# TITLE HERE

Mathias Pedersen, 201808137

Master's Thesis, Computer Science
February 2024
Advisor: Amin Timany

AARHUS
UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

# Abstract

**►in English…◄**

# Resumé

**►in Danish. . . ◄**

# Acknowledgments

►. . . ◄

*Mathias Pedersen*
*Aarhus, February 2024.*

# Contents

# Chapter 1

# Introduction

**►motivate and explain the problem to be addressed◄**

**►example of a citation: [1]◄ ►get your bibtex entries from `https://dblp.org/`◄**

# Chapter 2

# The Great Ideas

Two-Lock Michael-Scott Queue discussion (The need for $xs_{old}$):

As mentioned in the insights, it is possible for the tail to lag one node behind the head. This insight lead to including the old nodes of the queue in the queue invariant. This addition manifests in the end of the proof of dequeue. When we open the invariant to swing the head to the new node, we get that the entire queue is $xs$. After performing the write, we can then close the invariant with the same $xs$ that we opened the queue to (just written differently to signify that $x_{head}$ is now "old"). Because of this, we can supply the same predicate concerning the $tail$ (the or) that we opened the queue with, since these only mention $xs$, which remains the same.

Had we not used an $xs_{old}$ and essentially just "forgotten" old nodes in the queue, we couldn't have done this. Say that we defined $xs$ as $xs = x_{head} :: xs_{rest}$ instead. Then, once we have to close the invariant, we cannot supply the $xs$, which we got when we opened the invariant. Our only choice (due to the fact that $head$ must point to $x_{n_head}$) is to close the invariant with $xs' = xs_{rest} = x_{n_head} :: xs''_{n_rest}$. However, clearly $xs' \neq xs$, so we cannot supply the same predicate concerning the $tail$ (the or) that we got when opening the invariant, since this predicate talks about $xs$, not $xs'$. Now, if we opened the invariant in the Dequeue case, then we could assert that $lastxs = lastxs'$, and hence still be able close the invariant. However, if we opened the invariant in the Both case, then we would need to assert that $2lastxs = 2lastxs'$. This is however not provable, since it might be the case that $xs''_{n_rest}$ is empty, and hence $2lastxs'$ is $None$, whereas $2lastxs = x_{n_head}$.

## Chapter 3

# Conclusion

**▸conclude on the problem statement from the introduction◂**

# Bibliography

[1] Maged M. Michael and Michael L. Scott. Simple, fast, and practical non-blocking and blocking concurrent queue algorithms. In James E. Burns and Yoram Moses, editors, *Proceedings of the Fifteenth Annual ACM Symposium on Principles of Distributed Computing, Philadelphia, Pennsylvania, USA, May 23-26, 1996*, pages 267–275. ACM, 1996.

# Appendix A

# The Technical Details

►...◄