

Databases Autumn 2025

Hand-In Exercise 6

December 21, 2025

Aiysha Frutiger
Jannick Seper
Luis Tritschler

Total Points	

Task	Points

Task 1

a) S_1 :

Conflicting operations:

- $w_1(a)$ before $w_2(a)$ on item a
- $w_3(b)$ before $w_1(b)$ on item b

Dependency graph S_1 :

$$T^3 \longrightarrow T^1 \longrightarrow T^2$$

i) Acyclic dependency graph \Rightarrow conflict-serializable.

Equivalent serial schedule: $T^3 \rightarrow T^1 \rightarrow T^2$.

b) S_2 :

Conflicting operations:

- $w_3(y)$ before $r_4(y)$ on item y

Dependency graph S_2 :

$$T^3 \longrightarrow T^4$$

i) Acyclic dependency graph \Rightarrow conflict-serializable.

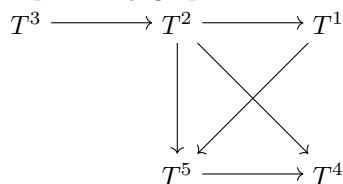
Equivalent serial schedule: $T^1 \rightarrow T^2 \rightarrow T^3 \rightarrow T^4$.

c) S_3 :

Conflicting operations:

- $r_2(w)$ before $w_1(w)$ on item w
- $w_3(x)$ before $w_2(x)$ on item x
- $w_2(x)$ before $w_4(x)$ on item x
- $w_2(y)$ before $w_5(y)$ on item y
- $r_1(v)$ before $w_5(v)$ on item v
- $w_5(v)$ before $w_4(v)$ on item v

Dependency graph S_3 :

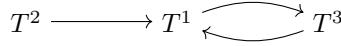


i) Acyclic dependency graph \Rightarrow conflict-serializable.

Equivalent serial schedule: $T^3 \rightarrow T^2 \rightarrow T^1 \rightarrow T^5 \rightarrow T^4$.

Task 2

- a) This is the dependency graph of this schedule:



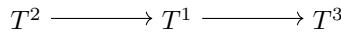
i) **CPSR:**

Cycle in dependency graph \Rightarrow not conflict-serializable $\Rightarrow S_4 \notin \text{CPSR}$.

ii) **OPSR and COPSR:**

$\text{COPSR} \subseteq \text{OPSR} \subseteq \text{CPSR} \Rightarrow S_4 \notin \text{OPSR}$ and $S_4 \notin \text{COPSR}$.

- b) This is the dependency graph of this schedule:



i) **CPSR:**

Acyclic dependency graph \Rightarrow conflict-serializable $\Rightarrow S_5 \in \text{CPSR}$.

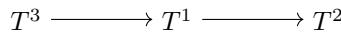
ii) **COPSR:**

Commit order is C_2 before C_1 before C_3 and dependencies $T^2 \rightarrow T^1 \rightarrow T^3 \Rightarrow$ commit-order preserved $\Rightarrow S_5 \in \text{COPSR}$.

iii) **OPSR:**

$\text{COPSR} \subseteq \text{OPSR} \Rightarrow S_5 \in \text{OPSR}$.

- c) This is the dependency graph of this schedule:



i) **CPSR:**

Acyclic dependency graph \Rightarrow conflict-serializable $\Rightarrow S_6 \in \text{CPSR}$.

ii) **COPSR:**

Dependency $T^1 \rightarrow T^2$ exists but commit order is C_2 before $C_1 \Rightarrow$ commit-order violation $\Rightarrow S_6 \notin \text{COPSR}$.

iii) **OPSR:**

Serial order enforced by graph is T^3 before T^1 before T^2 but the real-time order is T^2 before $T^3 \Rightarrow$ order mismatch $\Rightarrow S_6 \notin \text{OPSR}$.

Task 3

- a) **Given schedule:**

$$S_7 = \langle r_2(a) \ r_1(b) \ w_1(b) \ r_1(a) \ r_4(b) \ C_1 \ r_3(c) \ r_3(b) \ w_2(a) \ w_2(b) \ C_2 \ r_3(a) \ C_3 \rangle$$

Committed transactions: T_1, T_2, T_3 (via C_1, C_2, C_3). Transaction T_4 is *active* (no C_4/A_4).

(1) **CPSR: No.**

For conflict-serializability we consider the *committed projection* $C(S_7)$, i.e., remove all actions of non-committed transactions (here: remove T_4):

$$C(S_7) = \langle r_2(a) \ r_1(b) \ w_1(b) \ r_1(a) \ C_1 \ r_3(c) \ r_3(b) \ w_2(a) \ w_2(b) \ C_2 \ r_3(a) \ C_3 \rangle$$

Serialization graph on committed transactions $\{T_1, T_2, T_3\}$ has conflict edges:

- On b : $w_1(b)$ before $r_3(b) \Rightarrow T_1 \rightarrow T_3$.
- On b : $r_3(b)$ before $w_2(b) \Rightarrow T_3 \rightarrow T_2$.
- On a : $w_2(a)$ before $r_3(a) \Rightarrow T_2 \rightarrow T_3$.

Thus there is a cycle $T_2 \leftrightarrow T_3$, hence $S_7 \notin \text{CPSR}$.

(2) **RC: Yes.**

Reads-from relation:

- $r_3(b)$ reads from $w_1(b) \Rightarrow (T_1, b, T_3) \in RF(S_7)$.
- $r_3(a)$ reads from $w_2(a) \Rightarrow (T_2, a, T_3) \in RF(S_7)$.

Since $C_1 < C_3$ and $C_2 < C_3$, the schedule is recoverable: $S_7 \in \text{RC}$.

(3) **ACA: No.**

$r_4(b)$ reads the value written by $w_1(b)$ before T_1 commits:

$$w_1(b) < r_4(b) < C_1,$$

so a dirty read occurs and $S_7 \notin \text{ACA}$.

(4) **ST: No.**

Strictness is violated because T_4 reads b after T_1 wrote b but before T_1 commits:

$$w_1(b) < r_4(b) < C_1.$$

Hence $S_7 \notin \text{ST}$.

(5) **RG: No.**

Rigorous requires that between any two conflicting actions the earlier transaction has already terminated. The conflict $w_1(b) < r_4(b)$ has no C_1/A_1 in between, thus $S_7 \notin \text{RG}$.

Final classification:

$$S_7 \notin \text{CPSR}, \quad S_7 \in \text{RC}, \quad S_7 \notin \text{ACA}, \quad S_7 \notin \text{ST}, \quad S_7 \notin \text{RG}.$$

b) Given schedule:

$$S_8 = \langle r_3(b) \ r_1(a) \ r_4(a) \ w_4(a) \ C_4 \ r_3(a) \ w_3(b) \ r_2(b) \ w_1(c) \ C_1 \ r_2(c) \ w_2(b) \ r_3(b) \ C_3 \ A_2 \rangle$$

Committed transactions: T_1, T_3, T_4 (via C_1, C_3, C_4). Transaction T_2 aborts.

(1) **CPSR: Yes.**

For conflict-serializability we consider the committed projection $C(S_8)$, i.e., remove all actions of non-committed transactions (here: remove T_2):

$$C(S_8) = \langle r_3(b) \ r_1(a) \ r_4(a) \ w_4(a) \ C_4 \ r_3(a) \ w_3(b) \ w_1(c) \ C_1 \ r_3(b) \ C_3 \rangle$$

Serialization graph on committed transactions $\{T_1, T_3, T_4\}$ has conflict edges:

- On a : $r_1(a)$ before $w_4(a)$ $\Rightarrow T_1 \rightarrow T_4$.
- On a : $w_4(a)$ before $r_3(a)$ $\Rightarrow T_4 \rightarrow T_3$.

There are no other conflicts among T_1, T_3, T_4 , and the graph is acyclic (order $T_1 \rightarrow T_4 \rightarrow T_3$). Hence $S_8 \in \text{CPSR}$.

(2) **RC: No.**

Reads-from relation (relevant reads):

- $r_3(a)$ reads from $w_4(a)$ $\Rightarrow (T_4, a, T_3) \in RF(S_8)$.
- $r_2(c)$ reads from $w_1(c)$ $\Rightarrow (T_1, c, T_2) \in RF(S_8)$.
- $r_2(b)$ reads from $w_3(b)$ $\Rightarrow (T_3, b, T_2) \in RF(S_8)$.
- The later $r_3(b)$ reads from $w_2(b)$ $\Rightarrow (T_2, b, T_3) \in RF(S_8)$.

Transaction T_3 commits (C_3) but reads b from T_2 , and T_2 does *not* commit (it aborts: A_2). Therefore the schedule is not recoverable: $S_8 \notin \text{RC}$.

(3) **ACA: No.**

A dirty read occurs: $r_2(b)$ reads the value written by $w_3(b)$ before T_3 commits:

$$w_3(b) < r_2(b) < C_3.$$

Hence $S_8 \notin \text{ACA}$.

(4) **ST: No.**

Strictness is violated because after T_3 writes b , another transaction reads b before T_3 commits:

$$w_3(b) < r_2(b) < C_3.$$

(Also, $w_2(b) < r_3(b) < A_2$ violates strictness as well.) Thus $S_8 \notin \text{ST}$.

(5) **RG: No.**

Rigorous requires that between any two conflicting actions the earlier transaction has already terminated. The conflict $w_3(b) < r_2(b)$ has no C_3/A_3 in between, hence $S_8 \notin \text{RG}$.

Final classification (S8):

$$S_8 \in \text{CPSR}, \quad S_8 \notin \text{RC}, \quad S_8 \notin \text{ACA}, \quad S_8 \notin \text{ST}, \quad S_8 \notin \text{RG}.$$