

限界値に着目したテスト網羅度の提案

1D-8

中村 寿彦、白井 和敏、中本 幸一、門田 浩
NEC マイコンソフト開発環境研究所

1 背景

テスト工程において、未テスト部分発見のための指標、進捗を表す指標などとして、テスト網羅度（以下、単に網羅度と記す）が使われている。これまで、網羅度測定のための基準として、命令網羅度、分岐網羅度 [1] などが使用されてきた。これらの測定基準では、これまで実行されていなかった部分が実行されさえすれば、網羅度は向上する。

しかし、実行したかどうかだけではなく、テストデータの内容が重要である場合がある。境界条件の誤りなどがその例である。そして、このような境界条件の誤りは、ある特定のデータで実行しないと発見が困難である。また、この境界条件の誤りに関連した障害が実際に多く存在している。そこで我々は、このような誤りを検出できるテストデータで実行したかどうかを表す指標として、網羅度を利用することを考えた。この網羅度が限界値網羅度である。

以下では、誤りの傾向を分析し、それを踏まえた限界値網羅度の概要を示す。そして、この限界値網羅度の問題点とその解決策を検討する。

2 誤りの傾向

文献 [2] によれば、ソフトウェア開発のコーディング段階において誤りの存在する文の種別は、条件文が全体の 42% を占めている。

条件文に関して、よくある誤りを特に述部に着目して分類すると、以下のようになる。

- 関係演算子の間違い
- 境界値の間違い
- 境界値の余分や抜けなど

このような条件文の誤りは、特定のデータで実行しないと判明しないことが多い。

3 限界値網羅度

2章で分類した誤り、特に関係演算子の誤り、境界値の誤りを検出するためには、境界値の付近を通過するテストデータを使用してテストをする必要がある。そこ

で、境界値付近を通過するテストデータを使用したかどうかを網羅度という形で数値化することを考えた。この網羅度を限界値網羅度と呼ぶ。

考案にあたっては以下の点を考慮した。

- 境界の通過が実行時に容易に判別可能であること
- 網羅度 100% にするためのテストデータの量が、これまでの網羅基準を 100% にするためのテストデータの量から大幅に増えないこと

3.1 定義

まず、用語を定義する。

ソースコードにおいて、実行パスが同じになる入力値の集合をドメインとよぶ。ドメインは条件式の集合で表現される。但し、ここで条件式とは、真偽値を得られる最小単位の式のこととする。

また、個々の条件式で規定される面の中で、他のドメインと接する部分をドメインの境界と呼ぶ。1つのドメインには条件式の数だけ境界を持ちうる。但し、“ \neq ”の条件式は、2つの境界があると考えられる。また、else など条件式が明示されていない場合も同様に境界を持つ。

ある境界がドメインに含まれる場合は、その境界に関してドメインが閉じていると表現し、ある境界が別のドメインに含まれる場合は、その境界に関してドメインが開いていると表現する。

これらの定義を使って、以下では、限界値、限界値網羅度を定義する。

限界値とは：ある境界に関して、ドメインが閉じている場合は、その境界上の任意の値を、開いている場合は、その境界からある微小量 ϵ の距離以内のドメイン内にある任意の値を限界値とする。

ϵ は、ドメインの境界を定めている条件式で使用されている変数の最小ステップ値であることが理想である。例えば、変数が整数型であれば、 ϵ を 1 と扱えばよい。しかし、実数型などではこの規定は現実的でない。そこで、変数の型などにより、システム、又はユーザが分岐の条件式ごとに決定する。

限界値網羅度とは：限界値網羅度は、以下の式で定義できる。

$$\text{限界値網羅度}(\%) = \frac{F}{W} \times 100$$

W はテスト対象の総数で、ドメインの1つの境界に対して、テスト対象を1つとして算出する。一方、 M はテスト対象のテスト済み総数である。テスト対象となっている境界を限界値で通過した場合、そのテスト対象をテスト済みとなる。

3.2 テスト対象の例

```
1: read(a, b);
2: x = func1(a); y = func2(b);
3: if(y > 0 and x + y < 0) ...
4: else ...
```

上記プログラムにおける3行目の条件文では、 $y > 0$ and $x + y < 0$ が述部となる。ここで、限界値は x, y という変数を用いて判定される。この if が成立するバスのドメインを表したものが、図1の塗りつぶされた部分である。条件式が2つあるので、このドメインには、テストの対象となる境界が2つ存在する。一つが、(a) の領域の任意の1点であり、もう一つが(b) の領域の任意の1点である。また、4行目の else に対しては、それぞれの条件式で、太線で示された境界上の任意の1点がテスト対象となる。

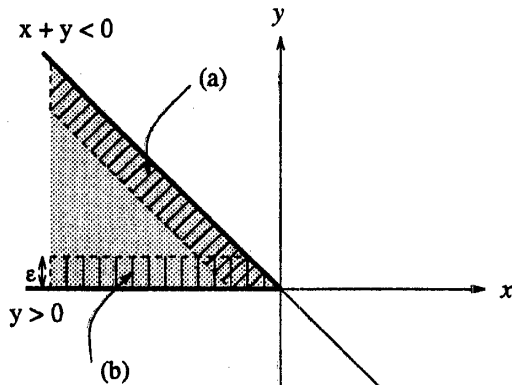


図1: 限界値網羅度のテスト対象

3.3 網羅基準の強弱比較

分岐網羅度では、同一データで複数の分岐が実行された場合、その全てがテスト済み集合にカウントされるが、限界値網羅度では、分岐を限界値で通過しなければ、既テスト集合の要素にカウントされない。一方で、全ての分岐は、限界値での通過という形でテストされ

る。従って、限界値網羅度は分岐網羅度より強い基準である。

また、一般的にバス網羅度が最も強い網羅基準とされている。テストデータの数という観点では、バス網羅度より少くなると考えられるが、バス網羅度で100%網羅するテストデータ集合が、常に限界値網羅度で100%網羅するテストデータ集合とはならない。従って、限界値網羅度は、バス網羅度に対して比較不可能である。

4 問題点とその解決策

分岐の限界値を通過させるという条件は、単に分岐を通過させるという条件よりも制約が厳しい。従って、その制約を満たすテストデータを考えることは容易ではない。そこで、何らかの支援が必要である。これに関して、現実的には次のような対応が考えられる。

1. 分岐中で使用されている変数ごとに限界値を通過するテストデータの範囲を示す。
2. 全ての分岐を通過するテストデータを与えたのち、ある分岐に着目して、その分岐に関して限界値を通過するための条件を示す。

1の方式の実現は容易であるが、ユーザが最も欲しい情報は、分岐の時点ではなく、入力の変数についての条件である。2の方式では、まず初めにテストデータがある程度用意しなければならないという欠点がある。今後まず、1、2の方式を併用して、それぞれの欠点を補うことを考え、さらにユーザにとって分かりやすい対処法を考えていく。

5 今後の課題

現在、限界値網羅度を測定するツールを作成中である。今後、このツールを用いて、実際の開発における限界値網羅度の効果などを評価していきたいと考えている。

参考文献

- [1] Beizer, B., (小野間彰, 山浦 恒央訳): ソフトウェアテスト技法, 日経 BP 出版センター (1994).
- [2] 下村隆夫: CASE ツールの開発におけるソフトウェアバグの分析, 情報処理学会論文誌, Vol. 35, No. 7, pp. 1380-1389 (1994).
- [3] White, L. J. and Cohen, E. I.: A Domain Strategy for Computer Program Testing, *IEEE Trans. Softw. Eng.*, Vol. SE-6, No. 3, pp. 247-257 (1980).