Parallel Implementation of the A* Pathfinding Algorithm on 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

Parallel Implementation of the A* Pathfinding Algorithm on 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

Parallel Implementation of the A* Pathfinding Algorithm on 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

Parallel Implementation of the A* Pathfinding Algorithm on 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

Parallel Implementation of the A* Pathfinding Algorithm on 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

Pathfinding algorithms are methods of finding a path between two vertices in a graph. Most pathfinding problems are concerned with finding the shortest path between two vertices, if there are more than one possible paths. There had been many pathfinding algorithms that had been developed throughout the years such as Minimum Spanning Tree (MST), Prim's Algorithm[16], and the A* algorithm, which will be used in this paper.[11]

Functional programming is one of the major programming paradigms where computations are done by function composition. Along with this, purely-functional programming is a subparadigm of functional programming where there are no side-effects (e.g, variable mutability). One of the major challenges of parallel programming is controlling the order of execution to prevent *race conditions*, which can often lead to bugs and are hard to maintain. However, since purely-functional programming languages such as Haskell[1] have no mutability and computations lead to the same result regardless of the order, they are a perfect candidate for writing parallel programs.[10] This research aims to find a parallel implementation of the existing A* pathfinding algorithm using a purely-functional setting with attention to program performance in terms of time and space complexity. [9, 19] In turn, this helps in the advancement of different programming languages that feature functional programming and lambda expressions when it comes to purely-functional algorithms and data structures.

1.1 Project Context

The A* pathfinding algorithm is now mostly used as a pathfinding algorithm for video games. While most video games are written in an imperative and object-oriented programming languages such as C#, C++, and JavaScript, it is entirely possible to write video games in functional programming languages using a reactive functional programming paradigm. [6]

Functional programming had been getting more popular recently with more people using ReactJS, TypeScript, PureScript, and Scala. However, imperative programming still dominates the industry, thus more algorithms are written for imperative programs. It can be observed that most algorithm books are written with imperative programming in mind such as *Introduction to Algorithms*[7], *The Algorithm Design Manual*[17], and *The Art of Computer Programming*[12]. Thus, the need for familiarity for functional approaches for some of the most popular algorithms should be discovered, since people are more familiar with imperative approaches and as such, it is moreoften used in developing video games than functional programming.

One reason for writing programs in a functional language is that pure functions are easy to reason about and can especially be aided with using a dependently-typed proof assistant such as Coq or Agda.[4, 18, 8]. Programs written in functional programming can be easily proven out by reasoning about the smaller components and composing two or more proven functions into a single function and it should also give the correct result with respect to the input, provided that the algorithm is correct. [3]

1.2 Purpose and Description

This research aims to utilize the existing A* pathfinding algorithm [9, 19] and find a way to develop a reasonably-efficient purely-functional implementation of the algorithm using parallel data structures such as STMs or MVars[13].

The A* Pathfinding algorithm is used heavily in video games, telephone traffic, and other graph traversal problems[11]. This research aims to aid in the development of video games using the functional programming paradigm in the future as video game development is dominated by imperative languages.

1.3 Objectives

The research aims to find an efficient parallel implementation of the A* pathfinding algorithm using a purely-functional programming language such as Haskell. Likewise, concrete comparisons between the number of cores and logical threads will be used to measure the most efficient performance runtime of the algorithm.

The researchers aim to create two programs that will be used in the research. A generator program that will generate an arbitrary-sized maze that will be relatively hard to solve without the aid of computers in a short amount of time.[5] And a solver program will be written for translating the output of the maze generator to an algebraic graph that the A* algorithm can understand. The solver program will be a browser-based application that will show the maze graphically and the generated path when the algorithm terminates. This pathfinding program must have an efficient performance while solving the maze. To measure the performance of the program, the application ThreadScope will be used to monitor the thread and core activities while the program is being run. [2]

1.4 Scope and Limitations

The research will only cover solvable-mazes as the A* algorithm does not halt when there is no reachable end goal (e.g, the start vertex and end vertex lie on different components of the graph).[11] Likewise, there will be no generality and all programs will be written in Haskell. Translation to other functional programming languages is not a priority and, thus, lambda notation will not be used. Other concurrent data structures besides MVar and Software Trasnactional Memory will not be utilized. The implementation of the graph that the research will use will be Algebraic Graphs.[14]

The concrete implementation and analysis is planned to be tested only on four CPUs such as Intel Core i7-9750H and AMD Ryzen 5 3500x. Other CPU architectures are not planned to be tested on.

Appendix A

Code Listing

REFERENCES

[1] *Haskell programming language*. `https://haskell.org`.

[2] *Threadscope*. `https://github.com/haskell/ThreadScope`.

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

[4] 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.

[5] J. Buck, *Mazes for Programmers*, The Pragmatic Programmers, 2015.

[6] M. H. Cheong, *Functional programming and 3d games*, (2006).

[7] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, Third Edition*, The MIT Press, 3rd ed., 2009.

[8] 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).

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

[10] K. Hammond, *Why parallel functional programming matters: Panel statement*, in Reliable Software Technologies - Ada-Europe 2011, A. Romanovsky and T. Vardanega, eds., Berlin, Heidelberg, 2011, Springer Berlin Heidelberg, pp. 201–205.

[11] P. E. Hart, N. J. Nillson, and R. Betram, *A formal basis for the heuristic determination of minimum cost paths*, IEEE Transactions of Systems Science and Cybernetics, SSC-4 (1968).

[12] D. E. Knuth, *The Art of Computer Programming, Volume 1 (3rd Ed.): Fundamental Algorithms*, Addison Wesley Longman Publishing Co., Inc., USA, 1997.

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

[14] A. Mokhov, *Algebraic graphs with class (functional pearl)*, SIGPLAN Not., 52 (2017), p. 2–13.

[15] ——, *Algebraic graphs with class (functional pearl)*, in Proceedings of the 10th ACM SIGPLAN International Symposium on Haskell, Haskell 2017, New York, NY, USA, 2017, Association for Computing Machinery, p. 2–13.

[16] R. C. Prim, *Shortest Connection Networks And Some Generalizations*, Bell System Technical Journal, 36 (1957), pp. 1389–1401.

[17] S. S. Skiena, *The Algorithm Design Manual*, Springer, London, 2008.

[18] 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).

[19] 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.