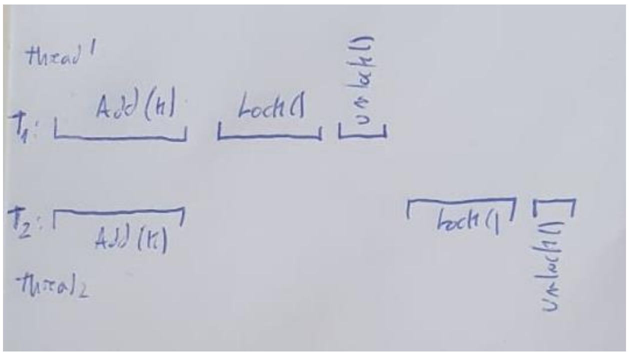
Concurrent Algorithms and Data Structures – Theory Assignment 2-3

Bernat Xandri Zaragoza

1. Problem 1:

No, the algorithm is not linearizable with respect to the mentioned guarantees.



In this scenario, both threads manage to add k to the list, so even though threads can effectively add new elements to the list, these elements do not fulfil the two guarantees mentioned in the exercise, since k is now duplicated in the list.

2. Problem 2:

Yes, in this case, the resulting algorithm is linearizable.
The linearization policy would be as follow:
For the enqueue part, the linearization point would be in the execution of the CAS on line 7. This is the point in the code where the new node is actually added to the queue. This CAS, because of it being an atomic operation, ensures that only one thread is adding the new node at the end of the list, making it linearizable.
For the dequeue part, the linearization point would be the CAS on line 13. This is where the code updated the head of the queue, dequeuing this way the element. In this case, like in the previous one, the CAS ensures that only one thread is accessing the queue, making it linearizable.