

ソフトウェア検証のための可変性マイニング手法

李 健翔 岸 知二[‡]

早稲田大学大学院 創造理工学研究科 経営システム工学専攻^{†‡}

1. はじめに

ソフトウェア開発において、ソフトウェアの正確性や機能性を検証することは重要であり、それらを確認する手法は多数存在している。しかしながら、システムが持つ可変性が増えてきたことで、システムは多くのバリエーションを持ち、従来の検証が困難になっている。そのため多くのバリエーションの中で実際によく使われるものを識別することが有用だと考えられ、その研究として可変性マイニングという手法が存在する。本研究では、検証を指向した可変性マイニング手法を提案し、評価として2つの検証手法への適用実験を行う。検証手法の1つであるテストに対し、テストケースの優先度付けを行うこと、そしてもう1つが可変性システムへの検証手法であるファミリーベースのモデル検査を行うことである。この2つの検証へ適用し評価をする。

2. 従来研究

2.1. 可変性マイニング

可変性マイニングとは何らかの成果物やデータから可変性に関わる情報を抽出する技術のことである。

Kishi ら[1]の研究では、Bluetooth(BT)スピーカーの利用状況や可変性情報を抽出し、FTSやFDを作成、また得られたマイニング情報を活用した効果的なテストケースの優先度付けを提案している。[1]ではテストに特化したマイニングを行っていたが、本研究ではPFM(2.3で説明)という汎用的なモデルを一旦構築することで、様々な活用をできるようにすることを目指す。

2.2. 可変性マイニング

内藤[2]の研究では、ファミリーベースのモデル検査を、単一システムを検証する一般的なモデル検査ツールを用いて行う手法を提案している。この研究では、フィーチャモデルの特徴を捉える指標の1つであるFIP(Featured Inclusion Probability)を求め、FTSに付与した拡張FTSを用いることで確率的モデル検査を可能にしている。本研究ではマイニングで得られた情報を基にモデルを作成し、利用確率やFTSの情報を付与する。作成したモデルを使ってファミリーベースのモデル検査を行うことで、実現性の高い検査ができることを目指す。

A Variability Mining Method for Software Verification

[†]Kensho RI

[‡]Tomoji KISHI

^{†‡}Department of Industrial and Management Systems Engineering CSE Graduate School, Waseda University

2.3. Probabilistic Feature Model [3]

フィーチャモデル(FM)とはフィーチャとその間の論理的制約が書かれた可変性モデルである。一方Probabilistic Feature Model(PFM)はフィーチャ間の制約を条件付き出現確率として表現したものである。PFMでは、親フィーチャが出現した時の子フィーチャが出現する確率や、他のフィーチャ間の条件付確率であるソフト制約も付け加えられている。本研究では、これを用いたテストおよびモデル検査での活用について言及する。

3. 提案手法

3.1. 提案手法の概要

提案手法の概要図を図1に示す。

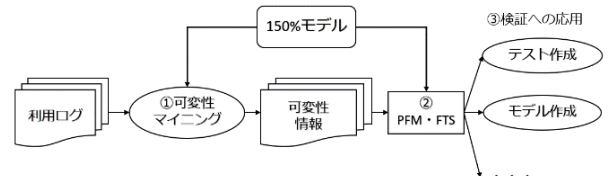


図1. 提案手法の概要図

提案手法は大まかに①可変性マイニング②PFMおよびFTSの作成③検証への応用(テスト作成、モデル作成など)の3つの手順から成り立つ。

3.2. ①可変性マイニング

[1]を基本にマイニングを行う。対象としたシステムを利用した際に得られるイベントログと150%モデル(既知の状態遷移モデル)を組み合わせることで[状態, イベント]という1つのイベント遷移列を作る。イベント遷移列はN-gramという方法で、長さ1~4で区切り抽出する。次に可変性情報の抽出を行う。得られたイベント遷移列より、それぞれのシステムがどのフィーチャを持っているかという情報が分かる。それを表したのが表1である。表1ではA~Eの5製品を対象に、それぞれが持つフィーチャには1, 持たない場合は0と表されている。

表1. 製品群のフィーチャ表

	A	B	C	D	E
['0x01', '0x4c'] playing backward	1	1	1	1	1
['0x01', '0x4b'] playing forward	1	1	1	1	1
['0x01', '0x46'] playing pause	1	0	1	1	1
['0x00', '0x4b'] stop forward	1	0	1	1	1
['0x00', '0x4c'] stop backward	1	0	1	1	1
['0x00', '0x44'] stop play	1	0	1	1	1
['0x00', '0x0d'] stop volume	1	0	1	1	0
['0x01', '0x0d'] playing volume	1	1	1	1	0
['0x01', '0x44'] playing play	0	0	0	1	0
['0x00', '0x46'] stop pause	0	0	0	0	1

次に製品群の変異性と共通性を特定する。例えば表1の場合オレンジ色の箇所は全製品から出現しているため共通性として捉えることができる。

3.3. ②PFM・FTSの作成

①可変性マイニングから得た情報より、PFMとFTSを作成する。①で得た可変性と共通性の情報を基に、それぞれのイベント遷移列をいくつかのグループとしてラベル付けを行う。その情報を基にPFMを作成する。表1の色付けはグループ分けに対応している。図2に表1から作成したPFMを示す。図の下にある数値はフィーチャ間の条件付き確率を表し、その値が0%のところはハード制約、つまりは排他的な関係が成り立つことになる。

[1]と同様にFTSを作成していく。既知の情報として150%モデルが与えられており、図2にあるグループ分けされたフィーチャが必ず対応することになるので、ブール式としてFTSのフィーチャに付与することでラベル付けされたFTSが作成できる。

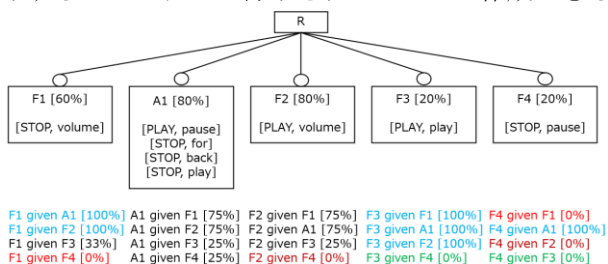


図2. PFM

3.4. ③検証への応用

テスト作成では、マイニングされた可変性情報と確率情報に基づき優先度付けを行う(以下hardと呼ぶ)。比較としてhard以外の2種類、ランダムに並べたもの(random)と確率情報のみを使用したもの(usage)の3手法で優先度付けを行う。

モデル作成では、ファミリーベースのモデル検査に使用するモデルをマルコフ決定過程(MDP)で作成する。そこに3つの情報を付与する。1つ目はラベル付けFTSの製品群の遷移とそれぞれのラベルである。2つ目は対象とした各製品の利用確率である。最後に3つ目が、図2のフィーチャの生起確率を基に製品選択やその選択率を変えることである。

4. 評価実験

提案手法の有効性を確認するため2つの評価実験を行う。評価実験の目的は得られた可変性情報で、1.テストケースの優先度付けができること、2.実現性の高いモデル検査ができることの2つである。

4.1. 実験結果

目的1を確認するためN-switchテストにn-gramで重み付けした累積グラフを作成し、その減少グラフの回帰分析を行う。その結果を表2に示す。

表2. λ 値と R^2 値

2switch	λ	R^2	3switch	λ	R^2
random-3gram	0.015	0.7985	random-4gram	0.002	0.849
usage-3gram	0.028	0.8192	usage-4gram	0.005	0.8749
hard-3gram	0.031	0.784	hard-4gram	0.007	0.8128
random-2gram	0.062	0.9393	random-3gram	0.015	0.8628
usage-2gram	0.03	0.9795	usage-3gram	0.006	0.9761
hard-2gram	0.03	0.9783	hard-3gram	0.007	0.9597
			random-2gram	0.065	0.9305
			usage-2gram	0.006	0.8096
			hard-2gram	0.006	0.7531

目的2を確認するため3.4の方法で作成したモデルに対し、(1)デッドロック(2)到達可能性(3)遷移可能性の3つを検証する。結果としては(1)のデッドロックはどの状態においても発生しないことを確認した。また(2)(3)の到達可能性と遷移可能性は共に、製品選択の割合次第ではあるが全ての状態にいずれは到達、遷移することを確認した。

5. 考察

提案手法であるhardはrandomやusageなどの優先度付けより優れており、評価実験の目的1を達成できたと言える。3.4の方法で作成したモデルのモデル検査を行った結果、それぞれの検証項目に問題はなく、実現性の高いモデル検査を行えた。評価実験の目的2を達成できたと言える。

6. 結論と今後の課題

本研究では、ソフトウェア検証を指向した可変性マイニング手法を提案した。また評価実験として、テストとモデル検査という2つの手法へ応用し、それぞれで提案手法の有効性を確認することができた。

今後の課題としては、ソフト制約を用いたテストケース優先度付けの有効性の確認が必要である。PFMのフィーチャ生起確率だけでは確率の粒度が粗く、優先度付けをする場合はそれぞれのフィーチャの関係を表したモデル(ベイジアンネットワークなど)を利用し、より細かいフィーチャ間の関係確率を用いることが考えられる。また、ソフト制約を用いたモデル検査の有効性の確認も必要である。例えばフィーチャ生起確率が5%以下のフィーチャは無視し、90%以上のフィーチャを重点的に検証するといった方法が考えられる。この場合、今回のような5製品だけでなく、より多くの製品群で行うことが必要になると考えられる。

参考文献

[1] T. Kishi, T. Koyama, N. Noda, K. Horiuchi, K. Ri, C. Zhang: A Test Prioritization Method for Configurable Software Systems based on Variability Mining, Asia Pacific Conference on Robot IoT System Development and Platform 2021(APRIS2021), 2021.

[2] 内藤裕暉, 岸知二: 可変性を持つシステムの確率的モデル検査手法, 研究報告ソフトウェア工学, 2022-SE-210(6), 1-6(2022-03-04), 2188-8825, 2022.

[3] K. Czarnecki, S. She and A. Wasowski: Sample Spaces and Feature Models: There and Back Again, 2008 12th International Software Product Line Conference, pp. 22-31, 2008.