

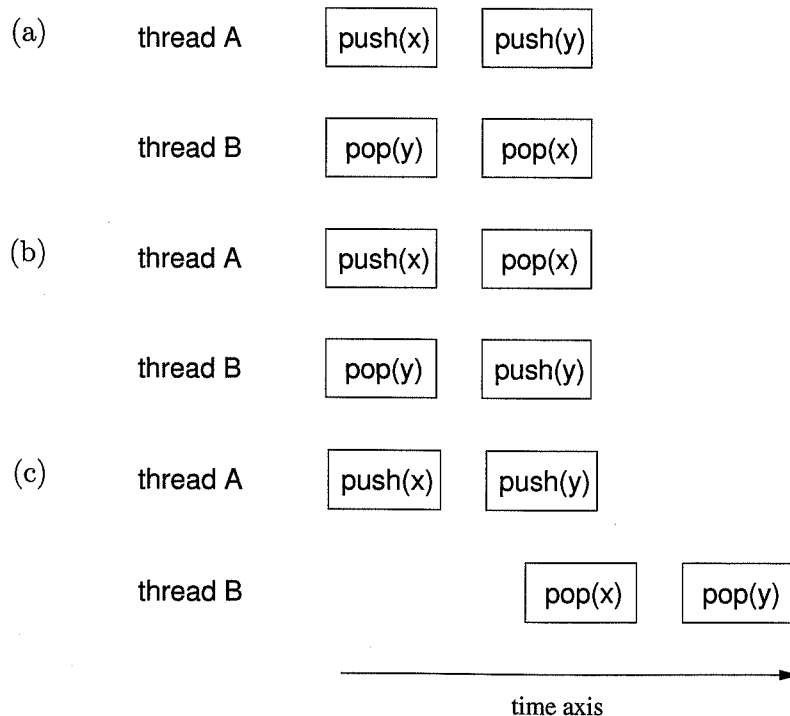
Exam Concurrency & Multithreading

Free University Amsterdam, 25 June 2010, 15:15-18:00

(At this exam, you may use copies of the slides without handwritten comments. Answers can be given in English or Dutch. Use of textbook or laptop is not allowed.)

(The exercises in this exam sum up to 90 points; each student gets 10 points bonus.)

1. Consider the stack data structure, with the standard `push()` and `pop()` methods. Explain for each of the following three executions, by two threads, whether it is linearizable and/or sequentially consistent. (12 pts)



2. Suppose we augment the FIFO queue class with a `peek()` method that returns but does not remove the first element in the queue. Show that this augmented queue has infinite consensus number. (10 pts)

3. Suppose that in the CLH queue lock, a thread would reuse its own node (instead of the node of its predecessor). Give a scenario to show that then the algorithm would be flawed. (10 pts)

4. A savings account object holds a nonnegative balance, and provides `deposit(k)` and `withdraw(k)` methods:

- `deposit(k)` adds `k` to the balance.
- `withdraw(k)` subtracts `k` if the balance is at least `k`, and otherwise blocks until the balance becomes `k` or greater.

Give a (Java pseudocode) implementation of this savings account using locks and conditions. (14 pts)

5. Describe how the `contains()` method works in case of the fine-grained linked-list implementation of sets. Show that your `contains()` method is linearizable (give linearization points, and consider the interplay with the `add()` and `remove()` methods). (12 pts)

6. Consider the work-stealing bounded queue. Give a scenario which shows that if `top` were an integer (instead of `AtomicStampedReference<integer>`), the algorithm would be flawed.

Also show that with `top` an `AtomicStampedReference<integer>`, your scenario does not lead to problems. (12 pts)

7. Suppose that in the static tree barrier, the node at the root would invert the global sense field before resetting the counter. Give a small example (a binary tree of depth one) to show that then the barrier would be flawed. (10 pts)

8. Give a (Java pseudocode) implementation of the `dequeue` method for the bounded transactional FIFO queue. (10 pts)