

第11章

異常の検出

デバッグの理論と実践 (P.283~294)

Andreas Zeller 著

発表者 吉富 楓雅

目次

- 正常な動作の取得
- カバレッジの比較

正常な動作の比較

アサーションの制約

- 仕様にプログラムの動作が漏れ無く記述されていれば感染の絞り込みは容易
 - * しかし実際はアサーションだけでは対応できない
 - ▶ コードの仕様が不明な時, アサーションの記述には時間がかかる
 - ▶ アサーションの記述が困難な場合がある
 - ▶ 全てを網羅する仕様を作成した場合, 元のプログラムと同じくらい複雑になる. したがって同じくらい欠陥を含む場合がある

別の使用方法

- アサーションは正常な動作を検証するものである
 - * その正常な動作を人間が記述しなければならない
- 一方でアサーションは異常な動作を検出する場合にも用いることができる
 - * 異常な動作：何らかの意味で平均から外れた動作など

異常な動作の識別

- コードカバレッジ
 - * プログラムが失敗する場合にのみ実行され成功する場合には実行されないコード
- 呼び出しシーケンス
 - * プログラムが失敗する場合にのみ発生する関数呼び出しのシーケンス
- 変数値
 - * プログラムが失敗する場合にのみ特定の値をとる変数

正常な動作の取得

- 複数の具体的事象から抽象モデルを推論する技術を用いる
 - * 具体的事象：プログラムの実行のこと
 - * 抽象モデル：そのような実行に対して当てはまる一般的なルール
- 所定のテストに成功する実行に対してこの手法を適用
 - * 通常の実行に関する抽出モデルを生成
- テストに失敗する実行の特徴の中でこの抽出モデルを満たさないものを特定する

カバレッジの比較

異常を検出するための仮説

- 異常を検出する方法は次の理論に基づく
 1. あらゆる障害には原因となる感染があり, 感染は欠陥によって引き起こされる
 2. 欠陥があるコード部分が実行されなければ感染は生じない
 3. したがって, プログラムが失敗する場合にのみ実行されるコードは常に実行されるコードよりも欠陥が含まれている可能性が高い

カバレッジツール

- 先の仮説を検証するにはコードが実行されたかどうかをチェックする手段が必要
 - * カバレッジツールを用いると簡単に実現できる
- カバレッジツール
 - * プログラムに記録コードを挿入して, 実行時に実行された全ての行を記録するように改変するツール
- 一般的にカバレッジツールはテストスイートの品質評価に用いられる

3つの数字の中間の値を返す

プログラム

```
// x, y, zの中間の値を返す
int middle(int x, int y, int z) {
    int m = z;
    if (y < z) {
        if (x < y)
            m = y;
        else if (x < z)
            m = y;
    } else {
        if (x > y)
            m = y;
        else if (x > y)
            m = x;
    }
    return m;
}
```

```
// テストドライバ
int main(int arc, char *argv[])
{
    int x = atoi(argv[1]);
    int y = atoi(argv[2]);
    int z = atoi(argv[3]);
    int m = middle(x, y, z);

    printf("middle: %d¥n", m);

    return 0;
}
```

異常の発見

- 大抵の場合うまくいくが、特定の値を入力した時処理が失敗する
 - * 例：「3,3,5」は成功するが、「2,1,3」は失敗
 - ▶ 「2,1,3」を入力すると1が返ってくる
- 様々な値を入れて実行し、コードカバレッジを見る

複数回のテストのカバレッジ比較

例 11-2 複数回のテストのカバレッジ比較

```
2 int middle(int x, int y, int z) {
3     int m = z;
4     if (y < z) {
5         if (x < y)
6             m = y;
7         else if (x < z)
8             m = y;
9     } else {
10        if (x > y)
11            m = y;
12        else if (x > z)
13            m = x;
14    }
15    return m;
16 }
```

• : 実行されたステートメント

出典 : Jones et al. (2002)

3	1	3	5	5	2
3	2	2	5	3	1
5	3	1	5	4	3
•	•	•	•	•	•
•	•	•	•	•	•
	•				
	•				
•				•	•
•					•
•		•	•		
		•			
		•			
•	•	•	•	•	•
✓	✓	✓	✓	✓	x

成功する
テストケース
の中で, 8行
目を実行す
るケースは1
つだけ

正常なステートメント

- 8行目以外の全ての行は最低2つの成功するテストケースの中でも実行される
 - * 8行目を実行するケースは1つだけ
- より多くの成功するケースで実行されているほど正常なステートメントである
 - * 12行目と13行目は全く実行されていないが、成功するケースと失敗するケースで共通するものなので異常ではない
- よって8行目が最も正常ではない

カバレッジの異常の視覚化

