氏　　名	Mohamed LAMRAOUI
学位(専攻分野)	博士(情報学)
学 位 記 番 号	総研大甲第 1837 号
学位授与の日付	平成28年3月24日
学位授与の要件	複合科学研究科　情報学専攻 学位規則第6条第1項該当
学 位 論 文 題 目	Formula-based Fault Localization for Imperative Programs with Multiple Faults
論 文 審 査 委 員	主　査　教授　中島　震 教授　米田　友洋 教授　佐藤　一郎 准教授　岡野　浩三　信州大学 教授　細部　博史　法政大学

(別紙様式 2)
(Separate Form 2)

論文内容の要旨
Summary of thesis contents

Program debugging is a trial and error process of finding and eliminating bugs or defects in a computer program, and thus making it behave as expected. As software and hardware systems grow in complexity, debugging techniques for ensuring their correctness are increasingly important. Manual debugging is tedious, time-consuming and error-prone. Thus, making debugging automatically has been one of the major research topics in automated software engineering. Automatic formal verification of programs, such as the bounded model-checking (BMC) method, is useful for checking if a program exhibits erroneous behavior or not. Identifying root causes, which are the fundamental reasons for the occurrence of failing program executions, still involves manual inspection and therefore needs a vast amount of human efforts. This calls for a new method that performs fault localization automatically.

Automatic fault localization of imperative programs is a well-known problem and has been studied from various approaches. Back in the early 1980's, program slicing was introduced. A few years later model-based debugging (MBD) was presented. MBD combines the slicing method with the Reiter's model-based diagnosis theory framework. Thereafter, a work proposed to replace in MBD the algorithmic method for calculating program slices by a method following the Boolean satisfiability problem. In hardware debugging, more specifically in very-large-scale integration (VLSI) and system on chip (SoC) designs, an alternative method for localizing fault was shown effective. The method uses a debugging formulation based on maximum satisfiability (MaxSAT), which is a promising approach to the fault localization tool for imperative programs.

This thesis introduces and studies a new automatic fault localization method, which is formula-based fault localization for imperative programs written in ANSI C. The presented method combines the MBD with MaxSAT, specifically with partial maximum satisfiability. In contrast to other work on fault localization of imperative programs, we focus in this thesis on the localization of faults in multi-fault programs. Fault localization of multi-fault programs is a problem of great importance since real-world programs often have more than one fault. We demonstrate in this thesis that the fault localization of multi-fault programs requires further considerations to be successful. Dealing with multi-fault programs implies that faults may be spread in different program execution paths and that fault localization reports contain information from different faults. Therefore, it is required to use many program failing inputs in order to cover faults as much as possible. Since more than one failing execution is considered, it implies that the complexity of the problem increases and thus it is necessary to have an efficient method to localize all faults in an acceptable amount of time. Moreover, generated fault localization reports have to

be processed so that the software engineers spend less time in a posteriori root causes inspection. Here are the main contributions of this thesis. First, we reformulate the problem of formula-based fault localization systematically from a theoretical viewpoint. Second, we introduce new methods for encoding imperative programs into trace formulas. The way programs are encoded has a significant impact on the precision and efficiency of the root causes identification procedure. Third, we present an efficient method to calculate and combine root causes obtained from different failing executions. Fourth, all the methods are implemented in a tool, SNIPER. Several experiments are conducted on SNIPER to show the capabilities of the presented approach.

This thesis is organized as follows. Chapter 2 presents backgrounds of the fault localization of imperative programs. Chapter 3 introduces a series of concepts and terminologies that are needed to define the formula-based fault localization problem. These concepts and terminologies are used in the other chapters. Chapter 4 details the architecture of the tool SNIPER. The implementation of SNIPER, which is based on the LLVM compiler infrastructure and the Yices 1 partial maximum satisfiability solver, is detailed in Annex A. SNIPER is a basis on top of which we implement the different trace formulas presented in Chapters 5 and 7 and the algorithm of Chapter 6. In each of these chapters, we use SNIPER to empirically study the presented methods. Chapter 5 introduces a method for encoding programs, the full flow-sensitive trace formula (FFTF), which is equivalent to the control flow graph of the target program. The FFTF with appropriate algorithms is successful in localizing root causes in multi-fault programs for at least two of the benchmarks we used. However, although the FFTF is expressive, it is not efficient in view of computing time. In the Chapters 6 and 7 we present methods to deal with this problem. Chapter 6 presents a fault localization algorithm, which enumerates minimal correction subsets (MCS) in an incremental fashion. We show on a benchmark that the computing time can be reduced with this algorithm. Chapter 7 introduces an alternative method for encoding programs, the hardened flow-sensitive trace formula (HFTF). We empirically show on two benchmarks that the use of the HFTF, as compared to the FFTF, makes the fault localization algorithm produce less spurious root causes and perform faster. The HFTF is shown to be as expressive as the FFTF for most programs. Finally, Chapter 8 summarizes the contributions of this thesis and presents a list of future work on formula-based fault localization of imperative programs.

(別紙様式 3)
(Separate Form 3)

博士論文の審査結果の要旨
Summary of the results of the doctoral thesis screening

日常生活や産業界の活動を支える社会基盤にソフトウェアが浸透し，その不具合が社会に与える影響が大きくなっている．ソフトウェアの品質を確かめる方法として，プログラムを実行し不具合を検知するテスト技術や，数理論理の方法でプログラムの正しさを確認する形式検証技術が大きく進展した．これらの方法を用いることで，検査対象プログラムに欠陥（バグ）があるか否かを調べることが可能になる．ところが，その欠陥がプログラム中の何処に潜んでいるかを知ることは容易ではない．プログラム規模が大きくなると共に，欠陥箇所の発見にかかる作業が膨大になり開発コストの増大を招くこととなった．不具合の原因となった欠陥箇所を自動発見する技術に関して，学術的な基礎の確立が産業応用の観点から期待されている．

本博士論文は，数理論理に基づく科学的な方法を用いて，プログラム欠陥箇所の自動発見という工学的な問題の解決を目的とする．プログラム実行履歴ならびに満たすべき性質を数理論理の式で表現する．プログラムに不具合がある場合，この論理式全体が充足不能となる．ここで，全体論理式の真偽値を偽とする原因部分式（充足不能コアと呼ぶ）の中に，求める欠陥箇所が存在するという考え方を導入する．特に本研究では，充足不能コア発見よりも高速化処理が期待できる，修正部分式の極小解を求める問題に帰着する．提案方式を実現した研究ツールを試作し,その有効性を適用実験によって確認するものである．

本博士論文は，全 8 章と付録からなる．第 1 章は，本研究の背景と論文の構成を紹介する．第 2 章では，プログラム欠陥箇所発見という問題への研究状況を述べ，本研究の着眼点を整理する．特に，従来の研究は，ひとつのプログラムが単一の欠陥のみを持つことを暗黙に想定していた．欠陥の互いの関係を整理し，複数の欠陥を持つプログラムに対する欠陥箇所自動発見という問題を明確に定義した．

第 3 章は，本研究の基礎となる数理論理の基本概念，数理論理のプログラム自動検証への応用ならびにプログラムの表現方法といった既存の研究成果を簡明に紹介する．

第 4 章では，試作ツールの構成技術要素を説明する．特に，修正部分式の極小解を探索する基本アルゴリズム，欠陥箇所が複数ある場合への拡張方式，について新しい方法を提案する．なお，付録に，試作ツールの内部実現方式を詳細に説明した．

第 5 章では，プログラムの制御フローグラフと等価な情報を含む実行履歴表現方法 FFTF を提案する．特に，ひとつのプログラムが複数欠陥を持つ場合であっても，欠陥箇所を見逃すことがないことを示した．研究用の標準ベンチマーク問題 TCAS に適用することで，既存研究よりも良い精度で欠陥箇所を自動発見することができた．

第 6 章では，修正部分式の探索を高速化する方法を提案し，第 5 章で用いたベンチマーク問題 TCAS へ適用することで，その有効性を実験的に示した．

第 7 章では，プログラム実行履歴表現の新しい方法 HFTF を導入する．第 5 章で提案した FFTF では，修正部分式探索の対象となる論理式が長大となり，欠陥箇所発見処理の実行効率が低下するという問題があった．一般に，プログラムが不具合を持つ場合であっても，テストに成功する実行履歴のみに関わる実行箇所は欠陥を含まないと見做せる．つまり，テスト実行に成功した時に得られる情報を用いれば，修正部分式探索対象を縮小することが可能になる．このアイデアを HFTF と呼ぶ方法で実現した．2 つのベンチマーク問題に適用することで，提案方法の有効性を確認した．

第 8 章では，まとめとして，欠陥箇所自動発見に関わる将来の研究課題を整理する．

(別紙様式 3)
(Separate Form 3)

本審査委員会では，上記の研究成果は，数理論理に基づく科学的な方法をプログラム欠陥箇所の自動発見という工学的な問題の解決に結びつけるものであり，ソフトウェア研究の学位として相応しい内容を持つと判断した．また，第 4 章から第 7 章の研究成果については，2 件の査読あり国際学会発表ならびに同予稿集掲載論文および 1 件の査読あり学術論文誌掲載論文がある．これらから，本博士論文の成果が，当該技術分野で高く評価されていることがわかる．

以上，発表及び質疑応答と博士論文原稿，研究成果の内容に基づき，審査委員会全員によって審査した結果，博士論文として十分な水準にあると認められた．