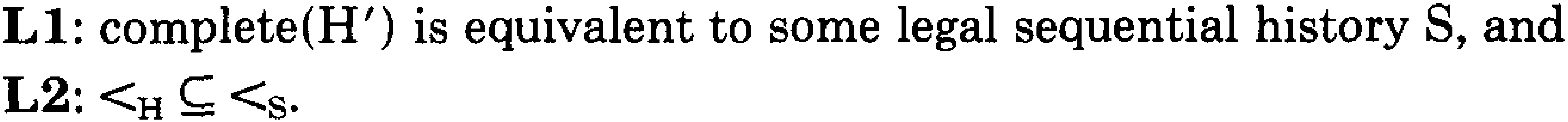
How do we specify correctness for concurrent objects?

Linearizability



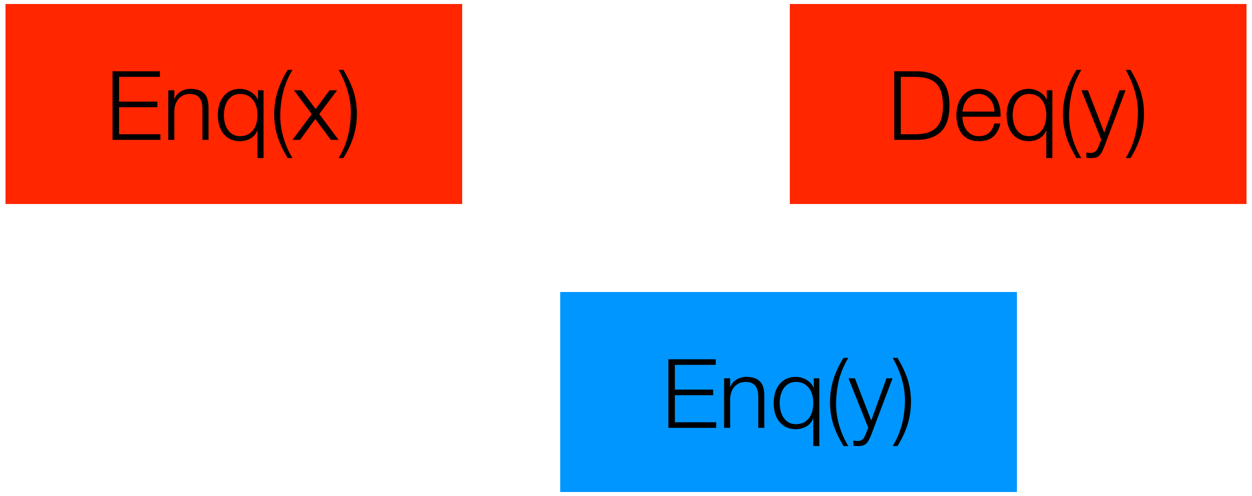
L1: complete(H′) is equivalent to some legal sequential history S, and
L2: $<_H \subseteq <_S$.

Execution history has equivalent behavior to some sequential execution

Each method call takes effect instantaneously at some point between call/return

Linearizability does not require blocking (or non-blocking!) behavior

Linearizable FIFO history?

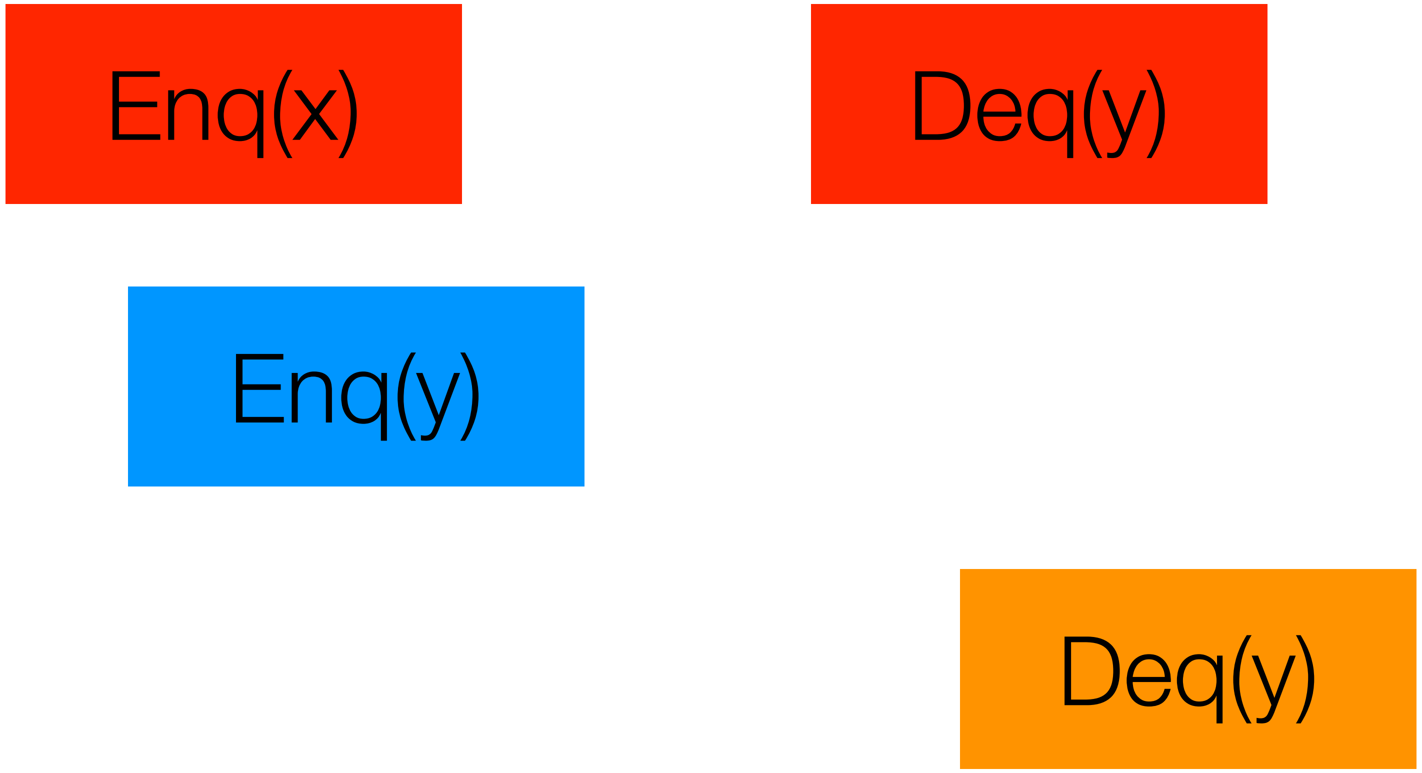


Enq(x)

Deq(y)

Enq(y)

Linearizable FIFO history?



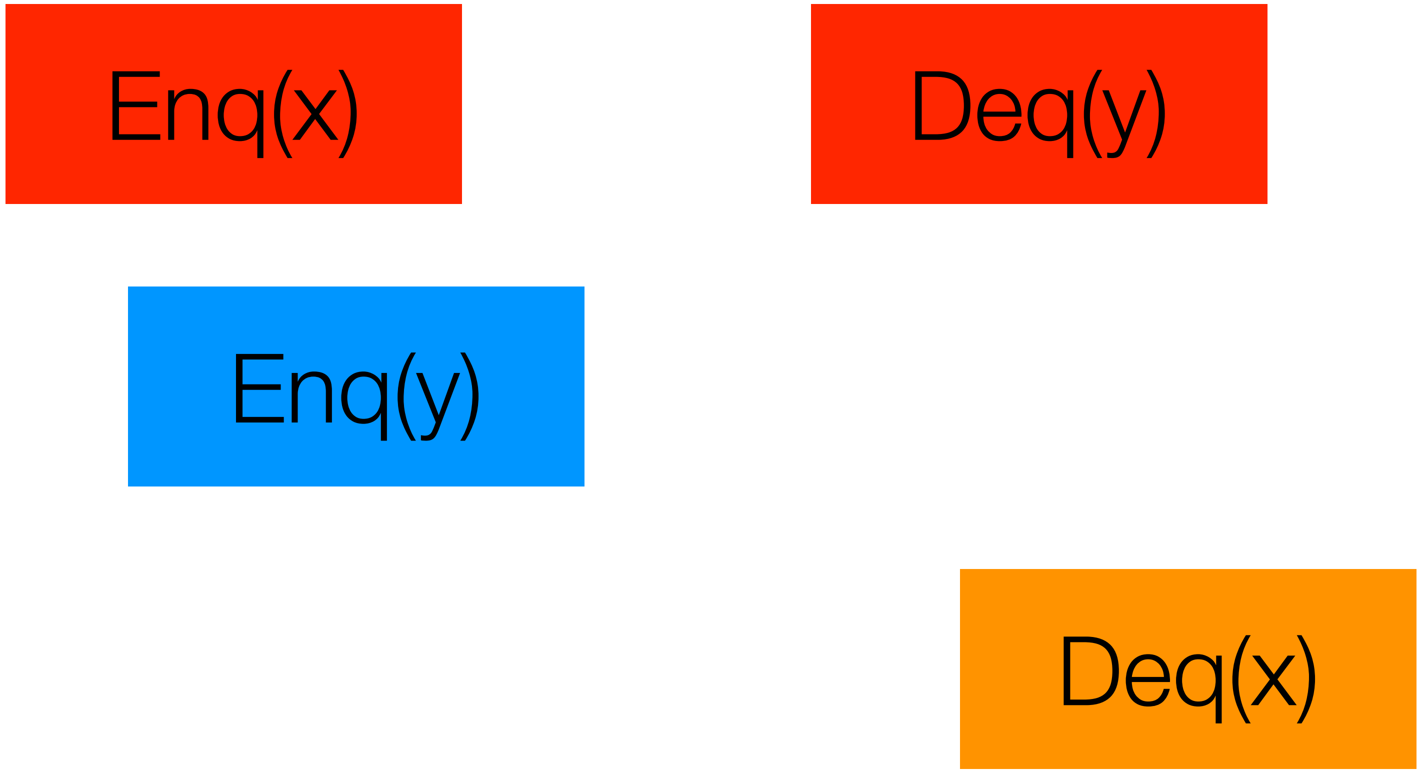
Enq(x)

Deq(y)

Enq(y)

Deq(y)

Linearizable FIFO history?



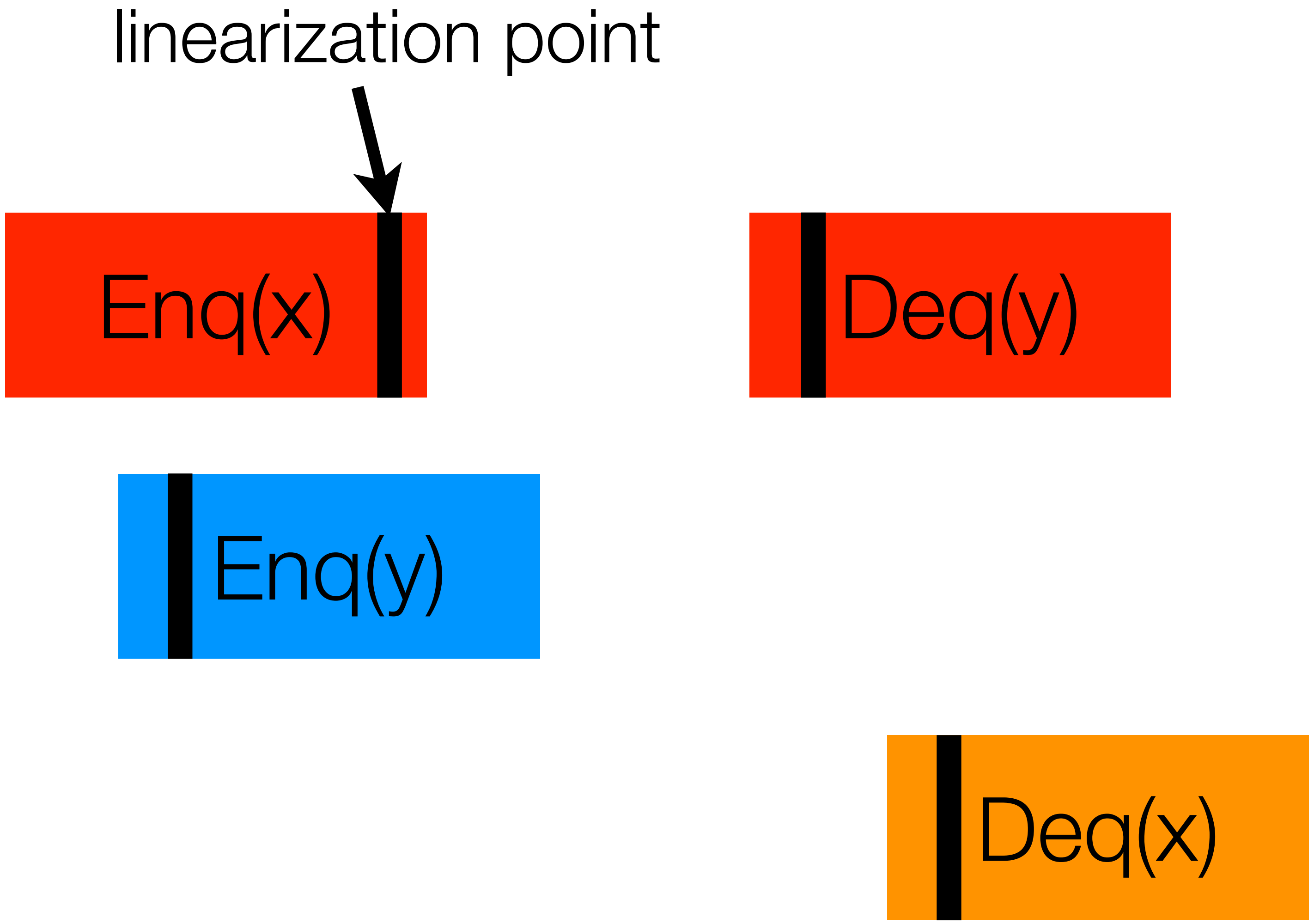
Enq(x)

Deq(y)

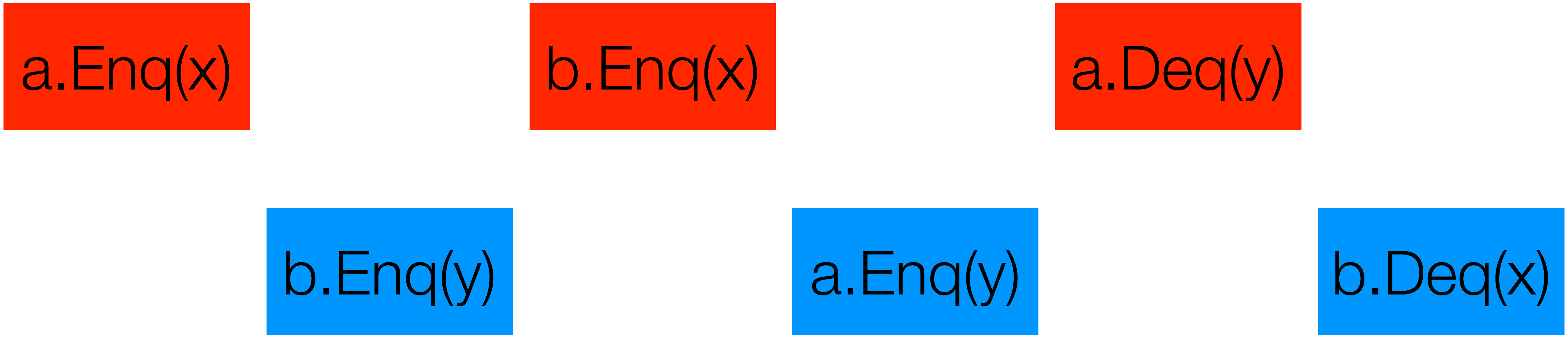
Enq(y)

Deq(x)

Linearizable FIFO history?



SC is not local



a.Enq(x)

b.Enq(y)

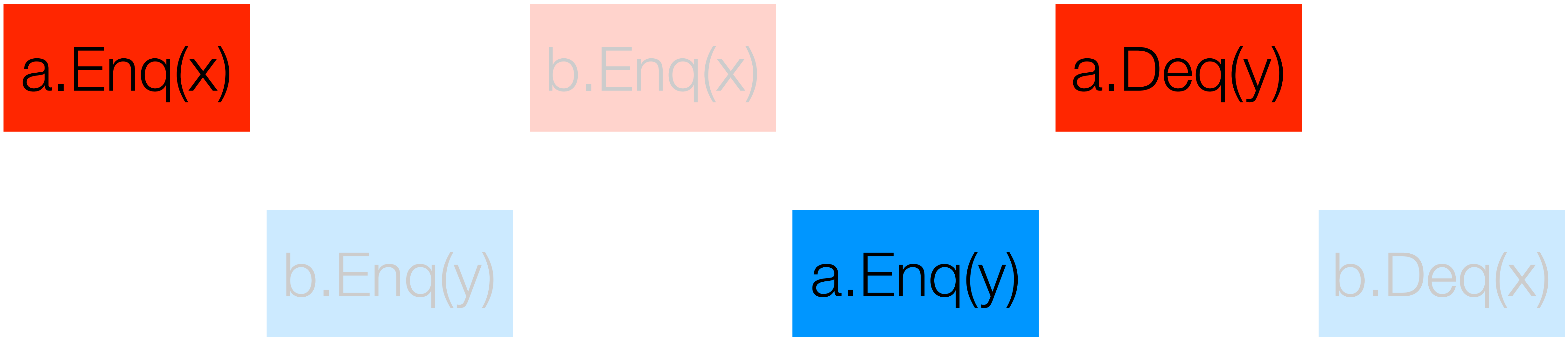
b.Enq(x)

a.Enq(y)

a.Deq(y)

b.Deq(x)

SC is not local



a.Enq(x)

b.Enq(y)

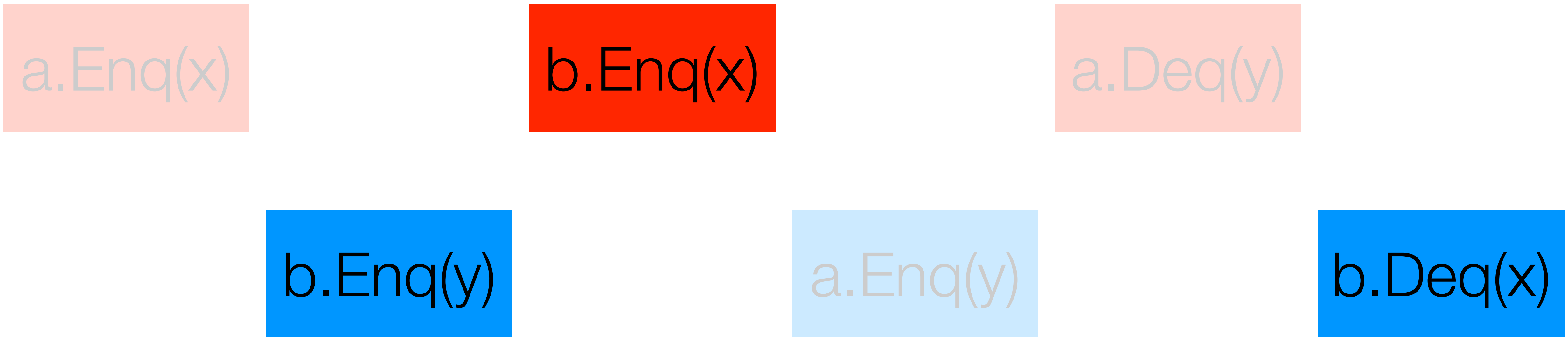
b.Enq(x)

a.Enq(y)

a.Deq(y)

b.Deq(x)

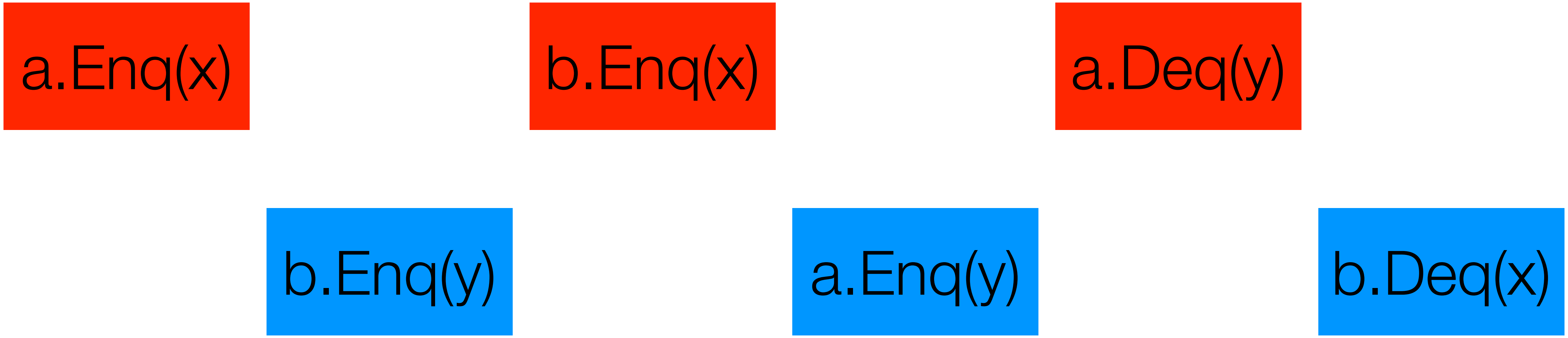
SC is not local



a.Enq(x) b.Enq(x) a.Deq(y)

b.Enq(y) a.Enq(y) b.Deq(x)

SC is not local



a.Enq(x)

b.Enq(x)

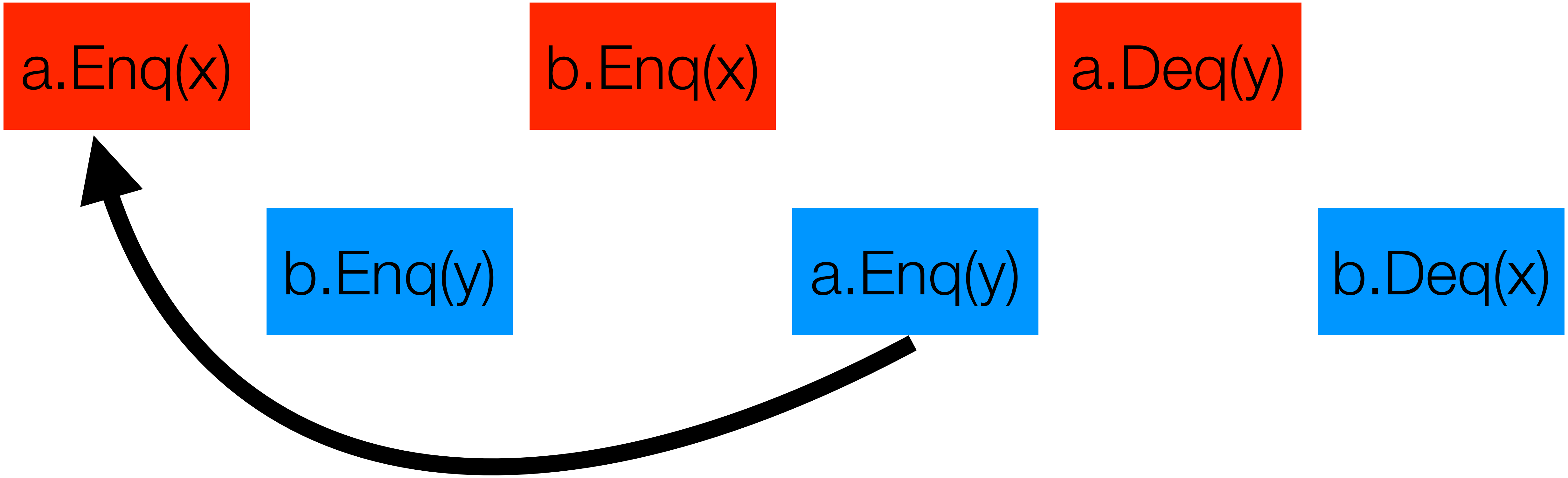
a.Deq(y)

b.Enq(y)

a.Enq(y)

b.Deq(x)

SC is not local



a.Enq(x)

b.Enq(x)

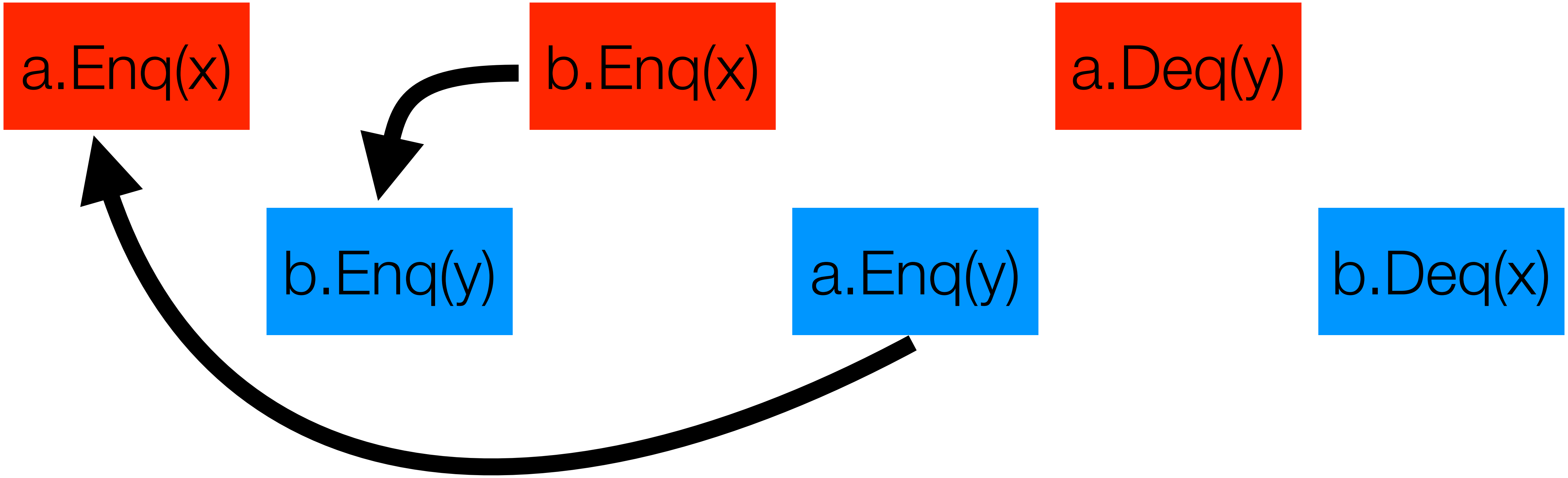
a.Deq(y)

b.Enq(y)

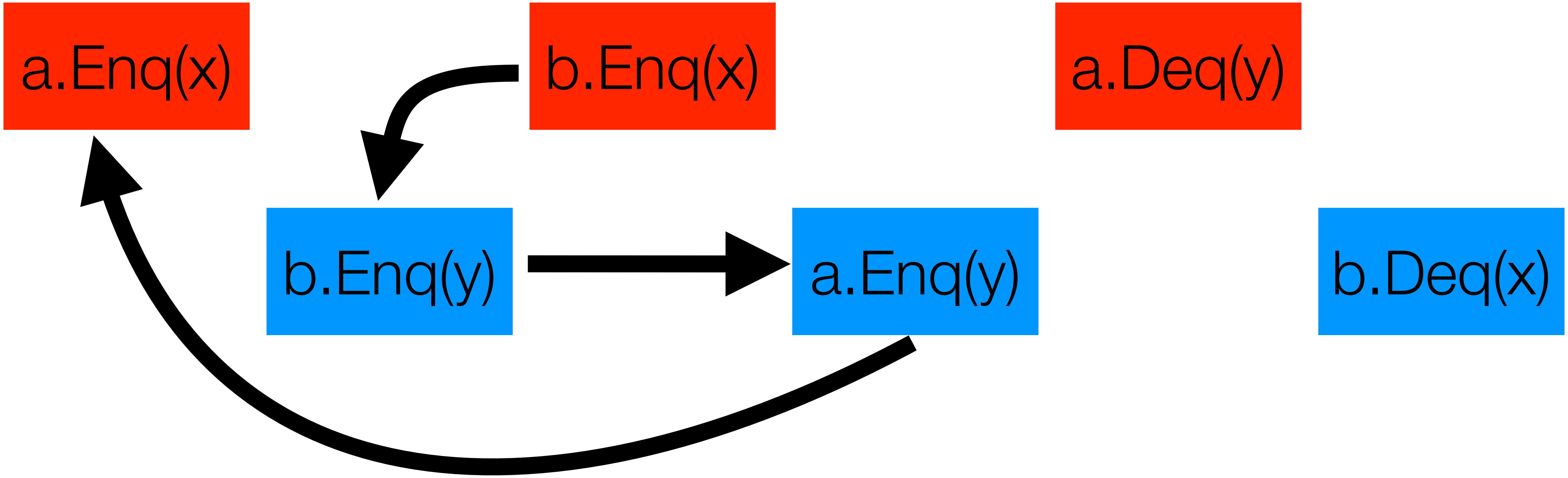
a.Enq(y)

b.Deq(x)

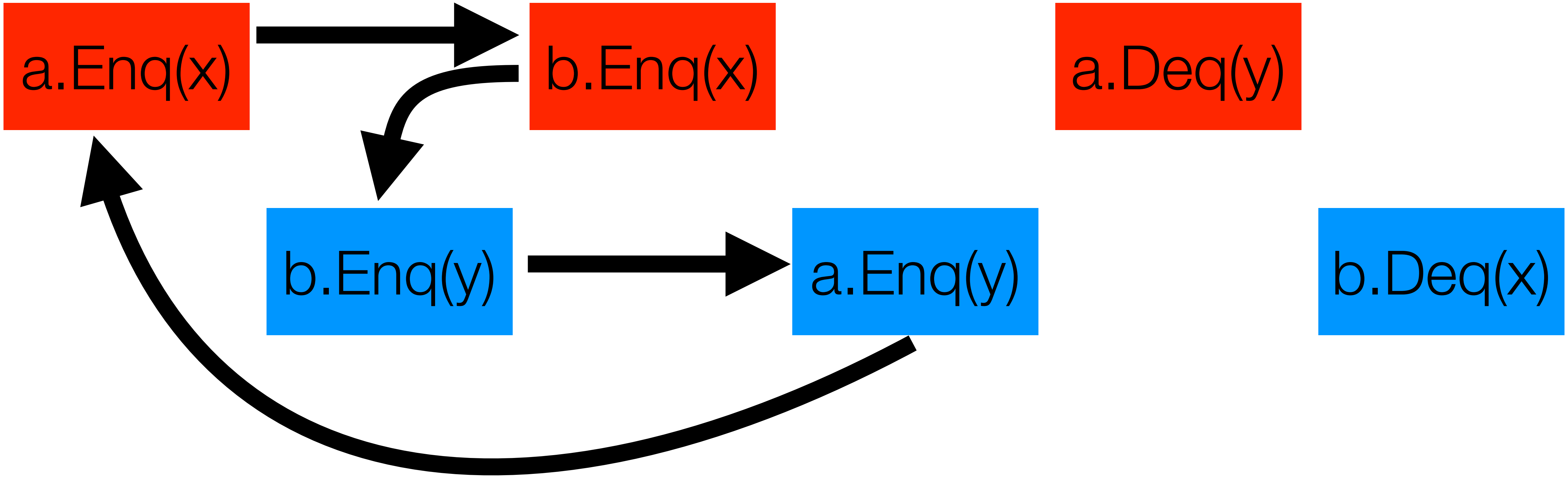
SC is not local



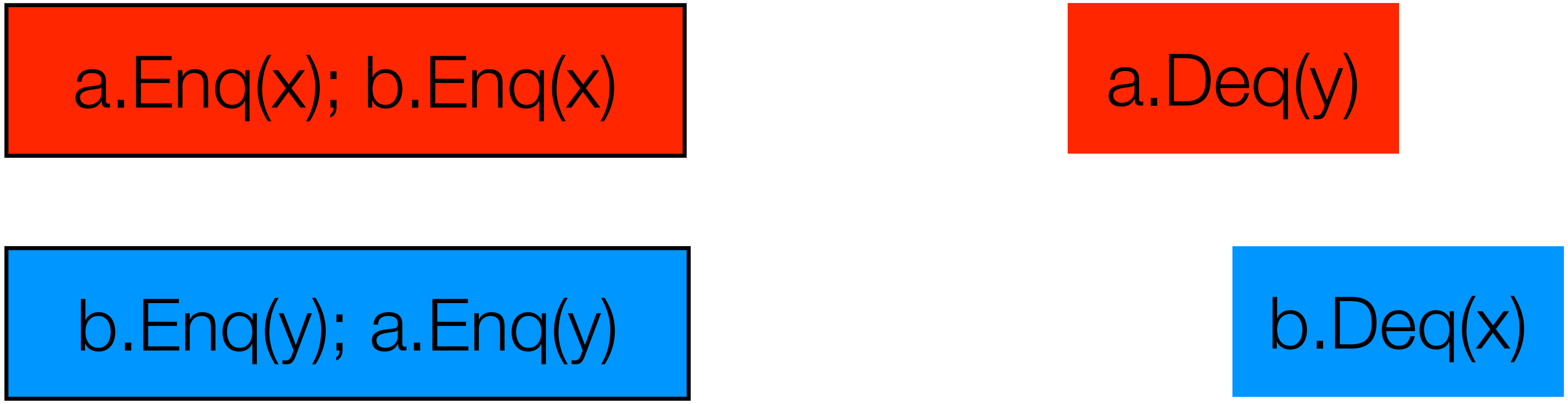
SC is not local



SC is not local



Serializability is not local



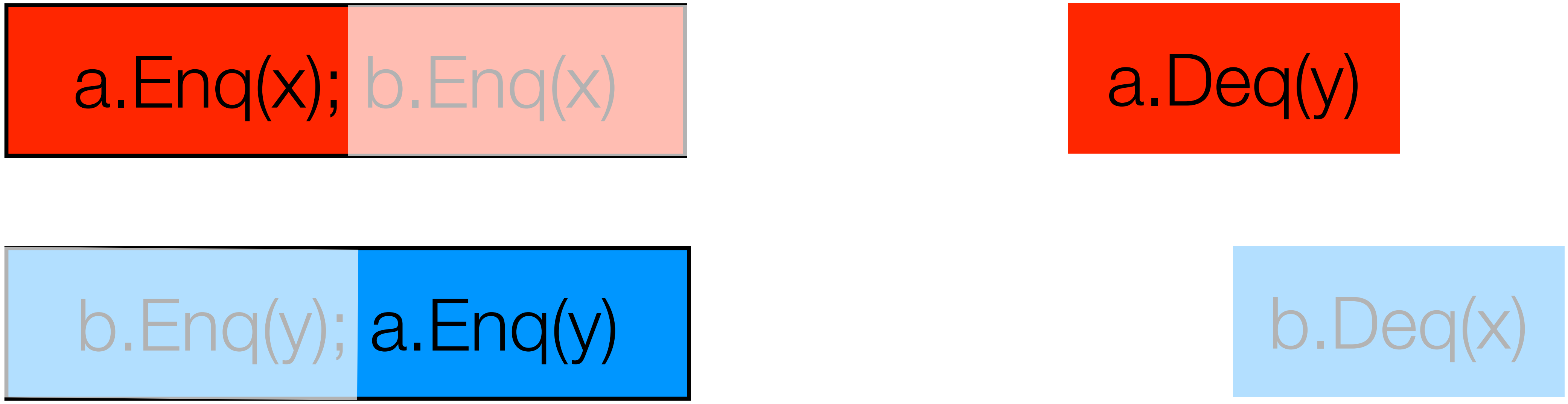
a.Enq(x); b.Enq(x)

a.Deq(y)

b.Enq(y); a.Enq(y)

b.Deq(x)

Serializability is not local



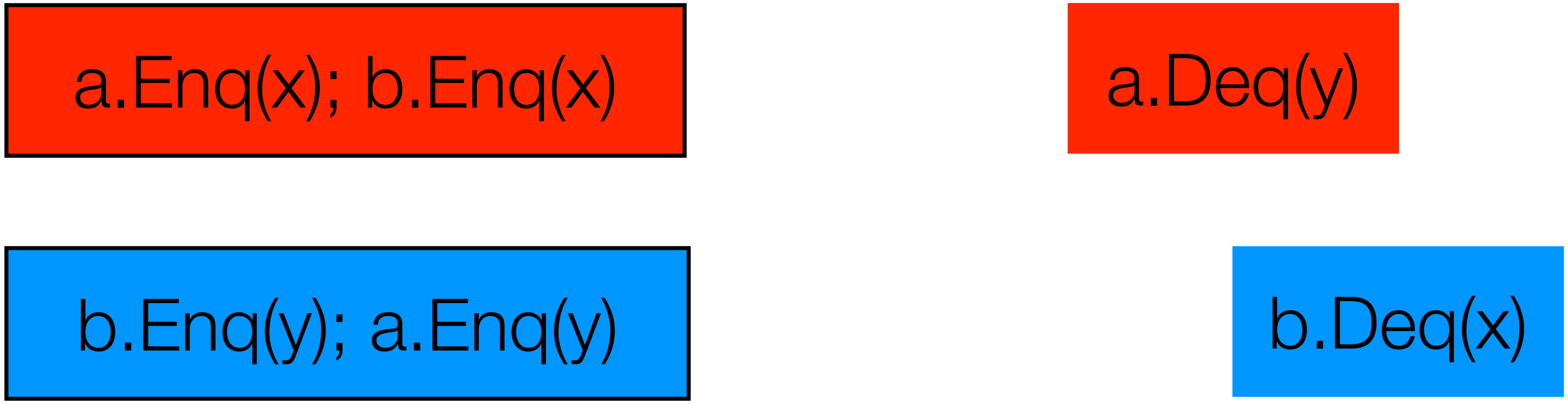
a.Enq(x); b.Enq(x)

a.Deq(y)

b.Enq(y); a.Enq(y)

b.Deq(x)

Serializability is not local



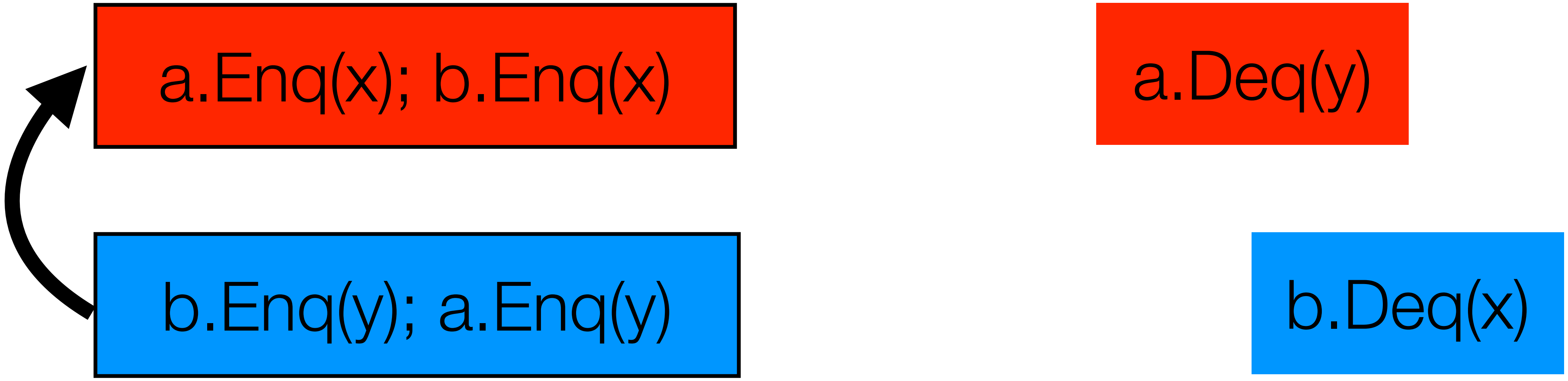
a.Enq(x); b.Enq(x)

a.Deq(y)

b.Enq(y); a.Enq(y)

b.Deq(x)

Serializability is not local



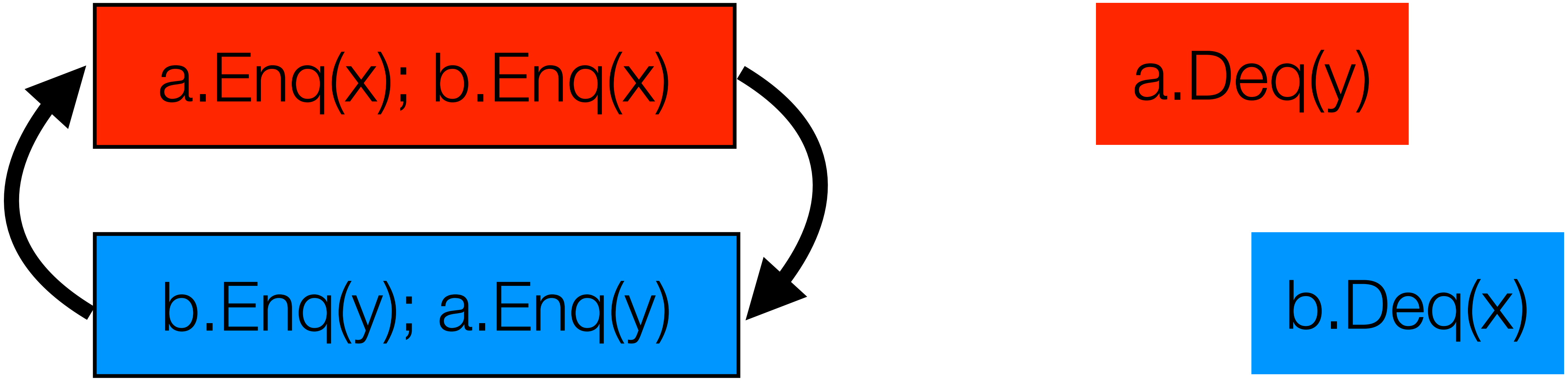
a.Enq(x); b.Enq(x)

a.Deq(y)

b.Enq(y); a.Enq(y)

b.Deq(x)

Serializability is not local



a.Enq(x); b.Enq(x)

b.Enq(y); a.Enq(y)

a.Deq(y)

b.Deq(x)

Different granularities

- SC: single memory location
- Linearizability: single object
- Serializability: arbitrary set of memory locations
- Unifying theme: reduce concurrency to (nondeterministic) sequential behavior