3L-7

影響ネットワークを用いたソフトウェアテストデータの自動生成

三本貴裕† 浦田祐毅‡ 芳賀博英†

同志社大学大学院工学研究科† 同志社大学理工学部‡

1 はじめに

ソフトウェア開発工程において，ソフトウェアテストは必要不可欠である．しかし，ソフトウェア開発は時間が限られており，テストは全工程の終盤に実施するため，時間が圧迫されがちである．そのため，ソフトウェアテストの効率化が求められている．

ソフトウェアテストの一手法としてホワイトボックステストがある．これはプログラムの論理構造を基に，テスト用の入力を作成する手法である．具体的には，コードの実行順を定め，その順路を通る入力値を求める．しかしこの工程は必ずしも容易ではなく，これまで様々な手法が提案されてきた[1]．本報告では，従来手法で述べられている，順路違反と呼ばれる問題に対処し，また効率的な入力値探索を実現するための手法について提案する．

2 入力値の探索

既存手法の中に，ソースコードの分岐式に着目した入力値探索手法がある[1]．テストにおいて，ソースコードの中の定めた順路を通るためには，入力値を変え，条件分岐の方向を変化させる必要がある．そのために，分岐方向を変化させたい分岐の条件式を，表 1 に従い関数の形に変換する．分岐方向を真にしたければ関数の値を負に，偽ならば正になるよう探索を行う．これを繰り返し，徐々に目的の順路に近づける．

表 1: 分岐関数への変換

分岐式	分岐関数F
$a > b$	$b - a$
$a \geq b$	$b - a$
$a < b$	$a - b$
$a \leq b$	$a - b$
$a = b$	$abs(a - b)$
$a \neq b$	$-abs(a - b)$

2.1. 影響ネットワーク

入力値探索の効率化のために，影響ネットワークという概念を用いる．これは，変数の定義と参照を解析し，分岐に影響する変数を特定する手法である[1]．図 1 の 1 番の分岐では，変数c

```
    void foo(a, b){
0   {   c := a * 5
    {   d := b / 2
1       if(c > 30){
2           if(c == d){
3               print "true true"
            } else {
4               print "true false"
            }
        } else {
5           print "false"
        }
6       return
    }
```

図 1: テスト対象プログラム

が用いられている．その前に c には，a*5 が代入されているため，分岐 1 の方向を変化させるためには入力 a を変えれば良いと解る．

2.2. 順路違反

分岐 2 では，d は b によって決まるので，入力 a, b を変化させれば分岐方向が変化するが，a を変化させると，その手前の分岐 1 の方向が変わる可能性がある．着目した分岐より手前の分岐方向が変わってしまう現象を順路違反という．従来手法では順路違反が発生すると，前の状態に戻し，他の変数を変化させるなどして対応する．しかし，それでは近傍探索だけ行うことになり，広い入力空間を探索できず，場合によっては目的とする順路にたどり着けない可能性もある．

3 提案手法

順路違反に対処するためには，広い空間を探索する必要がある．そのため，基となる入力から入力値を変化させてゆくという手法を用いた[2]．この手法においては，基の入力値を変化させる規則が重要となる．規則の決定するにあたり，先述の影響ネットワークを用いた．

3.1. 入力値変換規則

本報告では，3 つの入力値変換規則を提案する．1 つ目は値を増減させる規則である．ここで，増減させる入力変数は，先述の影響ネットワークを用い，方向を変化させたい条件式に影響している変数に限定する．この規則により順路違反が発生した場合は，増減させた値が大きいほど，入力空間を広くしていると捉え，探索を続行させる確率を増加させる．2 つ目は，値の入れ替えである．図 2（左）のように，分岐式の一方の変数に入力値 a が，もう一方に入力値 b が影響している場合，a と b の値を入れ替え，大小関係

Automated Software Test Data Generation based on Influenced Network
†Takahiro Mitsumoto, Hirohide Haga, Graduate School of Engineering, Doshisha University
‡Yuki Urata, Faculty of Engineering, Doshisha University

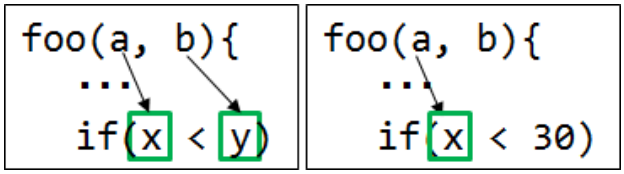

図 2: 分岐に影響している入力値

の逆転，分岐方向の変化を図る．この目的から，値の入れ替えにより順路違反が発生すると，その先には探索を行わない．最後が，値の参照である．図 2（右）の分岐を真としたい場合，その先はさておき，少なくともこの分岐が真となっている別の入力値が存在している可能性がある．そういった場合に，この分岐に影響している入力値を他から参照し，分岐方向の変化を図る．この規則により順路違反が発生した場合も値の入れ替えと同様に，その先の探索は行わない．

4 評価実験

入力値探索の速度を比較するために評価実験を 2 つのテストプログラムを用いて実施した．テストプログラムの概要を以下に示す．

【三角形の分類】

入力値は 3 つの整数(1～300)で，それぞれを三角形の一辺とした時の三角形を分類する．三角形の種類は，正三角形，二等辺三角形，不等辺三角形，更に三角形とならない場合がある．

【ポーカーの役判定】

入力値は 10 個で，トランプのカード 5 枚を表現する．その 5 枚のカードがポーカーというゲームにおいてどの役にあたるのかを判定する．

それぞれのプログラムで全順路を通る入力値を見つけるまで，ランダム生成と提案手法の生成データ数と速度を比較する．また，三角形の分類については人によるテストも実施した．

4.1. 実験結果

実験を実施した結果を表 2～4 に示す．人手による実験の被験者は皆情報工学分野の学生で，B～D はソフトウェアテストの心得もある．また，制限時間は 20 分とした．

表 2: 人手による三角形分類
テストデータ生成時間と正答数

被験者	時間	正答数(全18問)
A	20:00	2
B	20:00	11
C	18:56	17
D	20:00	14

表 3: ランダムテストと提案手法の比較

	三角形分類		ポーカー役判定	
	データ総数	生成時間(s)	データ総数	生成時間(s)
提案手法	9923	54.2	921967	3365
ランダム生成	93531	231.3	725413	1826

5 考察

三角形の分類はランダム生成，人手による生成よりも効率性の面で優れていることが解る．三角形の分類のプログラムで最も達成しづらい順路は，入力値が成す三角形が正三角形である場合の順路である．提案手法においては，参照規則が効果的に作用したと考えられる．ソースコードでは，順に辺の長さが同じであるか判定しているので，二等辺三角形のデータがあれば，その入力値を参照できる．そのため，あと一つのデータが他と一致しさえすれば正三角形の順路を達成できる．

一方ポーカーの役判定は効率が悪い結果となった．原因として，提案手法において，同一の入力値の個数は全体の 85%となり，無駄なデータが多かったことが挙げられる．その理由は，分岐に影響している変数の数が多く，それぞれに対して増減を繰り返すと，数ステップの後には同じデータを多数生成してしまうことが考えられる．

```
if(a == b){
    if(b == c){
        type = 2;//正三角形
    } else {
        type = 1;//二等辺三角形
    }
    ...
```

図 3: 三角形分類プログラムの一部

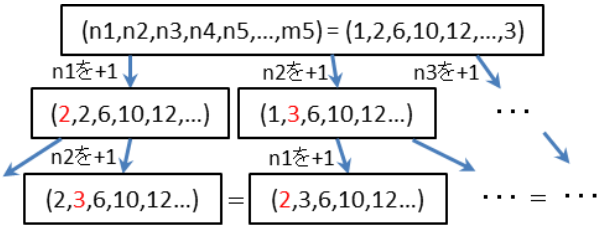

図 4: 値の増減により同一の入力を生成する過程

6 おわりに

本報告では，従来の探索における問題であった順路違反を適切に扱いながら，コード解析を用いた効率的な探索のためのデータ変換規則を提案した．その結果，分岐に関わる入力変数の数が少ない場合は有効に変換規則が作用したが，多い場合は，効率性を著しく低下させる要因となった．今後は同一の入力データ生成数を減らす必要がある．

参考文献

[1]Bogdan Korel. “Automated Software Test Data Generation” *IEEE Transactions on Software Engineering.* Vol. 16, No. 8, pp.870-879, (Aug, 1990)

[2] Lijun Shan , Hong Zhu. “ Generating Structurally Complex Test Cases By Data Mutation: A Case Study Of Testing An Automated Modelling Tool” *The Computer Journal.* Vol. 52, No. 5, pp.571-588, (2009)