Haskell versus Rust: Benchmarking Efficient Concurrent Pathfinding in a Purely-Functional Programming Language

Prince Bernie B. Colis

Bachelor of Science in Computer Science

John Kenneth S. Lesaba

Bachelor of Science in Computer Science

Jon Ariel N. Maravilla

Bachelor of Science in Computer Science

Karl Frederick R. Roldan

Bachelor of Science in Computer Science

Senior project submitted to the faculty of the
Department of Computer Science
College of Computer Studies, Ateneo de Naga University
in partial fulfillment of the requirements for their respective
Bachelor of Science degrees

Project Advisor: Adrian Leo Pajarillo
First Panel Member

Second Panel Member

Third Panel Member

Month Day, 2020

Naga City, Philippines

Keywords: parallel programming, functional programming, graph theory

The Senior Project entitled

Haskell versus Rust: Benchmarking Efficient Concurrent Pathfinding in a Purely-Functional Programming Language

developed by

Prince Bernie B. Colis
Bachelor of Science in Computer Science

John Kenneth S. Lesaba
Bachelor of Science in Computer Science

Jon Ariel N. Maravilla
Bachelor of Science in Computer Science

Karl Frederick R. Roldan
Bachelor of Science in Computer Science

and submitted in partial fulfillment of the requirements of their respective Bachelor of Science degrees has been rigorously examined and recommended for approval and acceptance.

First Panel Member
Panel Member
Date signed: ________________________

Second Panel Member
Panel Member
Date signed: ________________________

Third Panel Member
Panel Member
Date signed: ________________________

Adrian Leo Pajarillo
Project Advisor
Date signed: ________________________

The Senior Project entitled

Haskell versus Rust: Benchmarking Efficient Concurrent Pathfinding in a Purely-Functional Programming Language

developed by

Prince Bernie B. Colis
Bachelor of Science in Computer Science

John Kenneth S. Lesaba
Bachelor of Science in Computer Science

Jon Ariel N. Maravilla
Bachelor of Science in Computer Science

Karl Frederick R. Roldan
Bachelor of Science in Computer Science

and submitted in partial fulfillment of the requirements of their respective Bachelor of Science degrees is hereby approved and accepted by the Department of Computer Science, College of Computer Studies, Ateneo de Naga University.

Marianne P. Ang, MS
Chair, Department of Computer Science
Date signed: ________________________

Joshua C. Martinez, MIT
Dean, College of Computer Studies
Date signed: ________________________

Declaration of Original Work

We declare that the Senior Project entitled

Haskell versus Rust: Benchmarking Efficient Concurrent Pathfinding in a Purely-Functional Programming Language

which we submitted to the faculty of the

Department of Computer Science, Ateneo de Naga University

is our own work. To the best of our knowledge, it does not contain materials published or written by another person, except where due citation and acknowledgement is made in our senior project documentation. The contributions of other people whom we worked with to complete this senior project are explicitly cited and acknowledged in our senior project documentation.

We also declare that the intellectual content of this senior project is the product of our own work. We conceptualized, designed, encoded, and debugged the source code of the core programs in our senior project. The source code of third party APIs and library functions used in my program are explicitly cited and acknowledged in our senior project documentation. Also duly acknowledged are the assistance of others in minor details of editing and reproduction of the documentation.

In our honor, we declare that we did not pass off as our own the work done by another person. We are the only persons who encoded the source code of our software. We understand that we may get a failing mark if the source code of our program is in fact the work of another person.

Prince Bernie B. Colis
3 - Bachelor of Science in Computer Science

John Kenneth S. Lesaba
3 - Bachelor of Science in Computer Science

Jon Ariel N. Maravilla
3 - Bachelor of Science in Computer Science

Karl Frederick R. Roldan
3 - Bachelor of Science in Computer Science

This declaration is witnessed by:

Adrian Leo Pajarillo
Project Advisor

Haskell versus Rust: Benchmarking Efficient Concurrent Pathfinding in a Purely-Functional Programming Language

by

Prince Bernie B. Colis, John Kenneth S. Lesaba, Jon Ariel N. Maravilla, and Karl Frederick R. Roldan

Project Advisor: Adrian Leo Pajarillo

Department of Computer Science

EXECUTIVE SUMMARY

To be filled in later. `/*TODO*/`.

I dedicate this research work to all of humanity.

ACKNOWLEDGEMENTS

I thank everyone who helped me finish this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

Functional programming has suddenly risen to popularity with examples that include Scala, Clojure, ReactJS, and other languages adopting lambda expressions. Reasoning about program correctness in a pure function can be done in a dependently-typed, proof assistant such as Coq or Agda[2][8][3]. Likewise, pure functions can be easily proved by using induction. Composing two proven functions into a single function should also give the correct result[1].

This research aims to utilize the existing A* pathfinding algorithm[4][9] and find a way to develop a reasonably-efficient purely functional implementation of the algorithm using parallel data structures such as STMs or MVars[6]. The main objective of the research is to find an efficient concurrent implementation of a maze solver in a purely functional programming environment with comparable performance and space complexity of a performant imperative programming language.

1.1 Scope and Limitations

The researchers will utilize Haskell for concrete implementation of the parallel A* algorithm in a functional programming environment. Likewise, for performance comparison, the researchers will use the Rust programming language due to some of its features having similarities with Haskell such as correct concurrent programs[7] and guarantees a relatively safe program[5].

Other purely-functional languages or lambda notation for generality will not be used. Other concurrent data structures besides `MVar` and Software Transactional Memory will not be utilized.

Appendix A

Code Listing

REFERENCES

[1] A. Abel, M. Benke, A. Bove, J. Hughes, and U. Norell, *Verifying haskell programs using constructive type theory*, 01 2005, pp. 62–73.

[2] J. Breitner, A. Spector-Zabusky, Y. Li, C. Rizkallah, J. Wiegley, and S. Weirich, *Ready, set, verify! applying hs-to-coq to real-world haskell code*, 2018.

[3] Y. El Bakouny, T. Crolard, and D. Mezher, *A coq-based synthesis of scala programs which are correct-by-construction*, Proceedings of the 19th Workshop on Formal Techniques for Java-like Programs, (2017).

[4] Z. et al., *Parallelizing a* path finding algorithm*, International Journal Of Engineering And Computer Science, 6 (2017), pp. 22469–22476.

[5] R. Jung, J.-H. Jourdan, R. Krebbers, and D. Dreyer, *Rustbelt: Securing the foundations of the rust programming language*, Proc. ACM Program. Lang., 2 (2017).

[6] S. Marlow, *Parallel and Concurrent Programming in Haskell*, O'Reilly Media, Inc., 2013.

[7] A. Saligrama, A. Shen, and J. Gjengset, *A practical analysis of rust's concurrency story*, 2019.

[8] A. Spector-Zabusky, J. Breitner, C. Rizkallah, and S. Weirich, *Total haskell is reasonable coq*, Proceedings of the 7th ACM SIGPLAN International Conference on Certified Programs and Proofs, (2018).

[9] A. Weinstock and R. Holladay, *Parallel a* graph search*.

VITA

`/*TODO*/` are BS Computer Science student of the Department of Computer Science at the Ateneo de Naga University.