

Aufgabe 1

1 Punkt

Identify which of the properties: **conflict serializable**, **recoverable**, **cascadeless**, are fulfilled by the following schedule. If a property is not fulfilled, explain why.

T1:	T2:	T3:	T4:
read(A)			
write(A)			
		read(C)	
	write(B)		
	write(C)		
		read(A)	
			read(B)
read(A)			
	COMMIT		
		write(A)	
			write(B)
			COMMIT
read(B)			
COMMIT			
		COMMIT	

Aufgabe 2

1 Punkt

Consider the following schedule.

- Which transaction must abort to trigger a **cascading rollback** of all other transactions?
- Show the commit order (by inserting the commit commands into the schedule) such that the schedule is **cascadeless**.

T1:	T2:	T3:	T4:
	read(B)		
read(D)			
	write(A)		
			read(A)
			write(C)
		read(E)	
read(C)			
write(B)			
		read(B)	

Aufgabe 3**1 Punkt**

Can the following schedule be the output of a **strict two-phase locking** scheduler?
If yes, add all required lock/unlock instructions. Otherwise, explain why.

T1:	T2:	T3:
	read(A)	
	read(B)	
		read(A)
		write(A)
	write(B)	
	COMMIT	
		read(A)
		read(B)
read(B)		
read(A)		
		COMMIT
COMMIT		

Aufgabe 4

1 Punkt

Consider the following schedule and the **validation based** scheduler. The timestamps correspond to the validation order, i.e., $TS(T_i) = \text{validation}(T_i)$.

T1:	T2:	T3:	T4:
<hr/>			
start			
<hr/>			
	start		
<hr/>			
			start
<hr/>			
		start	
<hr/>			
validate			
<hr/>			
finish			
<hr/>			
	validate		
<hr/>			
	finish		
<hr/>			
		validate	
<hr/>			
		finish	
<hr/>			
			validate
<hr/>			
			finish
<hr/>			

The objects in the database that can be read or written are: A, B, C, D, E, F. The read and write sets of the transactions are:

T1: R-set(T1)={A,B}, W-set(T1)={C,D}
 T2: R-set(T2)={A,C}, W-set(T2)={D,E}
 T3: R-set(T3)={C,E}, W-set(T3)={B,F}

Answer the following questions:

1. Does T2 successfully validate? Explain which conditions are satisfied or which one is violated.
2. Suppose that the validation of T4 and all previous validations succeeded. What could have been the largest read set of T4?

Aufgabe 5

1 Punkt

Consider the following schedule. Indicate what happens at each step when the schedule is processed by a **multiversion timestamp-ordering** scheduler. The transactions start in order with $TS(T1)=1$, $TS(T2)=2$, $TS(T3)=3$, $TS(T4)=4$. Assume that no versions of data item A exist. State if a transaction must be rolled back and also consider cascading rollbacks.

T1:	T2:	T3:	T4:
write(A)			
	read(A)		
		write(A)	
			read(A)
	write(A)		
write(A)			

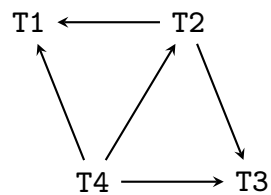
Aufgabe 6

1 Punkt

Consider the following graph in which a directed edge $T_i \rightarrow T_j$ indicates that T_i requests a lock currently held by T_j .

- Which of the transactions would be rolled back under the **wound-wait** deadlock prevention strategy?
- With what timestamp is the rolled-back transaction restarted and why?

Transaction timestamps are equal to their numeric identifiers (smaller numbers are older).



Aufgabe 7

1 Punkt

With the following starting values: A=10, B=20, C=30, D=40, E=50, F=60, write the log file (physical logging) for the following schedule including the log records generated during recovery. After the redo phase, which transactions are contained in the undo list?

```

T1:          T2:          T3:
start
read(A)
A:=A-5
write(A)

                start
                read(C)

read(B)
B:=B-15
write(B)
read(D)

                C:=C+20
                write(C)

                                start
                                read(E)

                COMMIT

-----CHECKPOINT-----
D:=D-20
write(D)

                                E:=E-15
                                write(E)

read(F)

                                COMMIT

F:=F-15
write(F)

-----CRASH-----

```

Aufgabe 81 Punkt

Consider a multi-granularity locking protocol with lock modes S, X, IS, IX, SIX. The following object hierarchy is given:

- root R with blocks A and B,
- block A contains records a1 and a2,
- block B contains records b1, b2, and b3.

Write the list of locks (objects being locked and lock type) required to execute the following transactions. Start with the root.

T1: read all records in block A

T2: modify record b3

Can T1 and T2 be executed concurrently? Why?