Semantics for Concurrency

Ana Sokolova UNIVERSITY of SALZBURG

Introduction - 5.3.19

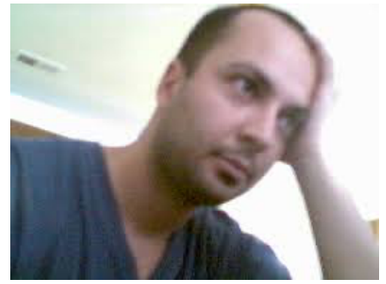
- Intro: Concurrent data structures
  correctness and performance

Concurrent Data Structures Correctness and Relaxations

Hannes Payer
Google

Tom Henzinger
IST AUSTRIA

Christoph Kirsch
UNIVERSITY of SALZBURG



Ali Sezgin UNIVERSITY OF CAMBRIDGE

Andreas Haas Google

Michael Lippautz

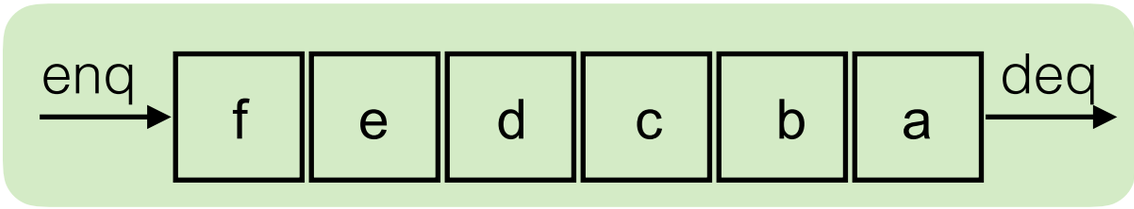
Andreas Holzer
Google

Helmut Veith TU WIEN

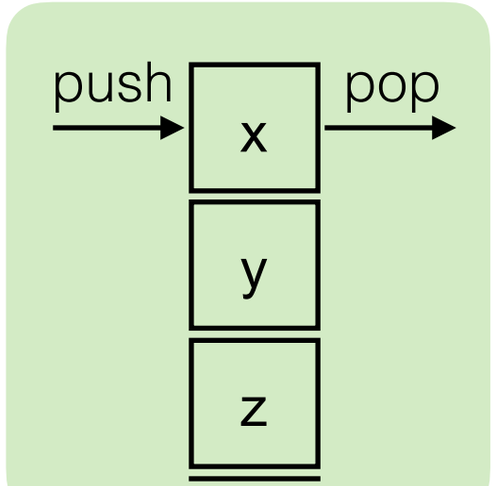
RiSE
Rigorous Systems Engineering

Data structures

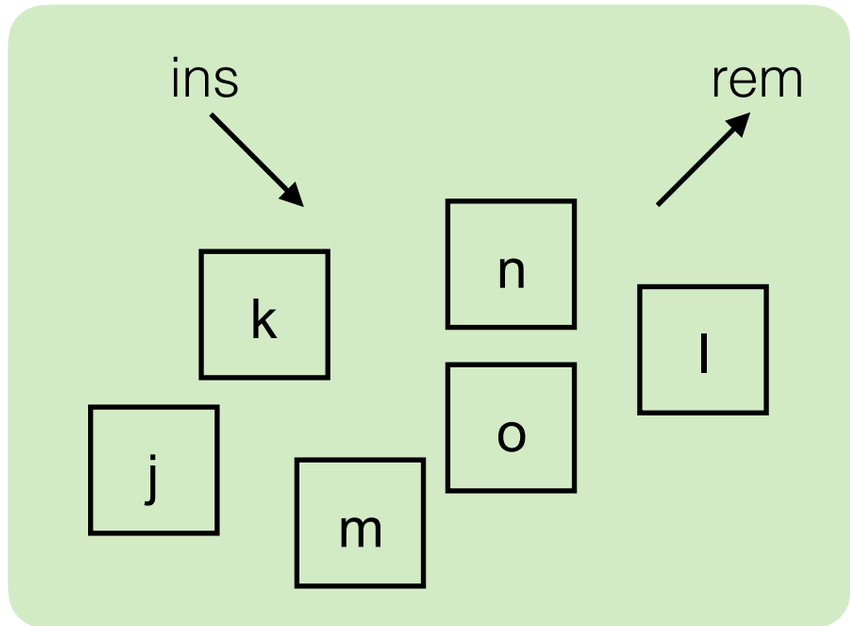
- Queue FIFO



- Stack LIFO

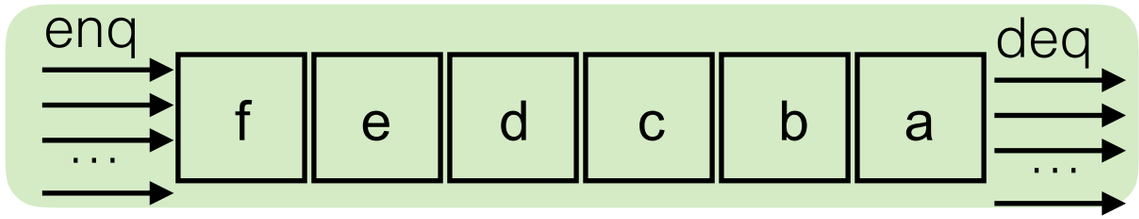


- Pool unordered

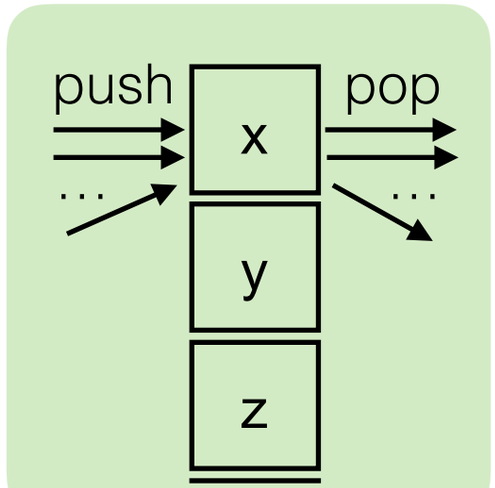


Concurrent data structures

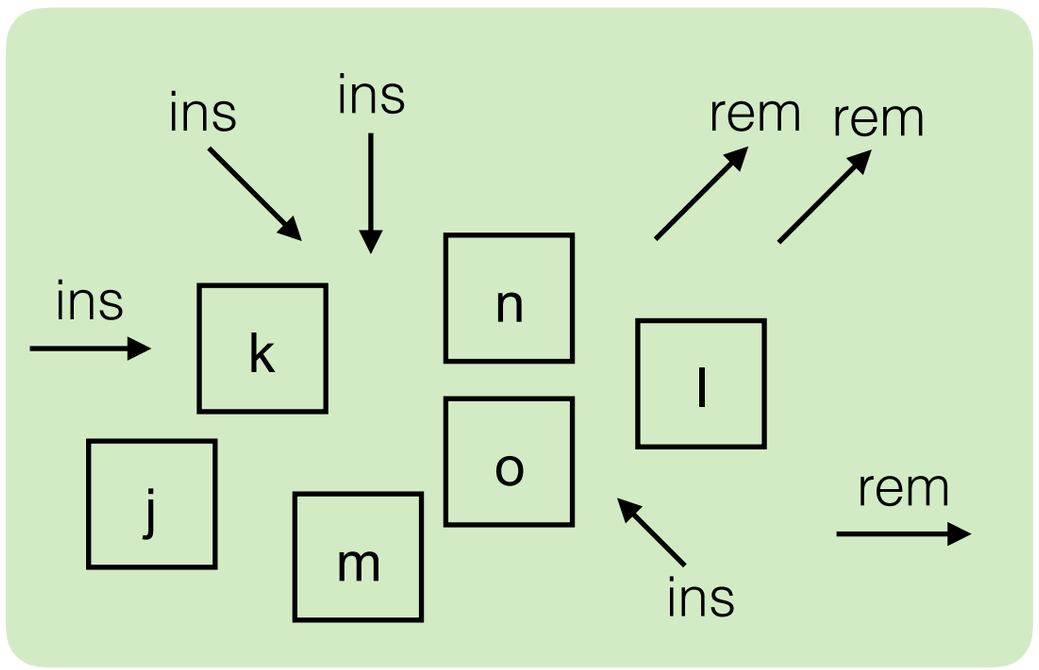
- Queue FIFO



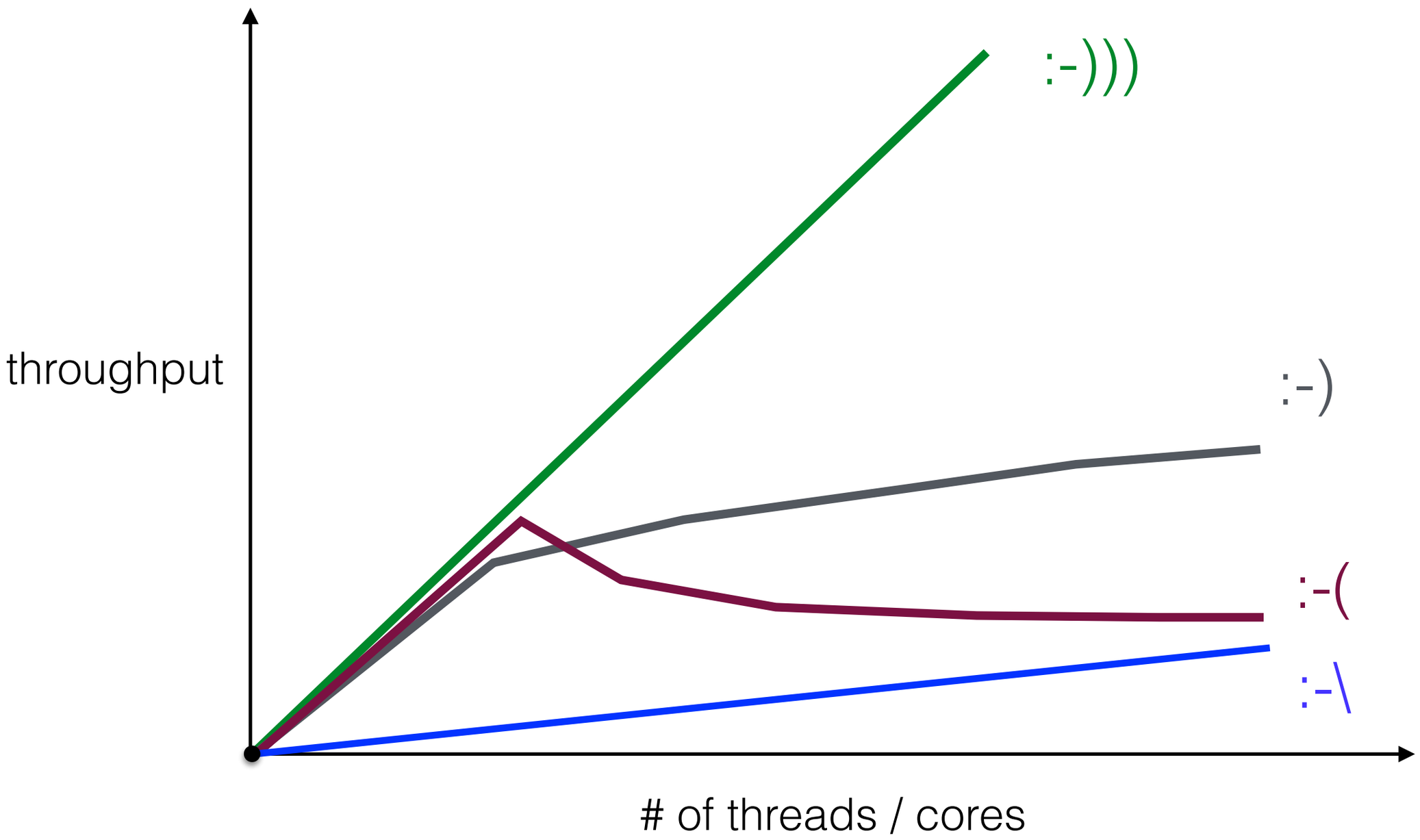
- Stack LIFO



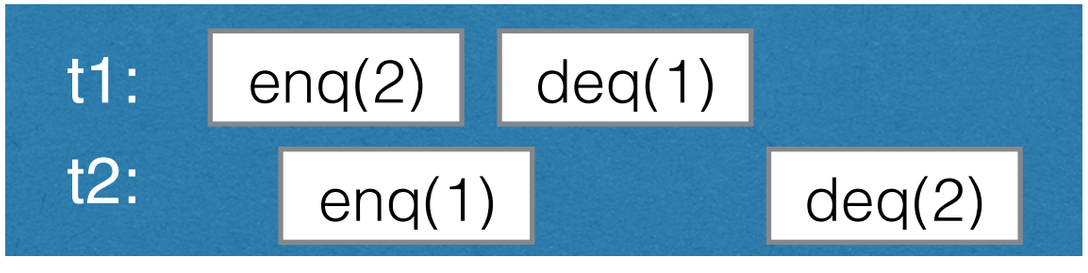
- Pool unordered



Performance and scalability



Semantics of concurrent data structures



e.g. queues

- Sequential specification = set of legal sequences

e.g. queue legal sequence
enq(1)enq(2)deq(1)deq(2)

- Consistency condition = e.g. linearizability / sequential consistency

e.g. the concurrent history above is a linearizable queue concurrent history

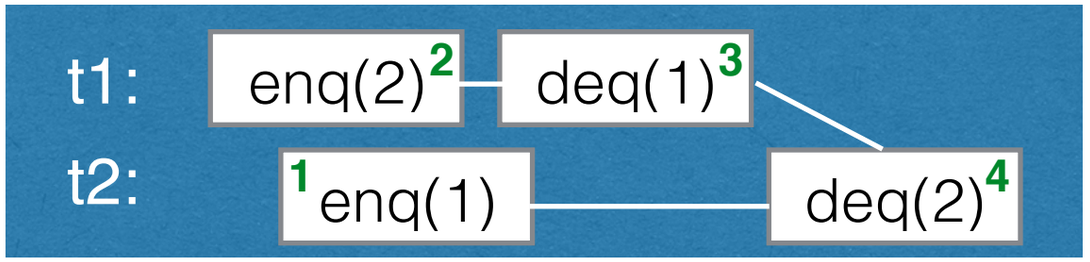
Consistency conditions

A history is … wrt a sequential specification iff

there exists a legal sequence that preserves precedence order

Linearizability [Herlihy,Wing '90]

consistency is about extending partial orders to total orders



Sequential Consistency [Lamport'79]

there exists a legal sequence that preserves per-thread precedence (program order)