

## Linearizability

### Exercise 1

For the following history of a shared register with the operations  $\text{write}(x)/\text{void}$  and  $\text{read}()/x$  answer the questions below.

B:  $r.\text{write}(1)$

A:  $r.\text{read}()$

C:  $r.\text{write}(2)$

A:  $r:1$

B:  $r:\text{void}$

C:  $r:\text{void}$

B:  $r.\text{read}()$

B:  $r:1$

A:  $q.\text{write}(3)$

C:  $r.\text{read}()$

A:  $q:\text{void}$

- What is  $H|B$ ?
- What is  $H|r$ ?
- Turn  $H$  into a complete subhistory  $H'$ .
- Is  $H'$  sequential?
- Is  $H'$  well-formed?
- Is  $H'$  linearizable? If yes, prove it!
- If the first two events are swapped, is the resulting history equivalent to  $H$ ?

### Exercise 2

In the following history, do the marked method executions overlap? Or does the method invocation denoted by bold events precede the one which is underlined?

**A:  $q.\text{enq}(x)$**

B:  $q.\text{enq}(y)$

**A:  $q:\text{void}$**

B:  $q:\text{void}$

B:  $q.\text{deq}()$

A:  $q.\text{deq}()$

B:  $q:x$

### Exercise 3

Is the following history of a fifo queue with the operations  $\text{enq}(x)/\text{void}$   $\text{deq}()/x$  linearizable? If yes, prove it! Is it sequentially consistent?

A: r.enq(x)

A: r:void

B: r.enq(y)

A: r.deq()

B: r:void

A: r:y

#### **Exercise 4**

Is the following history of a fifo queue with the operations enq(x)/void deq()/x linearizable? If yes, prove it!

A: q.enq(x)

B: q.enq(y)

A: q:void

B: q:void

A: q.deq()

C: q.deq()

A: q:y

C: q:y

#### **Parallel FIFO Queue Implementation**

Implement a queue with fifo semantics in C or C++, using either OpenMP or POSIX threads. Use locks to implement the queue. Benchmark your implementation on your own machine. Make a diagram out of your benchmark results.