Tutorial
Distributed Systems (IN2259)

Exercises on Consistency in Distributed Systems

Exercise 1 Data-centric consistency

Assume a system with N processes P_1 - P_N and shared resources x, y, z. The processes only communicate through the resources.

Example: In Figure 1.1, you see four operations executed by the processes P_1, P_2 and P_3. $w(x)a$ in Line P_1 means that process P_1 writes the value a to resource x. $r(x)a$ in Line P_3 means the value a is read by P_3 from resource x.

The processes are running their operations concurrently. In most cases, there are multiple execution sequences that can be derived from the diagrams. An execution sequence represents one possible execution order of the processes. Depending on the consistency model, different execution orders are possible. An execution order is valid if the read and write semantics are modeled correctly. This means every read on a resource is preceded by the corresponding write. Two possible execution orders in Figure 1.1 are:

1. $w(x)a$ $r(x)a$ $w(x)b$ $r(x)b$ is a valid execution sequence meaning that every read is preceded by the corresponding write.

2. $w(x)a$ $w(x)b$ $r(x)a$ $r(x)b$ is not a valid execution sequence because of $w(x)b$ $r(x)a$.

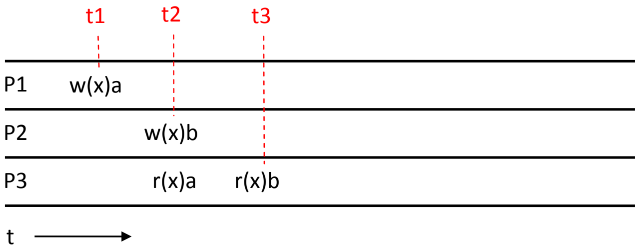

Figure 1.1: Example

(a) Strict Consistency means that any read on a resource x returns a value corresponding to the result of the most recent write on x. This definition implicitly assumes the existence of absolute global time. Is Figure 1.2 strictly consistent? If yes, find a valid execution sequence to prove it. If not, give an explanation.

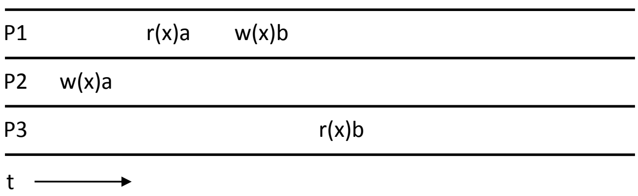

Figure 1.2: Strict Consistency

(b) Linearizability means that all operations are ordered according to their real-time timestamp in the execution sequence. We assume ordering according to a set of synchronized clocks. This implies the operation can be executed at an arbitrary point within a given interval. Is Figure 1.3 lineralizable? Is Figure 1.3 strictly consistent, assuming each operation is performed at the start of its time interval? If yes, find valid execution sequences to prove it. If not, give an explanation.

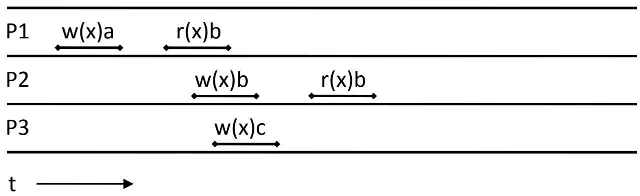

Figure 1.3: Linearizibility

(c) Sequential Consistency means that any valid interleaving of read and write operation is acceptable behavior, but all processes see the same interleaving of operations. Nothing is said about physical time. Is Figure 1.4 sequentially consistent? Is Figure 1.4 linearizable? If yes, find valid execution sequences to prove it. If not, give an explanation. Consider the intervals only to check linearizibility.

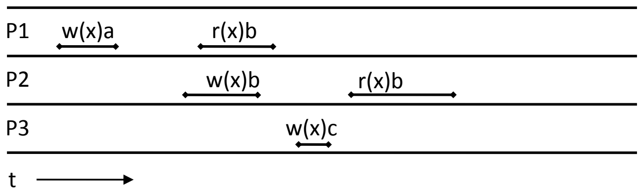

Figure 1.4: Sequential Consistency

(d) Causal consistency means that concurrent writes do not need to be seen in the same order by all readers. Causally related writes, however, must be seen in the same order by every process. This means processes can possibly have different execution sequences. Nothing is said about time. Is Figure 1.5 causally consistent? Is Figure 1.5 sequentially consistent? If yes, find valid execution sequences to prove it. If not, give an explanation.

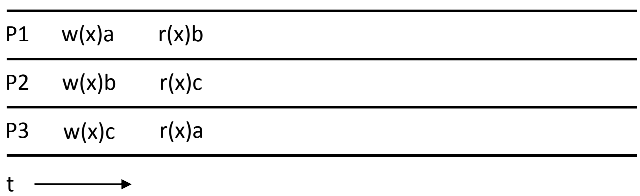

Figure 1.5: Causal Consistency

(e) Is Figure 1.6 causally or sequentially consistent? If yes, find valid execution sequences to prove it. If not, give an explanation.

(f) FIFO consistency means write operations by a single process are seen by all other processes in the order in which they were issued, but writes from different processes may be seen in different orders by different processes. Is Figure 1.7 FIFO? If yes, find valid execution sequences to prove it. If not, give an explanation.

P1	w(x)a				
P2		r(x)a	w(x)b		
P3		r(x)a	w(x)c		
P4				r(x)b	r(x)c
P5				r(x)c	r(x)b

t ⟶

Figure 1.6: Causal Consistency 2

P1	w(x)a			r(x)b	w(x)c			
P2		r(x)a	w(x)b	r(x)b	w(x)d			
P3						r(x)a	r(x)d	r(x)c

t ⟶

Figure 1.7: FIFO Consistency

EXERCISE 2 Client-centric consistency

Client-centric consistency considers consistency from the client's perspective. Client-centric consistency models apply in a mobile context. In Figure 2.8 you see six operations executed by different processes at the locations $L1$, $L2$ and $L3$. $w(x1)$ at Line $L1$ means that version $x1$ of resource x is written to the local replica at location $L1$. $r(x1)$ means that version $x1$ of resource x is read. This may for example represent a mobile device that contacts a local server to fetch emails of a user. After that the user changes the location (dotted line) and a different replica is contacted to fetch the emails again.

(a) Monotonic-read means that once read, subsequent reads on that data item return the same or a more recent value. Do you see any problems why monotonic-read in Figure 2.8 could not be guaranteed. If yes, how would you correct Figure 2.8?

L1	w(x1)		r(x1)		
L2		w(x2)		r(x2)	
L3			w(x3)		r(x3)

t ⟶

Figure 2.8: Figure 2.8: Monotonic-reads

(b) Monotonic-write means that a write operation by a process on a data item x is completed before any subsequent write operation can be issued on x by the same process. Do you see any problems why monotonic-write in Figure 2.9 could not be guaranteed? If yes, how would you correct Figure 2.9 ?

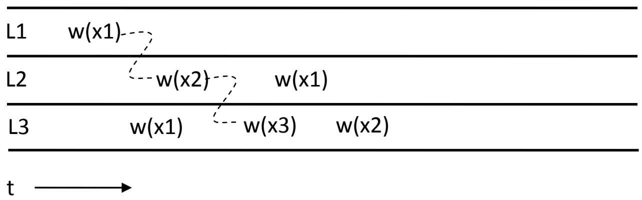

Figure 2.9: Figure 2.9: Monotonic-writes

(c) Read-your-writes means that the effect of a write operation by a process on data item x will always be seen by a successive read operation on x by the same process. Do you see any problems why read-your-writes in Figure 2.10 could not be guaranteed? If yes, how would you correct Figure 2.10?

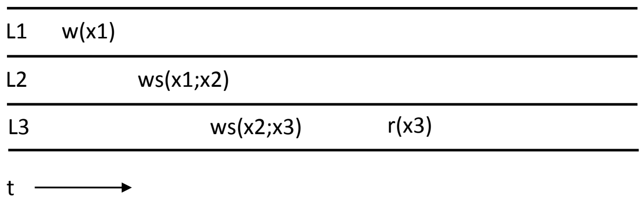

Figure 2.10: Read-your-writes

(d) Writes-follow-reads means that a write operation by a process on a data item x following a previous read operation on x by the same process, is guaranteed to take place on the same or a more recent value of x that was read. Do you see any problems why writes-follow-reads in Figure 2.11 could not be guaranteed? If yes, how would you correct Figure 2.11?

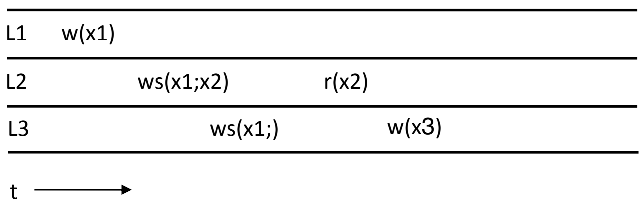

Figure 2.11: Writes-follow-reads