Project Proposal

Liziqiu Yang

July 2023

In this project, our chosen research paper is [1]. The paper presents a pioneering approach that leverages functional programming to implement graph algorithms. Building upon the insights from this paper, our project will adopt Haskell as the programming language for our implementations. Our primary goal is to replicate and validate the results reported in the paper. By doing so, we aim to explore the effectiveness and efficiency of functional programming in solving graph-related problems. Through this replication, we hope to gain a deeper understanding of the applicability of functional programming paradigms in graph algorithm development. Additionally, we will investigate potential areas of improvement and conduct comparative analyses with other programming paradigms to assess the strengths and weaknesses of functional programming in the context of graph algorithms. our project will use Haskell as the programming language for implementation. We plan to replicate the following results:

1. Simplify the abstract graph classes proposed by the author (since this paper was published in 2001, it has a history of 20 years, and language development allows us to simplify).

2. Establish connections between types of graphs and functors/monads.

3. Compare the time complexity of algorithms with the time complexity of imperative languages.

Implementation process and results during the final demonstration. Additionally, we will submit the project's source code and various analyses, such as time complexity and type definitions.

In the presentation, we will provide a detailed overview of the project's background and objectives, explaining the functional programming approach employed in the referenced paper and the rationale behind choosing Haskell as our implementation language. We will demonstrate how we replicated and validated the results, showcasing the effectiveness and performance of functional programming in different graph algorithms. Emphasizing the optimization strategies and design decisions employed during the implementation process will also be part of the presentation.Alongside the PPT, we will submit the project's source code to allow the reviewers and others to explore the implementation details. We will ensure the code is well-documented and annotated to facilitate comprehension.Furthermore, we will provide analysis reports as part of the submission, including time complexity evaluations and function type definitions. These analyses will elucidate the advantages of functional programming in graph algorithms and identify potential areas for improvement. Whenever possible, we will utilize charts and graphs to present the results in a visually intuitive manner.

References

[1] M. Erwig, "Inductive graphs and functional graph algorithms," *J. Funct. Program.*, vol. 11, no. 5, p. 467–492, sep 2001. [Online]. Available: https://doi.org/10.1017/S0956796801004075