

CONCURRENT SYSTEMS

LECTURE 6

Prof. Daniele Gorla



Atomicity

We have a set of n sequential processes p_1, \dots, p_n that access m concurrent objects X_1, \dots, X_m by invoking operations of the form $X_i.op(args)(ret)$.

When invoked by p_j , the invocation $X_i.op(args)(ret)$ is modeled by two events:
 $inv[X_i.op(args) \text{ by } p_j]$ and $res[X_i.op(ret) \text{ to } p_j]$.

A **history** (or **trace**) is a pair $\hat{H} = (H, <_H)$ where H is a set of events and $<_H$ is a total order on them

The *semantics* (of systems and/or objects) will be given as a set of traces.

A history is **sequential** if it is of the form $inv \ res \ inv \ res \ \dots \ inv \ res \ inv \ inv \ inv \ \dots$ (where every res is the return operation of the immediately preceeding inv)

→ a sequential history can be represented as a sequence of operations

A history is **complete** if every inv is eventually followed by a corresponding res , **partial** otherwise.





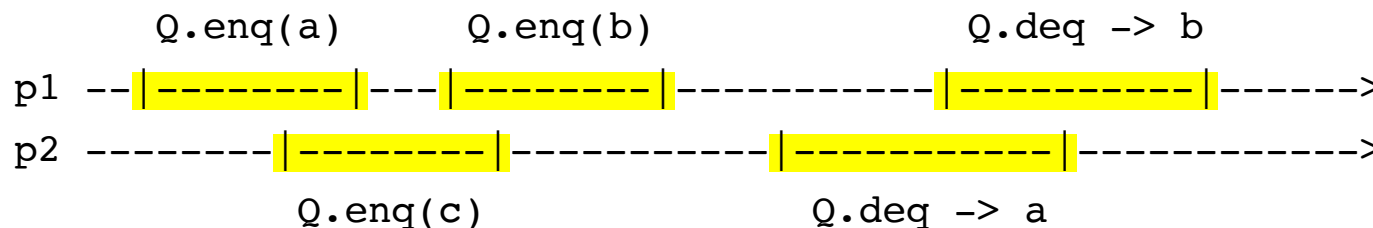
Linearizability

Def.: a complete history \hat{H} is **linearizable** if there exists a sequential history \hat{S} s.t.

1. $\forall X . \hat{S}|_X \in \text{semantics}(X)$
2. $\forall p . \hat{H}|_p = \hat{S}|_p$
3. If $\text{res}[\text{op}] <_H \text{inv}[\text{op}']$, then $\text{res}[\text{op}] <_S \text{inv}[\text{op}']$

Given an history \hat{K} , we can define a binary relation on events \rightarrow_K s.t. $(\text{op}, \text{op}') \in \rightarrow_K$ if and only if $\text{res}[\text{op}] <_K \text{inv}[\text{op}']$. We write $\text{op} \rightarrow_K \text{op}'$ for denoting $(\text{op}, \text{op}') \in \rightarrow_K$.
Hence, condition 3 of the previous Def. requires that $\rightarrow_H \subseteq \rightarrow_S$.

EXAMPLE: Let Q be a queue; let $p1$ and $p2$ be such that



This corresponds to the history

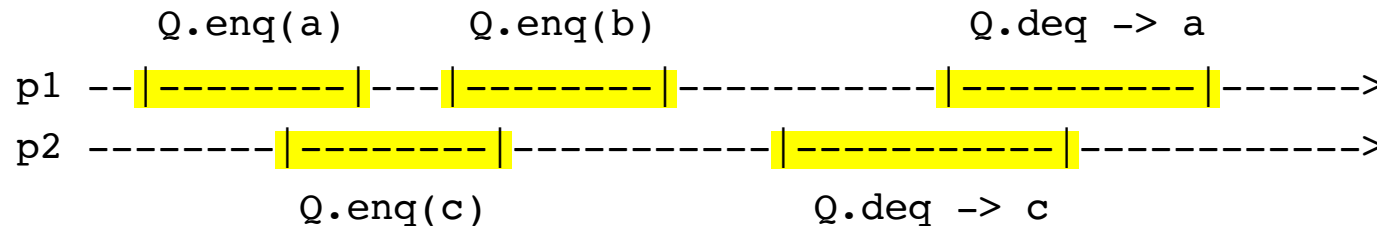
$\text{inv}[Q.enq(a) \text{ by } p1] \text{ inv}[Q.enq(c) \text{ by } p2] \text{ res}[Q.enq(a) \text{ to } p1] \text{ inv}[Q.enq(b) \text{ by } p1]$
 $\text{res}[Q.enq(c) \text{ by } p2] \text{ res}[Q.enq(b) \text{ by } p1] \text{ inv}[Q.deq() \text{ by } p2] \text{ inv}[Q.deq() \text{ by } p2]$
 $\text{res}[Q.deq(a) \text{ to } p2] \text{ res}[Q.deq(b) \text{ to } p1]$

It can be linearized as $[Q.enq(a)() \text{ by } p1] [Q.enq(b)() \text{ by } p1] [Q.enq(c)() \text{ by } p2] [Q.deq()(a) \text{ to } p2]$
 $[Q.deq()(b) \text{ to } p1]$



Linearizability (cont.'d)

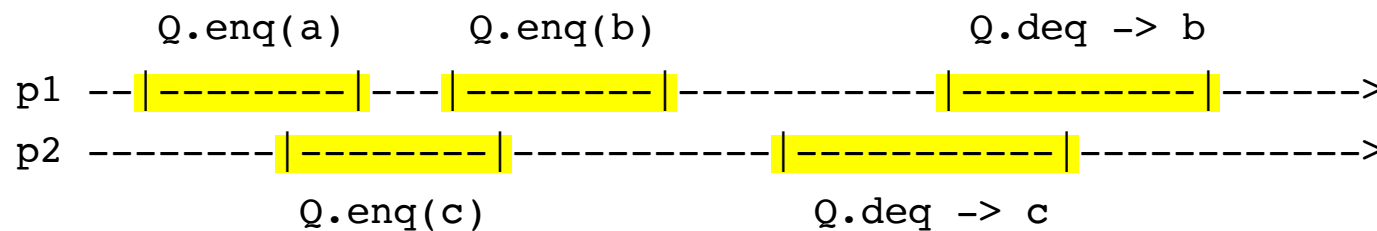
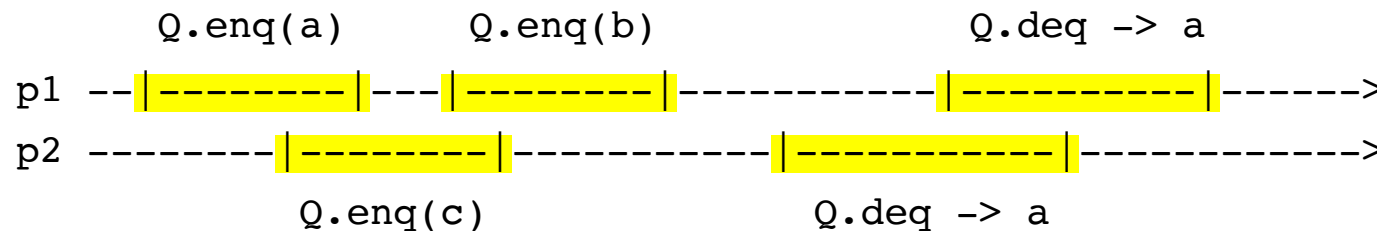
Now consider



The corresponding history can still be linearized as

$[Q.enq(c)() \text{ by } p2] [Q.enq(a)() \text{ by } p1] [Q.enq(b)() \text{ by } p1] [Q.deq()(c) \text{ to } p2] [Q.deq()(a) \text{ to } p1]$

By contrast, the following are not linearizable histories:





Thm (compositionality): \hat{H} is linearizable if $\hat{H}|_X$ is linearizable, for all X involved in H

Proof:

For all X , let \hat{S}_X be a linearization of $\hat{H}|_X$

$\rightarrow \hat{S}_X$ defines a total order on the operations on X (call it \rightarrow_X)

Let \rightarrow denote $\rightarrow_H \cup \bigcup_{X \text{ in } H} \rightarrow_X$ *(recall that a relation is a set of pairs, so here you take the union of all pairs of \rightarrow_H and of all \rightarrow_X)*

We now show that \rightarrow is acyclic.

1. It cannot have cycles with 1 edge (i.e., self loops): indeed, if $op \rightarrow op$, this would mean that $res(op) < inv(op)$
2. It cannot have cycles with 2 edges: by contr., assume that $op \rightarrow op' \rightarrow op$
 - both arrows cannot be \rightarrow_H nor \rightarrow_X (for some X), otw. such relations were cyclic
 - it cannot be that one is \rightarrow_X and the other \rightarrow_Y (for some $X \neq Y$), otw. op/op' would be on 2 different objects

Hence, it must be $op \rightarrow_X op' \rightarrow_H op$ (or vice versa)

Then, $op' \rightarrow_H op$ means that $res(op') <_H inv(op)$

Since \hat{S}_X is a linearization of $\hat{H}|_X$ and op/op' are on X , this implies $res(op') <_X inv(op)$, i.e., that $op' \rightarrow_X op \rightarrow \rightarrow_X$ would be cyclic



3. It cannot have cycles with more than 2 edges: by contr., consider a shortest cycle

- adjacent edges cannot belong to the same order (otw. the cycle would be shortable, because of transitivity)
- adjacent edges cannot belong to orders on different objects

Hence, at least one \rightarrow_X exists, and it must be between two \rightarrow_H , i.e.:

$$\text{op1} \rightarrow_H \text{op2} \rightarrow_X \text{op3} \rightarrow_H \text{op4}$$

is part of the shortest cycles chosen (possibly with $\text{op4}=\text{op1}$).

$\text{op1} \rightarrow_H \text{op2}$ means that $\text{res}(\text{op1}) <_H \text{inv}(\text{op2})$

$\text{op2} \rightarrow_X \text{op3}$ entails that $\text{inv}(\text{op2}) <_H \text{res}(\text{op3})$

Indeed, if not, we would have that $\text{res}(\text{op3}) <_H \text{inv}(\text{op2})$, since $<_H$ is
a total order \rightarrow we would have a cycle of length 2 ⚡

$\text{op3} \rightarrow_H \text{op4}$ means that $\text{res}(\text{op3}) <_H \text{inv}(\text{op4})$

By transitivity of $<_H$, we would then have that $\text{res}(\text{op1}) <_H \text{inv}(\text{op4})$, i.e. $\text{op1} \rightarrow_H \text{op4}$

\rightarrow in contradiction with having chosen a shortest cycle





Every DAG admits a topological order (i.e., a total order of its nodes that respects the edges)

→ Let \rightarrow' denote a topological order for \rightarrow

Let us then define a linearization of \hat{H} as follows:

$\hat{S} = \text{inv}(\text{op1}) \text{res}(\text{op1}) \text{inv}(\text{op2}) \text{res}(\text{op2}) \dots$ whenever $\text{op1} \rightarrow' \text{op2} \rightarrow' \dots$

\hat{S} is clearly sequential; moreover:

1. For all X , $\hat{S}|_X = \hat{S}_X$ ($\in \text{semantics}(X)$). Indeed:
 - $<_{\hat{S}_X} = \rightarrow_X \subseteq \rightarrow|_X \subseteq \rightarrow'|_X = \rightarrow_{\hat{S}|_X} = <_{\hat{S}|_X}$
 - Since $<_{\hat{S}_X}$ and $<_{\hat{S}|_X}$ are total orders on the same set of events (i.e., $A|_X$), they must coincide
2. For all p , $\hat{H}|_p = \text{inv}(\text{op1}_p) \text{res}(\text{op1}_p) \text{inv}(\text{op2}_p) \text{res}(\text{op2}_p) \dots$ (bec. p is sequential)
 $= \hat{S}|_p$ (bec. $\text{op1}_p \rightarrow_H \text{op2}_p \rightarrow_H \dots$ and $\rightarrow_H \subseteq \rightarrow'$)
3. $\rightarrow_H \subseteq \rightarrow \subseteq \rightarrow' = \rightarrow_S$





Alternatives to Atomicity (1)

Sequential consistency

Let us define $op \rightarrow_{\text{proc}} op'$ to hold whenever there exists a process p that issues both operations, with $\text{res}[op]$ happening before $\text{inv}[op']$.

Def.: a complete history \hat{H} is **sequentially consistent** if there exists a sequential history \hat{S} s.t.

1. $\forall X . \hat{S}|_X \in \text{semantics}(X)$ *(like linearizability)*
2. $\forall p . \hat{H}|_p = \hat{S}|_p$ *(like linearizability)*
3. $\rightarrow_{\text{proc}} \subseteq \rightarrow_S$ *(in place of $\rightarrow_H \subseteq \rightarrow_S$)*

This is a more generous notion than linearizability.

EXAMPLE: Let \hat{H} be [Q.enq(a)() by p1] [Q.enq(b)() by p2] [Q.deq()(b) to p2]

→ not linearizable: ■ the only possible linearization of \hat{H} is \hat{H} itself (because of cond.3)

■ it violates the semantics of a queue (cond.1)

→ it is sequentially consistent, by swapping the first two actions, i.e. by considering \hat{S} to be

[Q.enq(b)() by p2] [Q.enq(a)() by p1] [Q.deq()(b) to p2]





Alternatives to Atomicity (1)

The problem with sequential consistency is that it is NOT compositional.

EXAMPLE

Consider the following two processes:

p1: $Q.\text{enq}(a) ; Q'.\text{enq}(b') ; Q'.\text{deq}() \rightarrow b'$

p2: $Q'.\text{enq}(a') ; Q.\text{enq}(b) ; Q.\text{deq}() \rightarrow b$

In isolation, both processes are sequentially consistent

However, no total order on the previous 6 operations respects the semantics of a queue:

- If p1 receives b' from $Q'.\text{deq}$, we have that $Q'.\text{enq}(a')$ must arrive after $Q'.\text{enq}(b')$
- To respect $\rightarrow_{\text{proc}}$, also the remaining behaviour of p2 must arrive after
- Hence, $Q.\text{enq}(a)$ arrived before $Q.\text{enq}(b)$ and so it is not possible for p2 to receive b from its $Q.\text{deq}$

Hence, we have two histories that are sequentially consistent but whose composition cannot be sequentially consistent \rightarrow no compositionality!





Alternatives to Atomicity (2)

Serializability (typical notion in databases)

- We now have transactions instead of processes
- Consequently, we have also two other kinds of events: abort(t) and commit(t)
- The constraint is that, in every history, we have at most one of these events for every transaction; if the history is complete, we must have exactly one of these events for every transaction
- A sequential history is formed by committed transactions only

Def.: a complete history \hat{H} is **serializable** if there exists a sequential history \hat{S} s.t.

1. $\forall X . \hat{S}|_X \in \text{semantics}(X)$ *(like linearizability)*
2. $S = \{e \in H : e \in t \in \text{committedTrans}(\hat{H})\}$
3. $\rightarrow_{\text{trans}} \subseteq \rightarrow_S$ *(where $\rightarrow_{\text{trans}}$ is defined like $\rightarrow_{\text{proc}}$ in seq. cons.)*

Again, this is a more generous notion than linearizability, but it is not compositional

→ consider the previous two examples, where instead of processes, you have transactions

