

# 学 位 論 文 要 旨

学位論文題目     A Study on Test Data Generation for Software Testing by Applying  
Program Nets  
プログラムネットを適用したソフトウェアテストのための  
テストデータ生成に関する研究

申請者氏名     呉 彪

Along with the development of information technology, more and more software products have been applied to most aspects of society and facilitated people's lives. Consequently, the problems of software quality and security have been paid more attention. Software testing is an important means to ensure software quality and reliability, which is to find bugs, defects, or errors in a software program and is indispensable for all software development since it is a critical element of software quality assurance and represents the final review of specification, design, and coding.

Path testing is an important measure of general software testing, which searches specific test input data that covers every possible path in the software program. Among testing activities, test data generation is one of the most intellectually demanding tasks and also one of the most critical ones, since it has a strong impact on the effectiveness and efficiency of the whole testing process. However, with the increasing of complexity and scale of software, a software program may contain an infinite number of paths when the program has loops. In addition, the number of paths is also exponential to the number of branches of the program. As a result, test input data generation takes much computational time and is an NP-complete problem, which makes input data generation more complex.

Graphs can capture complex data dependencies and be widely used to model and analyze data-flow programs. As a kind of graph, data-flow program net (program net or net for short) is an important way to study data-flow programs. Program net is specially tuned to data-flows through arithmetic and logical operations. This dissertation discusses how to apply program nets to generate test input data for approaching the software testing problem theoretically. This dissertation is organized as follows: Chapter 1 gives an introduction of the research background and presents the motivation and the target of this dissertation.

Chapter 2 presents the basic definitions of program nets and proposes a necessary extension of program nets so-called exhaustive program nets to describe the dynamic behaviour information of program nets. The known basic properties and the whole process of applying program nets to test input data generation are also given.

Chapter 3 presents a proposal to generate such required subnets' set that can cover all nodes of a given program net and gives its corresponding algorithms to (1) construct the layer net of the given program net, (2) construct the acyclic program net from the given program net with directed circuits, (3) obtain the initial subnet, and (4) generate the remaining subnets so that the obtained subnets set containing the initial subnet can cover all nodes of the given program net in order to finally find a specific test data used in software testing.

Chapter 4 presents a method to find a specific test input data for such subnets that may possess input data based on the analysis result of the behaviours of all the subnets. Firstly, a brief introduction is given to Satisfiability Modulo Theories (SMT for short) that can be used to find test input data. Then, discussion is done on how to equivalently transform the subnets obtained in Chapter 3 into exhaustive subnets. After that, an algorithm is designed to obtain all constraint conditions from a given exhaustive subnet. Finally, we introduce an SMT solver, Z3 Prover, which can find the specific test input data according to the constraint conditions.

Chapter 5 gives a case study that shows how to generate required subnets for three actual Java programs by using the algorithms proposed in Chapter 3, as well as to indicate how to find a set of test input data for the subnets by using the SMT solver introduced in Chapter 4. Also, the experimental results are shown and discussed for the Java programs.

Chapter 6 concludes the results obtained in this dissertation and discusses future research works remaining to be solved.

注)

1. 日本語または英語により作成してください。
2. 日本語の場合は2,000字程度, 英語の場合は800語程度で要旨を記入してください。

## 学位論文審査の概要と結果

報告番号	東アジア博 甲 第 147 号	氏 名	呉 彪
論文題目	A Study on Test Data Generation for Software Testing by Applying Program Nets		
(論文審査概要)			
<p>本学位論文は、ソフトウェアの品質や信頼性を検証するためのソフトウェア・テスト問題に対する効率よい解決法について論じたものである。ソフトウェアの制御とデータの両方のフローを図式で表現できるプログラムネットを用いることにより、規模の大きいプログラムを実行の流れを保持した部分プログラムへ分割することを可能とし、さらに、部分プログラムを検証するためのテスト用データ（テスト入力データ）の生成の効率化が図れる。これにより、膨大な処理時間（指数時間）を要するソフトウェア・テストの計算時間の大幅な短縮が期待できる。以上の考えに基づいて、本論文はソフトウェア・テスト問題の解決法を提案している。論文の構成は次の通りである。</p> <p>第1章では、ソフトウェアの品質と信頼性を検証することの重要性および検証に用いられる種々のソフトウェア・テストの方法を紹介したうえで、ソフトウェア・テスト問題に関する研究の現状および課題を明らかにしている。</p> <p>第2章では、本研究で用いるプログラムネットの基本的な定義を示し、プログラムネットの動的な情報の記述に必要なプログラムネットの拡張（Exhaustive Program Net）について論じている。また、既知の基本的なプログラムネットの性質と、プログラムネットを利用した入力データの生成に関するプロセス全体の考え方について説明している。第3章では、プログラムネットに含まれるすべてのノードをカバーするサブネットセットを生成するために、次の方法に関連する高速計算可能な多項式時間のアルゴリズムを提案している。</p> <ol style="list-style-type: none"> <li>(1) 特定のプログラムネットのレイヤーネットを構築する方法</li> <li>(2) 有向回路を含む特定のプログラムネットから非巡回プログラムネットを構築する方法</li> <li>(3) 初期サブネットを取得する方法</li> <li>(4) ソフトウェア・テストで使用される特定のテストデータを最終的に見つけるために、初期サブネットを含む取得されたサブネットセットが特定のプログラムネットのすべてのノードをカバーできるように、残りのサブネットを生成する方法</li> </ol> <p>第4章では、すべてのサブネットの動作の分析結果に基づいて、入力データをもつ可能性のあるサブネットの特定のテスト入力データを見つける方法を示している。最初に、テスト入力データを見つけるために Satisfiability Modulo Theories (SMT) について簡単に紹介し、次に第3章で得られたサブネットを等価な Exhaustive Subnet に変換する方法について説明し、Exhaustive Subnet に関する制約条件を導き出すための多項式時間のアルゴリズムを設計している。また、ソフトウェア実行の流れを制約する諸条件に基づいて特定のテスト入力データを見つけることのできる SMT ソルバーである Z3 Prover を紹介している。</p> <p>第5章では、第3章と第4章で提案された方法を活用したケーススタディを示している。まずは、第3章で提案したアルゴリズムを活用して3つの実際の Java プログラムを対象としたサブネットを生成し、次に第4章で導入された SMT ソルバーを使ってサブネットのテスト入力データのセットを見つける、といった処理手順を述べている。最後に、3つの Java プログラムを対象に計算した結果、すべてのプログラムを検証するためのテストデータの生成ができたことを示し、その結果に関する考察を行っている。</p> <p>第6章では、本論文で得られた研究成果をまとめ、その研究成果の応用に関する今後の展望および研究をさらに発展させるための課題を述べている。</p>			

以上の学位論文の内容から、審査委員会は次のように評価した。

1. 創造性について

ソフトウェアの信頼性を検証するためのテストデータ生成は指数時間を要する NP・完全問題である。これまでは、パスやデータフローなどの解析による解決法が提案されてきた。本論文では、プログラムネットを用いた斬新な方法を提案している。テストデータを作り出すには既存のツールを活用しているが、それまでの処理はすべて指数時間より遥かに短い多項式時間のアルゴリズムを開発して実現できている。よって、本論文は、創造性において優れている。

2. 論理性について

第1章では、先行研究を引用しながら課題を明らかにしている。第2章から第5章までは、必要な概念や定義を示したうえで、プログラムネットを用いたテストデータ生成という課題の解決に必要な処理ステップを明示し、それぞれのアルゴリズムを綿密に設計・検証している。このように、本論文は、課題の提示から解決法まで一貫性のある展開をしていることから、論理性において優れている。

3. 厳格性について

ソフトウェア・テスト問題に関連する文献を可能な限りカバーしており、関連する先行研究についても丁寧に渉猟している。また、プログラムネットの構造解析結果やアルゴリズムの正当性についても厳密に証明している。こうしたことから、本論文は、厳格性において達成できている。

4. 発展性について

ソフトウェアのテストデータ生成にプログラムネットを導入することによって、ソフトウェアの信頼性検証の研究に新しい研究方向性をもたらすことが期待できる。さらに、テストデータを作り出すためのソフトウェアの構造解析が多項式時間で実現できたことによって、効率よいソフトウェアの開発にもつながる。よって、本論文は、発展性において達成できている。

以上より、全体的に達成できていることから、論文審査を「合」と判定した。

論文審査結果

合・否

審査委員

(氏名)

葛崎 洋

(氏名)

中田 亮

(氏名)

鷹岡 亮

(氏名)

北沢 千里

(氏名)