BLACK-BOX EQUIVALENCE CHECKING ACROSS COMPILER TRANSFORMATIONS

MANJEET DAHIYA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY DELHI
OCTOBER 2018

BLACK-BOX EQUIVALENCE CHECKING ACROSS COMPILER TRANSFORMATIONS

by

MANJEET DAHIYA

Department of Computer Science and Engineering

Submitted

in fulfillment of the requirements of the degree of Doctor of Philosophy

to the

Indian Institute of Technology Delhi
October 2018

Certificate

This is to certify that the thesis titled **Black-box Equivalence Checking across Compiler Transformations** being submitted by **Manjeet Dahiya** for the award of **Doctor of Philosophy** in Computer Science and Engineering is a record of bona fide work carried out by him under my guidance and supervision at the Department of Computer Science and Engineering, Indian Institute of Technology Delhi. The work presented in this thesis has not been submitted elsewhere, either in part or full, for the award of any other degree or diploma.

Sorav Bansal
Associate Professor
Department of Computer Science
and Engineering
Indian Institute of Technology Delhi
New Delhi, India 110 016

Acknowledgments

At the outset, I would like to wholeheartedly thank my advisor Sorav Bansal for guiding me on how to choose the problems to work on, how to write papers, and how to present them. Most importantly, he instilled in me the approach of doing principled research, which I would carry for the rest of my career. I am grateful to my PhD committee members Sanjiva Prasad, Akash Lal and Sandeep Sen for their guidance and evaluation of my work on a regular basis. I am thankful to S. Arun Kumar and Sanjiva Prasad for floating some of the best courses during the period of my PhD, which also helped me in my research.

I would like to thank Deepak and Piyus with whom I had valuable discussions on various topics. I would also like to extend my thanks to the staff of CSE and SIT departments. I also sincerely thank TCS Research for granting me the scholarship during my PhD.

On a personal note, during my PhD, my wife and I were blessed with our cute little children Vaanya and Uday — I am thankful to God for his grace. I am grateful to my parents and family for their unending love and support. I also would like to remember my friends who have encouraged me in my pursuit, and humoured me when needed. Finally, I am grateful to my wife Ruchi because of whom I was able to take the huge step of quitting my job to pursue PhD, and these words could not have been ever written, otherwise. This thesis is dedicated to my loving wife Ruchi.

Manjeet Dahiya

Abstract

Equivalence checking is an important building block for program synthesis and verification. Design of an equivalence checker is dependent on the application; program synthesis tools like superoptimizers demand that the underlying equivalence checker should perform the required equivalence checks in a *black-box* manner, i.e., without requiring the knowledge of the individual constituent transformation passes. This thesis presents techniques for black-box equivalence checking across compiler optimizations and across power environments.

In the first part, we present a technique for black-box equivalence checking across compiler optimizations. Unlike previous work on translation validation, our technique works across multiple composed transformations and does not employ a pass-by-pass approach. Our technique supports undefined behaviour related optimizations, and we are the first to handle undefined behaviour related optimizations in equivalence checking for programs with loops. We test our checker with the optimizations produced by multiple modern compilers, and our results are comparable to that of previous work on translation validation, albeit in a black-box setting.

In the second part, we discuss equivalence checking across power environments. Intermittent programs, which are transiently powered, keep checkpointing the program state to a persistent memory, and on power failures, the programs resume from the last executed checkpoint. An intermittent program is usually automatically generated by instrumenting a given continuous program (continuously powered). The behaviour of the continuous program should be equivalent to that of the intermittent program under all possible power failures. We present a technique to automatically verify the correctness of an intermittent program with respect to its continuous counterpart. We present a model of intermittence to capture all possible scenarios of power failures and an algorithm to automatically find a proof of equivalence between a continuous and an intermittent program.

सारांश

ब्लैक-बॉक्स इक्विवैलेन्स चेकिंग अक्रॉस कम्पाइलर ट्रांस्फ़ॉर्मेशन्स

प्रोग्राम सिंथेसिस और वेरिफिकेशन के लिए, समानता की जाँच (इक्विवैलेन्स चेकिंग) एक महत्वपूर्ण निर्माण खंड है। एक समानता परीक्षक (इक्विवैलेन्स चैकेर) का डिजाइन उसके प्रयोग पर निर्भर है; प्रोग्राम सिंथेसिस उपकरण जैसे सुपरऑप्टिमिज़ेर कि मांग है कि अंतर्निहित इक्विवैलेन्स चैकेर ब्लैक-बॉक्स जांच करे। ब्लैक-बॉक्स इक्विवैलेन्स चैकेर में आवश्यक है कि इसके जांच के तरीके को ट्रांसफॉर्मेशन पास्सेस के ज्ञान कि जरूरत ना पड़े। यह थीसिस, कम्पाइलर ऑप्टिमाईजेंशंस और पावर एनवाईरमेंट्स के ब्लैक-बॉक्स समानता जांच के लिये तकनीकें प्रस्तुत करती है

पहले भाग में, हम कम्पाइलर ऑप्टिमाईजेंशंस कि ब्लैक-बॉक्स इक्विवैलेन्स चेकिंग कि तकनीक पेश करते है।ट्रांसलेशन वेलिडेशन पर पिछले काम के विपरीत, हमारी तकनीक कई रचनाकृत परिवर्तनों (मल्टीप्ल कंपोज्ड ट्रांस्फ़ॉर्मेशन्स) पर काम करती है और पास-दर-पास दृष्टिकोण को नियोजित नहीं करती है। हमारी तकनीक अपरिभाषित व्यवहार (अन्डेफाइण्ड बिहेवियर) से संबंधित ऑप्टिमाईजेंशंस कि जाँच करती है। इस थीसिस ने अनिर्धारित व्यवहार से संबंधित ऑप्टिमाईजेंशंस व लूप के साथ प्रोग्राम्स के लिए समानता जांच में पहल करी है। हम कई आधुनिक कंपाइलर्स द्वारा उत्पादित ऑप्टिमाईजेंशंस के साथ हमारे इक्विवैलेन्स चैकेर का परीक्षण करते हैं, और हमारे परिणाम पिछले काम के तुलनीय है, हालांकि एक ब्लैक-बॉक्स सेटिंग में।

दूसरे भाग में, हम पावर एनवाईरमेंट्स कि समानता जांच पर चर्चा करते हैं। इंटरमिटेंट प्रोग्राम्स, जोकी प्रोग्राम स्थिति को एक परसिस्टेंट मेमोरी में चेंकपॉइंट करता है ताकि बिजली विफलताओं (पावर फेल्योर्स) में प्रोग्राम को फिर से शुरू कर सके। इंटरमिटेंट प्रोग्राम आमतौर पर स्वचालित रूप से कंटीन्यूअस प्रोग्राम (कन्टिन्यूसली पॉवर्ड) कि इंस्ट्रूमेंटेशन से उत्पन्न होता है। सभी संभावित बिजली विफलताओं के तहत, कंटीन्यूअस प्रोग्राम का व्यवहार इंटरमिटेंट प्रोग्राम के व्यवहार के समान होना चाहिए। हम स्वचालित रूप से इसके करेक्टनेस को सत्यापित करने कि तकनीक प्रस्तुत करते हैं। हम इंटरमिटेंस का एक मॉडल और इंटरमिटेंट प्रोग्राम्स व कंटीन्यूअस प्रोग्राम्स के बीच में समानता की जांच का अल्गोरिथम प्रस्तुस करते।

Contents

List of Figures

List of Tables

List of Algorithms