Carnegie Mellon University
Department of Computer Science
15-415/615 - Database Applications
C. Faloutsos, Spring 2013
Deadline: 1:30pmon Tue 4/23/2013

Homework 8 - Concurrency - Solutions

Reminders - IMPORTANT:

- Like all homeworks, it has to be done **individually**.
- Please **typeset** your answers.
- Please submit your answers in **hard copy, in class**, 1:30pm, on Tue 4/23/2013
- As before, for ease of grading, please solve each of the questions on a **separate** page, and type
  - the **homework number** (i.e., 'HW8')
  - your **name** and
  - your **andrew ID**

  on each of the answer pages.

Reminders - FYI:

- Weight: 5% of homework grade, as announced.
- The points of this homework add up to 100.
- **Explanations**: Optional, unless explicitly asked. If you do give explanations, they will be used to your benefit, for partial credit.
- Rough time estimates: 2-4 hours.

Revision : 2013/04/23 17:23

Question 1: Serializability 1 .....[20 points]

[*** SUBMIT ON SEPARATE PAGE ***]

Consider the Schedule A given below in Table 1. R(·) and W(·) stand for 'Read' and 'Write', respectively. Ignore the lock T1:S(Y), for the moment.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
T1	**S(Y)**	R(Y)				S(X)	U(Y)													R(X)	U(X)
T2			X(X)	W(X)	U(X)																
T3														S(X)	R(X)	X(Z)	W(Z)	U(X)	U(Z)		
T4								S(Z)	R(Z)	X(Y)	W(Y)	U(Z)	U(Y)								

Table 1: Schedule A, with 4 transactions.

(a) **[2 points]** Is the schedule serial? ◯ Yes √ **No**

(b) **[5 points]** Which of the following dependency graphs (a-d) of Figure 1 corresponds to the schedule?

◯ Figure 1a √ **Figure 1b** ◯ Figure 1c ◯ Figure 1d

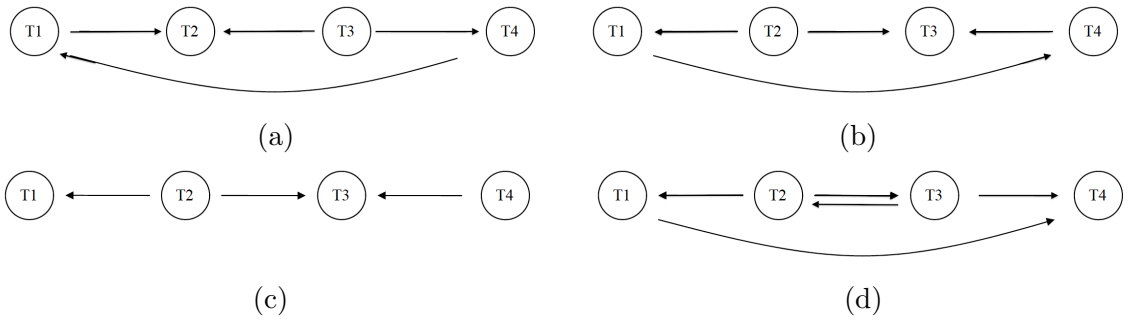

Figure 1: Potential dependency graphs for Schedule A.

(c) **[2 points]** Is Schedule A conflict serializable? √ **Yes** ◯ No

(d) **[3 points]** If not, briefly explain why. If it is conflict serializable, what is the conflict equivalent serial schedule?

Solution: Schedule A is conflict serializable, because the dependency graph does not contain any cycle. The equivalent serial schedule is T2-T1-T4-T3.

(e) **[1 point]** Is the schedule allowed by 2PL? √ **Yes** ◯ No

(f) **[7 points]** If not, briefly explain why. If yes, fill in Table 1 with the lock/unlock requests that could have happened.

- *Make sure* that the 2PL protocol is obeyed, by all.
- Use S(·) for shared lock, X(·) for exclusive lock and U(·) to unlock. We already put the lock T1:S(Y) (in bold), as an example.

Solution: See Table 1.

Grading info: 0.5 point per lock/unlock.

Question 2: Serializability 2 ..... [20 points]

[*** SUBMIT ON SEPARATE PAGE ***]

Consider Schedule B given below in Table 2.

time	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
T1		R(X)		R(Y)		W(X)					W(X)				W(X)		
T2									R(Y)								R(Y)
T3													W(Y)				

Table 2: **Schedule B**.

(a) **[2 points]** Is the schedule serial? ◯ Yes √ **No**

(b) **[5 points]** In Figure 2, which of the dependency graphs (a-d) corresponds to the schedule? ◯ Figure 2a ◯ Figure 2b ◯ Figure 2c √ **Figure 2d**

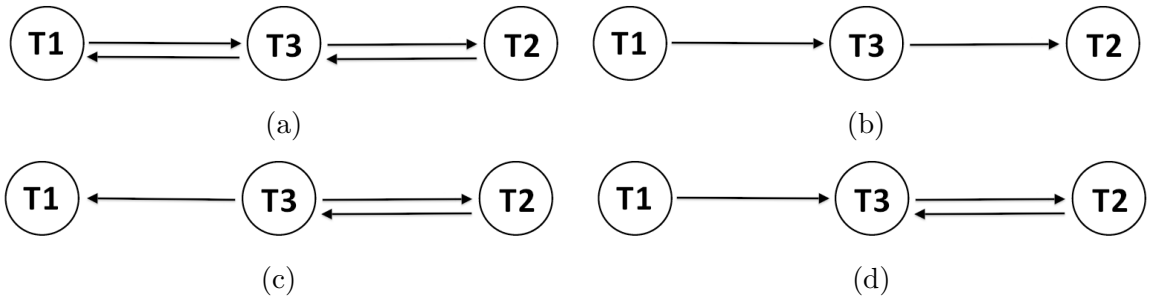

Figure 2: Potential dependency graphs for Schedule 2.

(c) **[2 points]** Is Schedule B conflict serializable? ◯ Yes √ **No**

(d) **[3 points]** If not, briefly explain why. If it is conflict serializable, what is the conflict equivalent serial schedule?

Solution: The dependency graph contains a cycle.

(e) **[1 point]** Is the schedule allowed by 2PL? ◯ Yes √ **No**

(f) **[7 points]** If not, briefly explain why. If yes, fill in Table 2 with the lock/unlock requests that could have happened. Use S(·) for shared lock, X(·) for exclusive lock and U(·) to unlock.

Solution: It can not possibly be generated by 2PL, since it is not serializable.

Question 3: Deadlock Detection . . . . . [20 points]

[*** SUBMIT ON SEPARATE PAGE ***]

The first questions refer to **Schedule 1** of Table 3, and the last ones refer to **Schedule 2** of Table 4.

time	1	2	3	4
T1	S(A)			S(B)
T2		X(A)		
T3			X(B)	

Table 3: Schedule 1 - Deadlock Detection.

time	1	2	3	4	5
T4	S(D)				S(F)
T5		X(D)			
T6			X(F)	X(D)	

Table 4: Schedule 2 - Deadlock Detection.

(a) **[2 points]** For Schedule 1, assuming no other transactions, mention which lock request will be granted (g) or blocked (b) by the lock manager

Solution: g, b, g, b

(b) **[5 points]** Give the wait-for graph for Schedule 1.

Solution: See Figure 3

Grading info: wrong direction of arrow: -0.5, wrong arrow: -1, missing arrow: -2

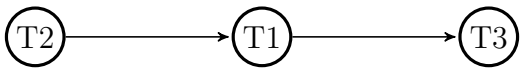

Figure 3: Solution

(c) **[3 points]** For Schedule 1, indicate whether there will be a deadlock or not at the end of this sequence, and give a 1-sentence explanation.

Solution: There will **not** be a deadlock. The wait-for graph is cycle-free.

(d) **[2 points]** For Schedule 2, which lock request will be granted (g), and which will be blocked (b) by the lock manager.

Solution: g,b,g,b,b

(e) **[5 points]** Give the wait-for graph for Schedule 2.

Solution: See Figure 4

Grading info: wrong direction of arrow: -0.5, wrong arrow: -1, missing arrow: -2

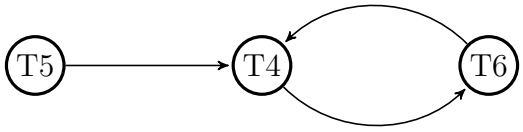

Figure 4: Solution

(f) **[3 points]** For Schedule 2, indicate whether there will be a deadlock or not at the end of this sequence, and give a 1-sentence explanation.

Solution: There will be a deadlock, since there is a cycle in the wait-for graph.

Question 4: B+ tree Locking . . . . . [20 points]

[*** SUBMIT ON SEPARATE PAGE ***]

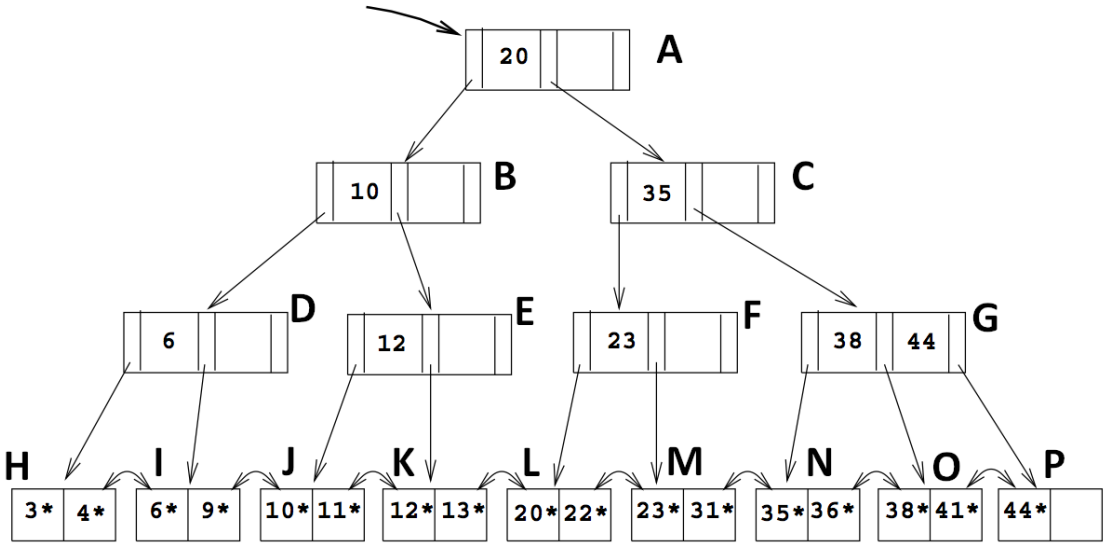

Figure 5: B+ tree Locking. The image is taken from the textbook; Fig. 17.5, p. 563.

Use the **non-conservative** crabbing algorithm, **Bayer-Schkolnick**, to lock the B+ tree. The algorithm is described in slide 54 of the lecture notes #22, as well as in p. 564 of the textbook. For each of the following transactions give the sequence of lock/unlock requests. For example, use S(A) / X(A) / U(A) if the transaction requests a shared lock / exclusive lock /unlock on A respectively.

Grading info: For Shared instead of Exclusive lock: -0.5pt

Grading info: For Exclusive instead of Shared lock: -0.5pt

Grading info: For not unlocking: -0.5pt

Grading info: For completely missing a lock: -1pt

(a) **[5 points]** Search for the data entry 25*.

Solution: S(A), S(C), U(A), S(F), U(C), S(M), U(F), U(M)

(b) **[5 points]** Insert the data entry 39*.

Solution: S(A), S(C), U(A), S(G), X(O), X(G), X(C), U(C), U(G), U(O).
Grading info: Possible error: not getting exclusive lock on G (-1 pt) or C (-1 pt).
Grading info: They are required because O and G are both full.

(c) **[5 points]** Insert the data entry 59*.

Solution: S(A), S(C), U(A), S(G), X(P), U(G), U(C)
Grading info: Possible error: releasing C after S(G) - this is wrong because G is full.
Grading info: -1 pt

(d) **[5 points]** Delete the data entry 13*.

Solution: S(A), S(B), U(A), S(E), U(B), X(K), U(E), U(K)

Question 5: Hierarchical Locking . . . . . [20 points]

[*** SUBMIT ON SEPARATE PAGE ***]

Consider a Database (D) consisting of two tables, `Employees` (E) and `Gym` (G).

- `Employees`, spans *500 pages*, namely E1-E500, while
- `Gym` spans *1000 pages*, i.e., G1-G1000.

Moreover, each page contains *200 records*. For example, the 1st record of the first page of `Employees` is noted as E1:1, the 2nd record as E2:2 etc. Similarly, the 100th record of the 200th page of `Gym` is noted as G200:100.

We use multiple-granularity locking, with S, X, IS, IX, and SIX locks, and four levels of granularity: (a) the *database*-level, (b) the *table*-level, (c) the *page*-level and (d) the *record*-level.

For each of the following operations, give the sequence of lock requests that must be generated by a transaction that wants to carry out these operations. For example, write IS(E)/IS(G) for Intention Shared lock on `Employees`/`Gym` respectively. You do not need to give the sequence of unlocking.

Grading info: Also acceptable: S lock on table

Grading info: Also acceptable: IS lock on table, S lock on record, S lock on another record;

Grading info: escalation S-lock on table

Grading info: If Shared lock instead of Exclusive lock (or vice versa): -1pt

(a) **[4 points]** Read all records on all pages of `Employees`.

Solution: IS(D), S(E)

(b) **[4 points]** Read all the records of `Gym`, and capitalize all initial letters of every gym name, if they are not already capital. That is, *Light weights* is updated to *Light Weights*, but *Skibo* will be left unchanged.

Solution: IS(D), IX(D), SIX(G)
Grading info: Also acceptable: IX(D), SIX(G)

(c) **[4 points]** Read the record G15:4.

Solution: IS(D), IS(G), IS(G15), S(G15:4)

(d) **[4 points]** Update the first record from each and every page of `Employees`.

Solution: IX(D), X(E)
Grading info: Also acceptable with escalation: IX(D), IX(E), X(E1), X(E2), X(E)

(e) **[4 points]** Read all the records from E400:198 to E402:2.

Solution: IS(D), IS(E), IS(E400), S(E401), IS(E402), S(E400:198), S(E400:199), S(E400:200), S(E402:1), S(E402:2)

Grading info: Also acceptable: IS(D), IS(E), S(E400), S(E401), S(E402)