

Abstract.

バグの場所を見つける最近の研究は正しい実行トレースと間違っている実行トレースをクロスチェックする。そこでの技術は関係ルールを見つけることです。その関係ルールはソースの**ある特定の行**が実行を失敗させてしまうことを示すルールです。けれども、この技術は限界があります。この論文ではまず、より表現力がある関係ルールを考えます。そのルールは**複数の行**が失敗を起こしてしまうようなことを表現できます。次に、FCA を使って多数のルールを分析して、ルールに含まれる情報の可読性を向上させます。この論文の主なポイントは二つのデータマイニング技術、つまり関係ルールと FCA 適用することによって、これまでのバグ特定技術より良い結果をもたらすことです。

1. Introduction

テスト環境においてプログラムを実行すると、実行のトレースと呼ばれる実行に関するデータ集合が生成されます。トレースを使うと実行エラーがある場合にプログラムをデバッグすることができます。トレースは様々な情報を含んでいます。例えば、実行された行や実行の成功失敗という情報です。バグの位置特定はトレースの内容を調べて失敗の原因を見つけます。トレースを調べる方法にはいくつかのアプローチがあります。

(中略)

特に、Jones 達はある行の実行とテストが失敗することの間の相関を調べる方法を提案した。Denmat 達は、これはある制限された形の相関ルールを見つけることと似ている、ということを示した。

相関ルールを見つけることは良く知られたデータマイニングのタスクです。

(データマイニングの話)

(FCA の話)

この論文の主なポイントは、この二つのデータマイニング技術、つまり相関ルールと FCA を適用することによって、従来のバグ発見技術よりもより良い結果が得られることを示すことです。このことは§ 6 で詳細に述べます。

2. Running Example

図 1 に示す tritype を例に使います。3 篇の長さを与えると 4 つに分類します。不当辺三角形、二等辺三角形、正三角形、三角形じゃないの 4 つです。プログラムは一つのクラスで 130 行です。このプログラムはテスト生成法の古典的なベンチマーク問題です。テスト生成法は失敗の原因となっているエラーを検出することです。この目的のために、ベンチマーク問題の変形が作られます。その変形は Web 上にあって、我々も自分たちの提案方法を評価するために使います。この tritype に対して 400 個のテストケースが生成されました。これにより、すべての実行可能な経路が一様にカバーされました。

3. バグの位置特定のための二つの形式文脈

この節ではバグの位置特定処理が基づいている情報について述べる。まず、実行トレースから文脈が作られる。それをトレース文脈と呼ぶ。次にある特定のアソシエーションルールを使ってトレース文脈をチェックします。最後に二つ目の文脈を導入します。それは多数のルールについて推論するためです。それをルール文脈と呼びます。そのルール文脈に付随するルール束の解釈の仕方については第 4 節で述べます。

The Tracce Context.

プログラムの実行について推論するためにその実行のトレースを使います。トレース情報の方について議論するのはこの論文の範囲外です。ここでは各トレースは少なくとも実行された行と成功失敗の結果を含んでいると仮定します。これはトレース文脈を構成します。トレース文脈のオブジェクトは実行トレースです。トレース文脈の属性はプログラムの行と成功失敗の判定です。

Association Rules.

失敗した実行の原因を理解するために、アソシエーションルールを見つける。それにはデータマイニングアルゴリズム[CFRD07]を使います。アソシエーションルールの数を減らすために飽和アイテム集合に基づくアソシエーションルールに注目します。すなわち、我々はアソシエーションルールで 1) ルールの前提が概念の内包で、外延は大部分失敗した実行トレースを含んでいて、成功した実行トレースもいくらかは含んでいます。(統計的な指標に従って) そして 2) ルールの結論は属性 FAIL(Definiton1)。このことは FAIL というラベルを持つ概念に関連しているコンセプトを選ぶことに対応しています。ただし、これらの概念は PASS とラベル付けられている概念に関連している可能性もあることに注意しなさい。

アソシエーションルールで統計的な尺度の最小閾値を満たすものだけが生成されます。ここではサポートとリフトを使います。サポート尺度はルールが出現する頻度を示します。ここでの応用ではルールの前提となる行が FAIL の実行例の中でどれくらいの頻度で実行されているかを測ります。リフト尺度はルールの前提の出現により、結論が現れる確率が増加するかを示します。関連するルールは最小サポートと最小リフトの値によってフィルターされます。minsupp の閾値は大変小さくなりえます。それに対して minlift の最小閾値は常に 1 より大きい等しくなります。