

オブジェクト指向プログラムの信頼性向上のための カバレッジ基準の提案

A New Coverage Criterion for Improving Reliability of Object-Oriented Programs

花本洋平[†] 古宮誠一[†]
Youhei Hanamoto[†] Seiichi Komiya[†]

[†] 芝浦工業大学 工学部

[†] Faculty of Technology, Shibaura Institute of Technology

要旨

オブジェクト指向言語で開発されたプログラムでは、ポリモーフィズムによって分岐網羅の基準が実際の分岐を反映しなくなるため、分岐網羅のテストカバレッジが100%であっても十分に分岐をテストしているとは言えない。本研究では新しい分岐網羅基準を提案し、その基準に基づいたカバレッジを取得するためのツールを開発することでオブジェクト指向プログラムの信頼性を向上させることを目的とする。また、有用性についてはオープンソースソフトウェアに対して実験を行うことで検証する。

1. はじめに

ソフトウェア開発ではモジュールの信頼性を保証するために単体テストが実施される。開発者はテストが十分に行われているかどうかを分岐網羅のようなテストカバレッジを計測することで判断する。

オブジェクト指向言語を利用した開発では、ポリモーフィズムによって分岐網羅の基準が実際の分岐を反映しなくなる。ポリモーフィズムによるプログラムの制御構造の動的な変化は、従来の手続き型言語における条件分岐の制御構造をオブジェクトのメソッド呼び出しに変換することを意味する。このため、既存の分岐網羅のテストカバレッジが100%であっても十分に分岐をテストしているとは言えなくなる。

本研究ではオブジェクト指向言語で開発されたソフトウェアの信頼性向上のために、ポリモーフィズムによるメソッド呼出しを分岐に含めたカバレッジの基準を提案する。

以下、本論の構成を示す。2章では既存のカバレッジにおける問題点を説明する。3章でポリモーフィズムによるメソッド呼び出しを分岐に含めたカバレッジを提案し、2章で説明した問題がどのように解決されるかを説明する。4章で提案における支援ツールの説明をする。5章で本研究のまとめを述べる。

2. 問題提起

本章では既存のカバレッジの問題点を、構文解析プログラムを例に用いて説明する。

カバレッジとは「プログラムのテストを実施し、どの程度期待通りに動作することを確認できた割合」を示す評価指標である。手続き型言語では、制御構造の条件分岐の実行された割合を計る分岐網羅基準が広く用いられている。分岐網羅とはプログラム内の全分岐数に対してテストケースによってどれだけ分岐が実行されたか、という基準である。分岐網羅のカバレッジの計算式は以下ようになる。

$$\text{カバレッジ} = \frac{\text{実行した分岐数}}{\text{全分岐数}} \times 100(\%)$$

図1のフローチャートで表わされているプログラムは2つの分岐を持つプログラムである。このプログラムに対して分岐網羅のカバレッジを100%にするためには、state==Defaultstate の分岐における Yes, No, c=='/' の分岐における Yes, No をそれぞれ少なくとも1回はテストする必要がある。A, B, C の経路を通るようなテストケースを用意すればいい。

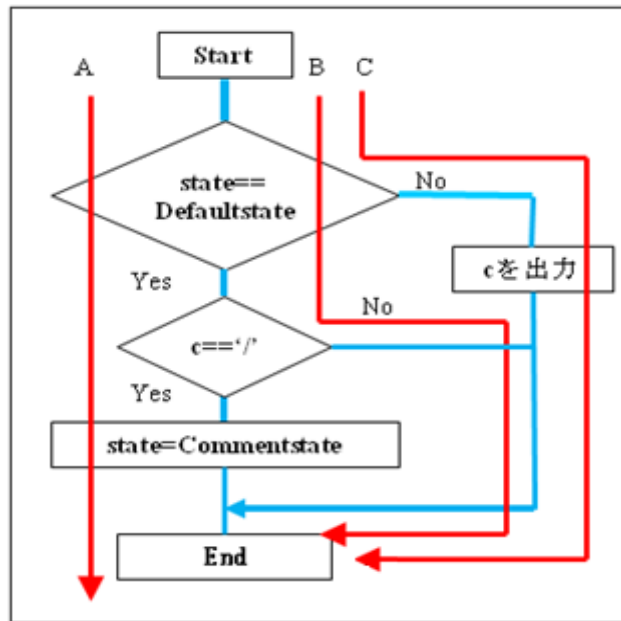


図1 分岐網羅

既存のカバレッジにおける問題点の例として、図2のような State パターン[1]を利用した構文解析プログラムを考える。Parser クラスは現在の状態を表す State 型のオブジェクトを保持し、次の文字を読み取り、State の parse メソッドを呼び出す。State の実装クラスである DefaultState クラスや CommentState クラスはそれぞれオーバーライドされた parse メソッドで読み取った文字を処理し、状態遷移として Parser クラスの保持する State を次の State オブジェクトに設定する。

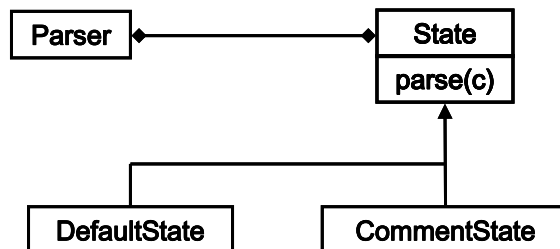


図2 構文解析プログラム

図2のプログラムを手続き型言語の条件分岐を利用して作成したものを図3、オブジェクト指向言語のメソッド呼び出しを利用して作成したものを図4に示す。

図3では2行目と4行目のif文による条件分岐、図4では4行目のポリモーフィズムによるメソッド呼び出しでstateの状態をそれぞれ判断している。図3と図4は同様の処理を行っているのでポリモーフィズムによるメソッド呼び出しは、if文と同様の機能を持っていると言える。しかし、既存の分岐網羅基準はメソッド呼び出しを分岐として判断しないため、テストしたときに図3の2行目と4行目のif文はテストされるが図4の3行目のメソッド呼び出しはテストされない。そのため、同様の処理を行っているにもかかわらずテスト結果に差が出てしまう。

図4の例ではParserとStateの組合せを網羅したparseメソッドのテストがなされるべきであるが、前述の通り既存の分岐網羅基準ではメソッド呼び出しを分岐として判断しない。したがって、既存の分岐網羅基準はオブジェクト指向ソフトウェアに対して十分ではない。

```

1: void parse(char c){
2:   if(state==DefaultState){
3:     if(c=='/') state=CommentState;
4:   }else if(state==CommentState){
5:     printf("%c¥n",c);
6:   }
7: }
    
```

図3 手続き型言語による条件分岐

```

1: class Parser{
2:   main(){...
3:     State state =new DefaultState();... }
4:   void parse(char c){ state=state.parse(c);}
5:
6: class DefalutState implements State{
7:   State parse(char c){
8:     if(c=='/'){return new CommentState();}
9:     else return this;}
10:
9: class CommentState implements State{
10:   State parse(char c){System.out.println(c);}
    
```

図4 オブジェクトのメソッド呼び出し

3. 提案手法

本研究ではカバレッジの分岐網羅の基準にポリモーフィズムによるメソッド呼び出しを追加することを提案する。ポリモーフィズムによるコード上のメソッド呼び出しを分岐として判断するためには型階層を利用する。

プログラムから既存の分岐網羅基準による分岐とポリモーフィズムによるメソッド呼び出しを合わせた総分岐数を調べ、以下の基準によってカバレッジを算出する。

$$\text{カバレッジ} = \frac{\text{実行した (分岐数+メソッド呼び出しの組み合わせの数)}}{\text{(全分岐数+全メソッド呼び出しの組み合わせの数)}} \times 100(\%)$$

ここでメソッド呼び出しの組み合わせの数とは（呼び出し側のメソッド，呼び出される側のメソッド）の組の集合の要素数である。例えば、図5のようなクラス階層の場合、Stateクラスのparseメソッドからは、DefaultState,CommentState,ComentState2,StdOutPrintStateクラスのparseメソッドが呼び出される可能性があるため、メソッド呼び出しの組み合わせの数は“4”になる。AbstractPrintStateクラスではparseメソッドがオーバーライドされていないため、組み合わせとしては計上しない

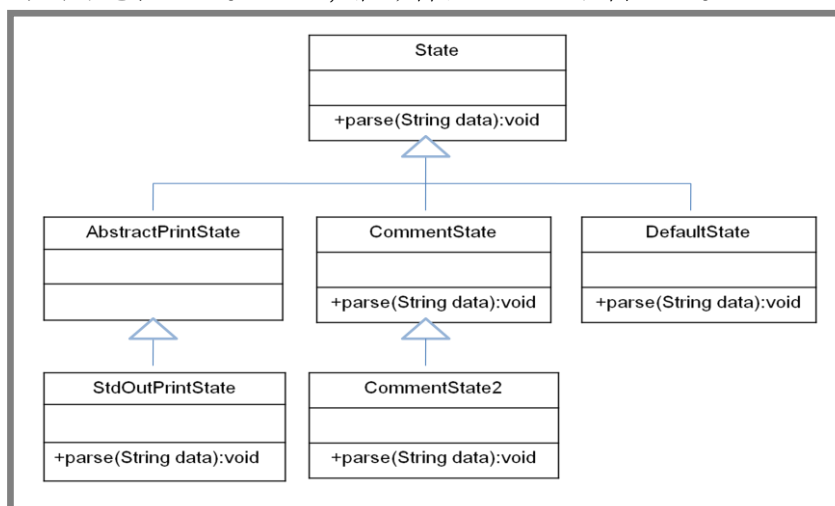


図5 メソッド呼び出しのクラス図の例

このように、ポリモーフィズムによるメソッド呼び出しを分岐として扱う新しい分岐網羅基準を利用することで、2章で述べた図4のメソッド呼び出しもテストすることが可能になる。そのため、図3と図4のテストの差はなくなるはずである。

4. 実装

本研究ではJavaプログラムに対して前章で提案したカバレッジ基準を計測するためのツールをSoot[2]を利用して実装する。ツールは対象プログラムのクラスの各メソッドに以下の手順によって、計測するためのコードを埋め込み、改変したクラスファイルを実行することで計測を行う。

1. メソッドの基本ブロック(先頭から順にプログラムを途中で分岐を持たないまとまりに分けたもの)の先頭に位置情報を記録するコードを追加する。

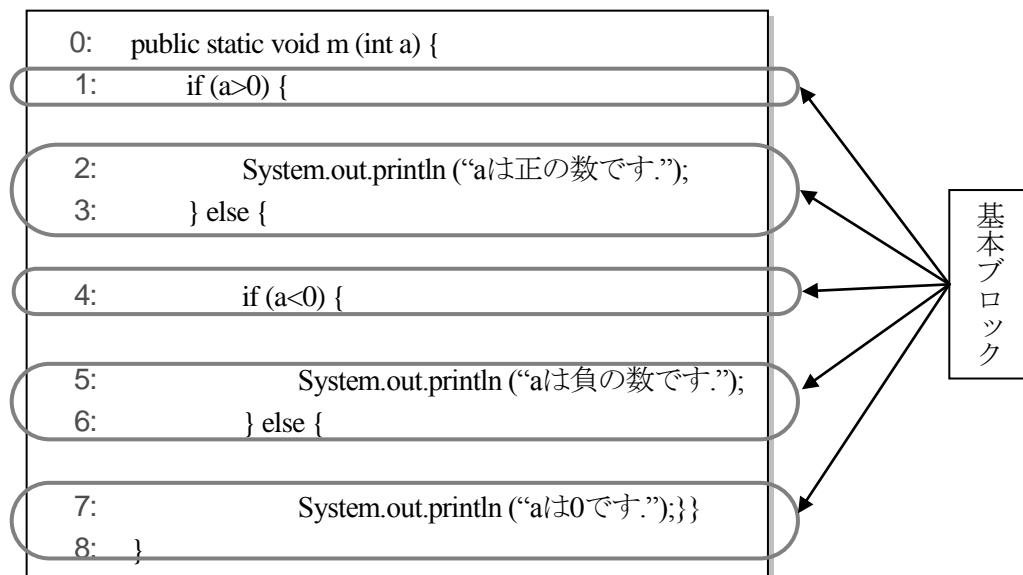


図6 基本ブロック

2. メソッド呼び出し式の前後に呼び出し側メソッドの名前をスタックのプッシュ、ポップするコードを追加する
3. 各メソッドの先頭に(呼び出し側のメソッド, 呼び出されるメソッド)の組を記録するコードを追加する。呼び出されるメソッドは2の手順でプッシュしたスタックの先頭の情報を利用する。

カバレッジの算出のためにテスト対象のプログラムから全分岐数と全メソッド呼び出しの組み合わせの数を調べる。取得した型階層を利用してテストした分岐数を数えて前述の計算式に当てはめて評価する。

5. おわりに

オブジェクト指向言語で開発されたプログラムをテストする場合、ポリモーフィズムによる制御構造の動的な変化が従来の分岐網羅基準ではテストされないという問題に対し、本研究ではポリモーフィズムによるメソッド呼び出し式を分岐に含めたカバレッジ基準を提案した。現在、ツールの実装状況はメソッド呼び出しを分岐として判別するためにメソッドの呼び出し側と呼び出される側の組を記録する機能を実装している。今後はテスト対象のプログラムから全メソッド呼び出しの組み合わせの数を調べ、テストした分岐数を数えて計算式に当てはめて評価する部分を作成する。実装終了後、オープンソースソフトウェアに対して実験を行い、有効性を検証する。

参考文献

- [1] Erich Gamma, Richad Helm, Ralph Jonson, John Vissides “Design Patterns Elements of Reusable Object-Oriented Software”.
- [2] Soot: <http://www.sable.mcgill.ca/soot/>