制御フローグラフ上の辺のペアを考慮したコードカバレッジ基準の提案

大阪大学大学院 情報科学研究科
西本 哲，土屋 達弘，菊野 亨

目次

1. はじめに

- **背景**
  - コードカバレッジとは，ソフトウェアテストにおいて実行されたプログラムコードの範囲を表す指標である
- **目的**
  - 制御フローグラフを利用したカバレッジ基準の提案
  - 既存の代表的なコードカバレッジでは見逃す可能性のある実行をカバーすることにより，より良いソフトウェアテストの実現を目指す

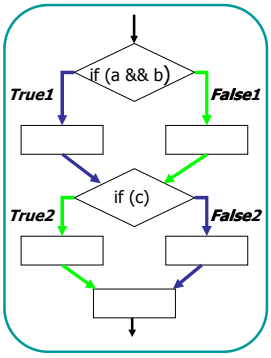
2. 代表的なコードカバレッジ

- 命令網羅率(C0)
  - 実行した命令文/全命令文
- 分岐網羅率(C1)
  - 実行した分岐/全分岐
- 条件網羅率(C2)
  - 実行した条件/全条件

より多くの指標を用いてテストを進めるほど，網羅できる経路が多くなり，より効果的なソフトウェアテストを行うことができる

3. 制御フローグラフとコードカバレッジ

- 二つの経路を通ることにより，C0,C1,C2が100%となる例あり
  - (a,b,c)=(T,T,T)の時
    True1-True2
  - (a,b,c)=(F,F,F)の時
    False1-False2
- しかし，上記二つの経路だけでは網羅できていない経路が存在
  - **True1-False2**
  - **False1-True2**



制御フローグラフの例

4. 提案カバレッジ基準について

- 対象プログラムの制御フローグラフの辺のペアを利用したカバレッジ基準C_nの提案

定義

制御フローグラフにおいて，任意のパス上に存在する辺のペアをどの程度網羅しているか

提案カバレッジ

$$C_n = \frac{A}{B}$$

A：実行された辺のペアの数
B：制御フローグラフにおける全ての辺のペアの数

5. 提案カバレッジの導出例

パス①②⑦⑧とパス③④⑤⑥について考える

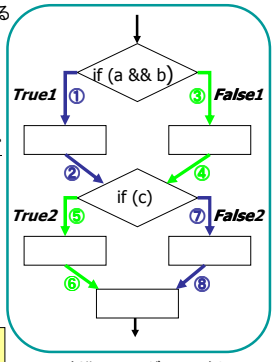
パス①②⑦⑧
(①,②), (①,⑦), (①,⑧), (②,⑦), (②,⑧), (⑦,⑧)
6ペア

パス①②⑦⑧
(③,④), (③,⑤), (③,⑥), (④,⑤), (④,⑥), (⑤,⑥)
6ペア

計 12ペア

グラフ全体のペアの総数は**20ペア**なので，
$C_n = 0.6\ (60\%)$
となる

同一パス上に存在しない辺のペアは考慮しない！
（例：(①,③), (①,④), (⑤,⑦) 等）



制御フローグラフの例

6. 提案カバレッジの特徴

- 制御フローグラフにおいて，全ての命令文・分岐を網羅
  ⇒**C0，C1を保証**
- 変数の定義が行われるブロックと使用が行われるブロックの組み合わせを網羅
  ⇒**定義-使用の道筋の網羅を保証**

制御フローグラフを用いたカバレッジでありながら，同時にデータフローテストとしても利用することが可能！

7. C言語プログラムの実例

```
int main(int argc, char *argv[]){

  int x,y,z = 0;
  int sum = 0;
  int avg = 0;
  int i = 0;

  x = atoi(argv[1]);
  y = atoi(argv[2]);
  z = atoi(argv[3]);

  if (x != 0){
    sum += x;
    i++;
  }

  if (y != 0){
    sum += y;
    i++;
  }

  if (z != 0){
    sum += z;
    i++;
  }

  avg = sum / i;  // i=0 の場合エラー発生
  printf("Average of %d integer : %d¥n",
          i, avg);
  return 0;
}
```

入力された3整数の平均値を出力するプログラム
（0が入力された場合は，他の数値のみの平均値）

プログラムに含まれているバグ

入力が (0,0,0) であった場合，
Floating point exception エラーが発生

◆例：(1,0,3)，(0,2,0) の2組を実行した場合

全ての命令文，分岐，条件を実行しているため，
C0，C1，C2 いずれも**100%**となる

⇒バグを発見できずテスト終了となる可能性有

一方，提案カバレッジC_nを求めると，**86%**という結果が出る

⇒網羅されていないペアがあると判明し，さらにソフトウェアテストを進めることによってバグ発見に繋がる