UNIVERSITY
OF TECHNOLOGY
SYDNEY

Improving the Efficiency of Graph-Based Static Analysis

by Yuxiang Lei

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy (Software Engineering)

under the supervision of Dr Yulei Sui

University of Technology Sydney
Faculty of Engineering and Information Technology

12/2022

Improving the Efficiency of Graph-Based Static Analysis

A thesis submitted in fulfilment of the requirements
for the degree of

Doctor of Philosophy
in
Software Engineering

by

Yuxiang Lei

to

School of Computer Science
Faculty of Engineering and Information Technology
University of Technology Sydney
NSW - 2007, Australia

December 2022

Certificate of Original Authorship

I, *Yuxiang Lei* declare that this thesis is submitted in fulfilment of the requirements for the award of *Doctor of Philosophy*, in the *School of Computer Science, Faculty of Engineering and Information Technology* at the University of Technology Sydney. This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis. This document has not been submitted for qualifications at any other academic institution. This research is supported by the Australian Government Research Training Program.

Production Note:
Signature removed prior to publication.

SIGNATURE: ______________________________

DATE: 30^{th} December, 2022

PLACE: Sydney, Australia

Abstract

Generally speaking, static program analysis is to figure out whether a program can do whatever the program designers want it to do without actually executing the program. From different perspectives, static analysis studies various properties of a program, including correctness, robustness, liveness, safety and efficiency. As contemporary programs usually tend to be large and complex, developing efficient automatic program analysis techniques while maintaining soundness and precision is desirable.

Transitivity extensively manifests in the executions of programs, where controls and data are propagated and processed via flows. Taking data flow as an example, two assignment instructions `a = b` and `b = c` imply a result `a = c`, which means that the value of `c` flows into `a` via `b`. Static analyses inevitably include the analysis of flows, which is usually conducted in the form of solving dynamic transitive closure on the abstract graph of programs. The inefficiency arises from not only the high complexity of transitive closure itself but also the high redundancies of the analysis techniques.

This dissertation studies improving the efficiency of dynamic transitive closures on graph-based static analysis. Specifically, it focuses on improving the efficiencies of three popular static analysis frameworks: context-free language reachability, recursive state machine reachability and set constraint analysis. All the three frameworks are under the scope of graph analytics. Namely, all the analyses operate on an abstract graph of the target program.

In this dissertation, the methodologies focus more on eliminating redun-

dancy rather than theoretically lowering complexity.

For transitive redundancy that arises from the massive re-computations and re-derivations during the analysis procedures, we design a hybrid data structure and apply it to context-free language reachability. Based on this, we propose a partially ordered algorithm, which significantly improves the scalability of context-free language reachability analysis by eliminating most re-computations and re-derivations.

For trivial nodes and edges in the abstract graphs of programs, which cause extra computations in the analysis procedure, we develop a graph folding technique to remove redundant nodes and edges in the preprocessing stage and apply it to recursive state machine reachability. The graph folding technique extends the applicability of some existing techniques from particular scenarios to general analysis as long as the recursive state machine is given and is well compatible with other preprocessing techniques.

For set constraint analyses where the graph contains weighted edges, we discover the derivation equivalence property and propose an approach that avoids the infinite iterations caused by weighted cycles during constraint solving. The derivation equivalence based constraint solving is highly efficient while maintaining the precision.

Notably, the three dynamic transitive closure based program analysis frameworks, i.e., context-free language reachability, recursive state machine reachability and set constraint solving, are generally recognized as interconvertible. Accordingly, the three techniques proposed in this dissertation are mutually compatible. The empirical study on real-world clients, including value-flow analysis, alias analysis and pointer analysis, shows that our approaches are practical and effective.

Acknowledgments

First, I would like to express my sincere gratitude to my supervisor, Prof. Yulei Sui, for presenting me with a chance to study at the University of Technology Sydney as a sponsored student. He told me how to improve my communication, writing and programming skills. His wisdom, dedication, scrupulousness and patience guided me throughout my PhD candidature and continuously illuminated my future research.

Besides, I would like to thank Prof. Qirun Zhang and Prof. Shinhwei Tan. Their deep insights and sweet suggestions inspired me when I got lost in my research topics.

Moreover, I would like to thank my colleagues Dr. Pei Xu, Dr. Guanqin Zhang and Dr. Xiao Cheng, who helped me with my school issues when I was on leave of absence and not in Sydney due to COVID-19. I would also like to thank Dr. Chandranath Adak, who provided this beautiful thesis template.

My greatest thankfulness is owed to my parents. They brought me up, sharing knowledge with me and supporting me to pursue a higher education degree from material and spiritual tiers. Without them, I would never have had a chance to study and meet friendly people at UTS. When I was frustrated, their encouragements were always the most powerful motivation to me.

Table of Contents

List of Publications

The following is a list of publications that are included in this thesis.

Chapter 3:

Yuxiang Lei, Yulei Sui, Shuo Ding, and Qirun Zhang. "Taming transitive redundancy for context-free language reachability." *Proceedings of the ACM on Programming Languages 6. OOPSLA2 (2022): 1556-1582.* (SIPLAN 2022 Distinguished Artifact Award)

Chapter 4:

Yuxiang Lei, Yulei Sui, Qirun Zhang, and Shinhwei Tan. "Recursive State Machine Guided Graph Folding for Context-Free Language Reachability." *Proceedings of the ACM on Programming Languages 7. PLDI (2023).*

Chapter 5:

Yuxiang Lei and Yulei Sui. "Fast and Precise Handling of Positive Weight Cycles for Field-sensitive Pointer Analysis." *Static Analysis: 26th International Symposium, SAS (2019), Proceedings 26 (pp. 27-47).* (Radhia Cousot Young Researcher Best Paper Award)

List of Figures

Figure **Page**

List of Tables

Table **Page**