1(y))$, $(r_1(x), w_3(x))$, $(w_1(x), r_2(x))$, $(w_1(x), r_3(x))$, $(w_1(x), w_3(x))$, $(r_2(x), w_3(x))$ }

   c) Sketch the corresponding conflict graph using the "follow relation" (defined in chapter 4.3 on page11).

Answer:



   d) Is schedule H conflict serializable?

Answer:

Conflict graph is cyclic ⇒ H is not conflict serializable

2. Let t_1, t_2 and t_3 be three transactions that are defined in the following way:
   t_1 = (bot, r(u), r(y), w(y), r(z), w(z), w(u), eot)
   t_2 = (bot, r(z), r(u), r(y), w(z), eot)
   t_3 = (bot, r(z), r(y), r(u), w(u), eot)

   a) Compute $\overline{conf}$ for the set of operations in t_1, t_2 and t_3.

Answer:

$\overline{conf}$ = { $(w_1(y), r_2(y))$, $(w_1(y), r_3(y))$, $(w_1(z), r_2(z))$, $(w_1(z), w_2(z))$,
$(w_1(z), r_3(z))$, $(w_1(u), r_2(u))$, $(w_1(u), r_3(u))$, $(w_1(u), w_3(u))$,
$(w_2(z), r_1(z))$, $(w_2(z), r_3(z))$, $(w_3(u), r_1(u))$, $(w_3(u), r_2(u))$ }

   b) Consider the following schedule H1:

t_1	t_2	t_3
r(u)		
r(y)		
	r(z)	
	r(u)	
w(y)		
		r(z)
		r(y)
	r(y)	
r(z)		
w(z)		
		r(u)
		w(u)
w(u)		
	w(z)	

timeline

   Determine the relation $conf_{H1}$ for schedule H1.

Answer:

$conf_{H1}$ = { $(r_1(u), w_3(u))$, $(r_2(z), w_1(z))$, $(r_2(u), w_1(u))$, $(r_2(u), w_3(u))$,
$(w_1(y), r_2(y))$, $(w_1(y), r_3(y))$, $(r_3(z), w_1(z))$, $(r_3(z), w_2(z))$,
$(r_1(z), w_2(z))$, $(w_1(z), w_2(z))$, $(r_3(u), w_1(u))$, $(w_3(u), w_1(u))$ }

c) Consider a second schedule H2:

t_1	t_2	t_3
r(u)		
	r(z)	
r(y)		
	r(u)	
	r(y)	
w(y)		
	w(z)	
r(z)		
w(z)		
		r(z)
		r(y)
w(u)		
		r(u)
		w(u)

Show that H2 is conflict serializable.

Answer:

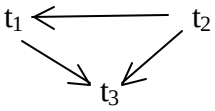
$conf_{H2}$ = { $(r_1(u), w_3(u))$, $(r_2(z), w_1(z))$, $(r_2(u), w_1(u))$, $(r_2(u), w_3(u))$,
$(r_2(y), w_1(y))$, $(w_1(y), r_3(y))$, $(w_2(z), r_1(z))$, $(w_2(z), w_1(z))$,
$(w_2(z), r_3(z))$, $(w_1(z), r_3(z))$, $(w_1(u), r_3(u))$, $(w_1(u), w_3(u))$ }

Consider serial execution of the three transactions in H2': t_2; t_1; t_3.

$conf_{H2'}$ = { $(r_2(z), w_1(z))$, $(r_2(u), w_1(u))$, $(r_2(u), w_3(u))$, $(r_2(y), w_1(y))$,
$(w_2(z), r_1(z))$, $(w_2(z), w_1(z))$, $(w_2(z), r_3(z))$, $(r_1(u), w_3(u))$,
$(w_1(y), r_3(y))$, $(w_1(z), r_3(z))$, $(w_1(u), r_3(u))$, $(w_1(u), w_3(u))$ }
= $conf_{H2}$

$\Rightarrow$ H2 is conflict serializable

conflict graph of H2 is acyclic:



d) Describe some possibilities for modifying H2 to a new schedule H3, so that H3 is also serializable.

Answer:

A variety of answers are possible, e.g.
- the order of the first 5 read operations in H2 can be chosen freely
- $w_1(y)$ and $w_2(z)$ do not affect each other, so H3: …, $w_2(z)$, $w_1(y)$, … is possible
- …

e) List the lock requests for each transaction t_1, t_2 and t_3 according to the (R,A,X) protocol.

Answer:

Lock requests at the time of transaction commit:

t_1: (t_1, u, A), (t_1, y, A), (t_1, z, A)
t_2: (t_2, z, A), (t_2, u, R), (t_2, y, R)
t_3: (t_3, z, R), (t_3, y, R), (t_3, u, A)

3. Assume optimistic concurrency control (OCC) regarding this schedule:

t_1	t_2
bot	
	bot
r(x)	
r(y)	
	r(z)
w(x)	
	w(z)
eot	
	r(x)
	eot

a) Determine the ReadSet and WriteSet of each transaction.

Answer:

	ReadSet	**WriteSet**
t_1	x, y	x
t_2	z, x	z

b) What will happen to transaction t_2 in this execution order? Why?

Answer:

$ReadSet_{t2} \cap WriteSet_{t1} = \{x\}$, i.e. $\neq \emptyset$

? t_2 will be aborted and must be repeated (although it is serializable).