

レビュー支援ツールを利用した テストケース作成支援

4K-2

藤本 卓也、堀川 博史、高野 彰
三菱電機㈱ 情報電子研究所

1. はじめに

プログラムのテストを行う場合、最も難しいのはテストケースの設定である。テストケースの自動生成については、プログラムをブラックボックスとみなし、その機能等を形式的に記述したものの(論理式、BNF等)から生成するツールが発表されている。一方、プログラムの内部構造に着目したテスト支援ツールとしては、テストパス自動生成、網羅度測定を行うものが発表されている。ここでは、プログラムの内部構造に基づき、生成したテストパスをユーザに提示することで、テストケースをユーザに導き出させ、整理させる会話的ツールについて述べる。

2. レビュー支援ツールの概要

レビュー支援ツールはC言語で書かれたプログラムのコードレビュー、及びテストを支援するために考えたツールである。特定の値を設定し、これに対して机上トレースを行うタイプのコードレビュー作業を対象としている。現在、このツールはワークステーションMELCOM ME-1000シリーズ上で動作している。

このレビュー支援ツールはまず、ウィンドウ上にソースコードを表示する。ユーザはそのコードをマウスでクリックすることによってレビューの軌跡(レビューパス)を指定したり、変数に値を設定したり、変数値を確認したり、またコメントを記述したりしながらコードレビューを行う。レビュー支援ツールはこのとき得られたデータ(レビューの軌跡、コメント等)を収集する。レビューが終了したら、収集したデータが正しいかどうかを調べるため、プログラムを実際に行ってレビュー検証を行う。

このレビュー支援ツールを用いることにより、

- ①レビューデータが保存されるので、これをテスト時のデータとして利用することができる。
- ②レビューの確認を行うことができるので、誤りを早期に発見することができる。
- ③網羅度の測定を行うので、レビュー、テストの

ぬけをチェックできる。

SOURCE	REVIEW DATA
51:	ID normal_2
52: for (i = 0; i <	[cal.c:main]
53: pstr(str	RUN cal 1988
54:)	SET argc!2
55: exit(0);	24-24 If
56:	COM argc==2
57: /* print out com	28-28 If
58: xlong:	CHK argc==2
59: y = number(argv[29-29 Goto
60: if (y < 1 y >	59-59 SIMP
61: fail("Ba	CHK y==1988
62: printf("%n%n%n")	60-60 If
63: printf("	
64: printf("%n");	
65:	
66: for (i = 0; i <	
67:	
68: memset(s	
69: printf("	
70: printf("	

図1 レビュー支援ツールの表示例

3. プログラム構造に基づくテストケース出力

3.1 モジュールテスト

人間がモジュールテストを行う場合、次のような点に注目してテスト設計を行っている。

- ①今、考えている状態はプログラムのどのような状態なのか
- ②どのようなパスを通るか
- ③あるパスを通るための変数値(引数を含む)の設定

これらの項目はそれぞれ、テストケース、テストパス、及びテストデータの設計に対応している。

3.2 レビュー支援ツールの利用

レビュー支援ツールではレビューパスをユーザが決定していた。しかし、このようなパスはプログラムの内部構造から自動的に求めることができる。そこで、レビューパスをテストパスの自動生成を行う機構を用いてあらかじめ与えてしまう。このようなパスに対してレビュー支援ツールを起

動し、変数値の設定、及びコメントを収集する。このようにすれば、1回のレビューにより得られた変数値の設定を集めたものがテストデータとなり、コメントを集めて整理したものがテストケースとなる。図2に例を示す。被テストプログラムより、五つのテストパスが生成される（ループは0、1、及び2回の場合を考えるものとする）。このパスに対してレビューを行う。例にパス②でレビューを行った場合のコメントを示す。すべてのパスに対してレビューを行い、得られたデータを整理すると、例えばパス①はテーブルがない場合というテストケースであり、テストデータとしては変数table_sizeの値を0にすれば良いということがわかる。このツールを使うことにより、テストパスの発見、テストケース、及びテストデータの作成をまとめて扱うことができる。また、レビュー検証部を利用することにより、テストの実行も支援できる。

被テストプログラム

```

1 list_reserve()
2 {
3     for (i = 0; i < table_size; ++i)
4         if (l_table[i].reserve == 0) {
5             l_table[i].top = l_table[i].bot = NULL;
6             l_table[i].l_no = i;
7             l_table[i].reserve = 1;
8             return (i);
9         }
10    return (-1);
11 }

```

テストパス

① 3-10
 ② 3-4-3-10
 ③ 3-4-3-4-3-10
 ④ 3-4-(5~7)-8
 ⑤ 3-4-3-4-3-(5~7)-8

レビュー時のコメント (テストパス②の場合)

① 3: i==0 and table_size==1の場合
 ② 4: l_table[0].reserve!=0の場合
 ③ 3: i==1 and table_size==1の場合
 ④ 10: 異常終了の場合

データ整理後

・テストケース

- ① テーブルがない場合 (異常終了)
- ② テーブル数が1で、そのテーブルが使用中である場合 (異常終了)
- ③ テーブル数が2以上で、そのすべてのテーブルが使用中の場合 (異常終了)
- ④ テーブル数が1以上で、1番目のテーブルが未使用の場合 (正常終了)
- ⑤ テーブル数が2以上で、2番目以降のテーブルのどれかが未使用の場合 (正常終了)

・テストデータ

- ① table_size<-0
- ② table_size<-1, l_table[0].reserve<-1

~

4. おわりに

モジュールテストに対して、テストパスの発見とテストケース、及びテストデータの作成を支援するツールの概要について述べた。今後は本ツールの試作を行い、その有効性を調査していくつもりである。

・参考文献

- [1] Duncan, A.G. and Hutchison, J.S., Using Attributed Grammars to Test Designs and Implementations, Proc. of 5th ICSE, pp.170~178, Sep. 1984.
- [2] 藤本他, プログラム図を用いたテスト支援ツール, 情処学会第32回全国大会5Y-5, pp.665~666, 1986.
- [3] 藤本他, レビュー支援ツール(1), 情処学会第36回全国大会4M-5, pp.1031~1032, 1988.
- [4] 山本他, レビュー支援ツール(2), 情処学会第36回全国大会4M-6, pp.1033~1034, 1988.