

サンプリング法を適用したテスト網羅率の 測定方法

石黒正典 (日本電信電話公社 横須賀電気通信研究所)

1. はじめに

ソフトウェアの品質確認の最終段階であるテスト工程においては、テスト網羅率についての客観的な評価技術の確立が重要な課題の1つとなっている。比較的小規模なソフトウェアを対象としたテスト網羅率の測定方法は幾つか提案されているが、大規模なソフトウェアに関しては、そのニーズが高いにも拘わらず実用的な方法が提案されていない。

本稿は、大規模なソフトウェアの統合テスト工程に適用できる実用的なテスト網羅率の測定方法の確立を目的として、

- ① サンプリング法を導入した測定対象の削減方法
- ② 測定準備・測定工数を従来方式より大幅に削減するテスト網羅率の測定方法

を提案し、その有効性、適用領域についての評価を行った結果、大規模なソフトウェアに対して適用可能である見通しを得たので報告する。

2. ねらい

現在、テスト方法として、仕様書からテスト・データを作成し、出力結果を確認する機能テストが広く用いられている。しかし、機能テストにおいては、テスト・ケースの抽出方法が確立されておらず担当者の経験や勘に依存する部分が多く、テスト・ケースに漏れが生じ易いという問題があり、テストの網羅性を客観的に評価するのは難しい。ソフトウェアの大規模化、処理の複雑化に伴い本問題は顕著となる傾向があり、機能テストのみで解決するのは困難である。

機能テストの上記問題を補う方法として、条件分岐等のプログラムの内部構造に着目しそれらをできるだけ網羅するようなテスト・データを作成し実行する構造テストが最近注目されている。構造テストを用いればプログラムの内部構造の網羅率(全てのパス、エッジ、ステートメント等の内部構造の内、実行された部分が全体に占める割合)により、テスト網羅率を客観的に評価することができる。しかし、例えばエッジに着目すると、表 2-1 に示すように大規模プログラムの場合、4万~20万ものエッジが存在し、実際のテスト工程においてこれら全てを対象に網羅率を測定することは非現実的である。また、パスに着目した場合は天文学的な数にのぼり、網羅率の測定は不可能である。これらの理由から比較的小規模なプログラムレベル

に対して構造テストを適用した例はみられるが⁽¹⁾⁽²⁾、大規模プログラムへの適用例は見当たらない。このため大規模プログラムのエッジの網羅率を現実的な工数で測定可能とすることをねらいとして、測定対象を大幅に削減しても網羅率を正確に得られるような、サンプリング法の網羅率測定への適用を可能とする。

表 2-1 プログラムのエッジ数の比較

	大規模プログラム	小規模プログラム
規模 (ks)	100 ~ 500	1 ~ 5
エッジ数	4万 ~ 20万	400 ~ 2000

また、現在までに開発されているエッジの網羅率測定ツールの測定方式の大部分は、ソース・プログラムに測定機構を挿入する方式であり、大規模プログラムに適用する場合以下の問題点が挙げられる。

- ① ソース・プログラムに測定機構を挿入するため、ロードモジュールを再作成するリコンパイル/リンク等の測定準備のための工数が必要となる。
- ② リコンパイルによりプログラムのアドレスが変わるため、既存バッチ・カードが無効になり測定対象ファイル専用のバッチが必要となる。

一方、測定方式に拘わらず、現在のツールにおいては以下の共通的な問題がある。

- ③ 測定機構の挿入によりテスト実行時間が数倍~数百倍になる⁽³⁾。

④ プログラムに変更があるとプログラムの内部構造が変化するため、その時点までの測定結果が無効となる。
 このため、測定準備作業の削減、測定時間の短縮を可能とするエッジの網羅率測定法を実現する。

3. 測定対象エッジの削減方法（サンプリング法の導入）

2章で述べたように大規模プログラムに対して全てのエッジを対象に網羅率の測定を行うのは非現実的である。測定対象を大幅に削減し、かつ正確な網羅率の測定を可能とする方法として、統計処理において広く用いられているサンプリング法を利用することとした。このサンプリング法をプログラムのエッジに適用し、測定対象エッジ数を大幅に削減し、現実的な工数での網羅率測定を実現可能とする。

アルゴリズムの簡易性、サンプル数の削減効果、プログラムの特徴にあったサンプリング法、といった観点から、以下のサンプリング法について実際にプログラムに適用し、その有効性を検証した（表 3.3-1）。

表 3.3-1 適用するサンプリング法

No	項目	ランダム・サンプリング	モードによるサンプリング	層別サンプリング
1	概要	<p>(式1) によりサンプル数を求める。</p> $n = \frac{N}{(N-1)\left(\frac{1}{k_a}\right)^2 + 1} \quad (式1)$ <p>N: 母集団の数 L: 信頼幅 k_a: 信頼度 サンプル数 n はランダムに抽出する。</p>	<p>モードを用いて母集団から一定間隔にサンプリングする。 n個おきにサンプルを抽出することをモード n でサンプリングすると呼ぶ。</p>	<p>母集団をある分割基準で層に分割し、各層からサンプリングを行う。各層のサンプル数の決めかたには以下の2つがある。</p> <p>① 比例配分 ② ネイマンの最適配分</p>
2	選定理由	<p>サンプリングという統計的手法のプログラムへの適用性を検証するため、最も一般的な本手法を適用し、その有効性を確認する。</p>	<p>・アルゴリズムの実現が容易 ・一定間隔にエッジを抽出するため、独立性の高いエッジを多くサンプリングできる可能性が高く、精度の向上が期待できる。</p>	<p>プログラムの様々な属性（エッジ数、CALL文の数、チェック項目数等）をもとに層に分割できる可能性がある。最適な層が見つかれば、少ないサンプル数で高い精度が期待できる。</p>
3	特徴	<p>母集団の数Nがどれだけ大きくなってもサンプル数は高々1500程度で、十分な推定精度が保証される。</p>	<p>モードを用いることにより、母集団から一様な偏りのないサンプリングが可能。</p>	<p>母集団の性質を生かしたサンプリングが可能。</p>

① ランダム・サンプリング

母集団からランダムにサンプルを抽出する最も一般的なサンプリング法。

② モードによるサンプリング

ランダム・サンプリングの1つの場合に相当し、プログラムの先頭のエッジから n 個おきにサンプルを抽出する（これを、モード (mod) n でサンプリングするという）方法。本方法は、サンプリング・アルゴリズムの適用が容易であるという特徴をもつ。測定精度を維持しながら、n をどこまで大きくすることができるかが本方法のサンプル数削減効果の決め手である。

③ 層別サンプリング

網羅率と相関をもつプログラム属性（例えばエッジ数、バグ数、Cyclomatic Number）を抽出し、プログラムを幾つかの層に分けてサンプリングを行う。各層の網羅率がほぼ等しいように層に分割されていれば、層内から数個のルーチンを選んでそこからサンプリングしても高い精度で網羅率を推定することが可能となる。有望な層をいかにして見いだすかがサンプル数削減効果の決め手である。

4. 測定準備削減、測定時間短縮のための測定方法

大規模プログラムに適用可能なエッジ網羅率測定方法の実現を目的に、①測定機構の挿入に伴うリコンパイル等の準備のための工数を削減し、パッチによる修正を可能とすることを狙いとして、オブジェクト・プログラムに測定機構を挿入し測定を行う方法、②測定時の実行時間を短縮するための測定方法、をそれぞれ実現した。

4.1 オブジェクト・プログラムからのエッジ抽出法

当社で開発したシステムプログラム用記述言語であるSYSL⁽⁴⁾を対象言語とした。SYSLソースコードにおいては、①IF文 ②DO WHILE文 ③CASE文、の3つの構文がエッジに相当する。SYSLコンパイラは、エッジに相当する構文はオブジェクト・コードとしてBC命令 (Branch on Condition) を生成する。しかし、エッジに相当する構文以外にもBC命令を生成する場合があります、それらのケースと対策を検討し、オブジェクト・プログラムの先頭から無条件ブランチを除いてBC命令をサーチすることによりオブジェクト・プログラムからのエッジ抽出法を確立した (表 4.1-1)。

表 4.1-1 SYSLオブジェクトからエッジ抽出時の問題点と対策

No	項目	問題点	対策
1	エッジ以外のBC命令の出現	以下の構文はBC命令に展開される。 ① 特定のオプション指定時のPROC文 ② 特定のオプション指定時のENTRY文 ③ ①のPROC文に対するEND文 ④ GO TO文 ⑤ RETURN文	①～⑤は、無条件ブランチ指定のBC命令である。当該BC命令はエッジとしてカウントしないようにした。
		コレクタ*のSDP (Static Deraill Point) で指定されるSDP-IDの先頭がBC命令とおなじヘキサパターン ('47') を持つ場合がある。 *DSをトレースするツール	SDPは出現パターンが以下の様に固定である。 SVC 15,112 (0,0) DC HL4 ('SDP-ID') このため、SDPのSVC命令であれば次の4バイトを読みとばす。
2	命令部と定数部の区別	SYSLでは命令部と定数部を同一のセクション内にもつ。定数部にBC命令とおなじヘキサパターンをもつ値で始まる定数が出現すると、エッジとみなす	命令部の最後のEND文は、以下の形式に展開される。 BCR 15,BR15 本形式の命令が出現すれば、命令部の終了と判断する。

4.2 網羅率測定時間の短縮化

網羅率の測定方法として、エッジに相当する命令を不当命令に変換し、実行時に不当命令による割り込みの検出をもってエッジ通過とみなす方法を採用した。本方法を用いて、従来、測定時の実行時間増加をもたらしていた主な要因であった以下の2点を解決した。

① エッジ通過状況記録テーブルのサーチ時間

エッジ通過記録においては、通過状況記録テーブルの対応エントリのサーチ時間が問題となる。そこで、エッジに相当する命令に不当命令とエッジ番号を埋め込んでおき、エッジ番号をキーとして通過状況記録テーブルにダイレクトアクセスする方式を実現した。これによって、テーブルサーチ処理なしで対応エントリへのアクセスを可能とした (図 4.2-1)。

② ファイルへの測定記録

サンプリング法の導入により、通過状況記録テーブルを全てメモリ上に置くことが可能となった。そのため、測定途中のエッジ通過状況をまずメモリ上にスタックしておき、測定終了時にファイルに保存する方法を採用し、エッジ通過毎のディスクI/Oを削減した。

①の実現により、従来テーブルサーチにテーブル長に依存した時間 ($O(\log n) n$: エントリ数) を要していたものが、一定時間となり、②により、I/O時間を無くすことができ大幅な測定時間の短縮を実現することができた。

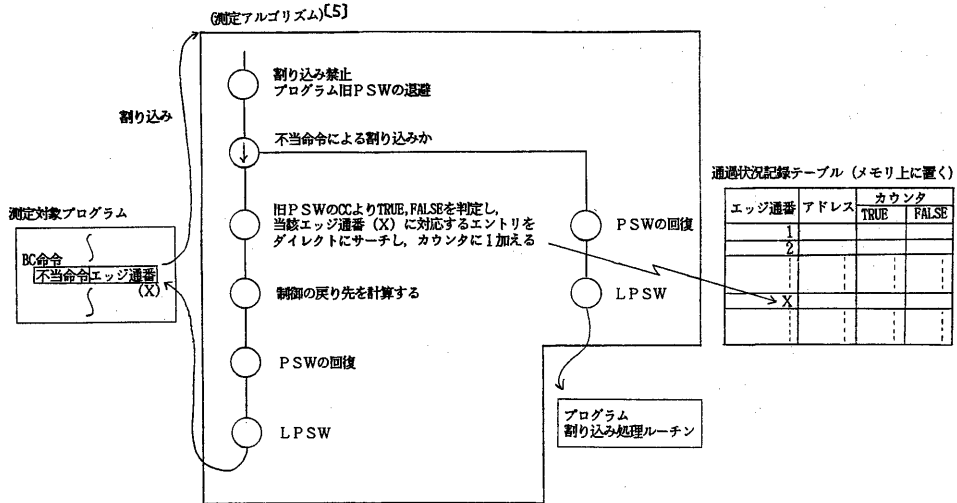


図 4.2-1 測定時間短縮のアルゴリズム

5. 実験

3, 4章で述べた測定方法を実現するエッジ網羅率測定ツール [ROUTE-C (Real-time Monitor for Evaluating Edge Coverage)] の作成を行い統合テストで用いたテスト・データを実際の試験順序に従って実行し, エッジ網羅率の測定を行い, 測定方法の有効性, 実用性を検証した。ROUTE-CはSYS Lを記述言語とし, 規模は約5.3ksである。

5.1 測定対象プログラム

測定対象プログラムは, サービス管理プログラム (以下プログラムAと略す) 及びデータベース処理用プログラム (以下プログラムBと略す) のそれぞれ一部を用いた (表 5.1-1)。

表 5.1-1 測定対象プログラムの特性

プログラム名	規模 (steps)	エッジ数	エッジの出現頻度	チェック*項目数	チェック項目の出現頻度	テストプログラム数
プログラムA	6814	2018	1/3.4steps	168	1/41steps	151
プログラムB	5873	2818	1/2.1steps	168	1/35steps	5

*機能チェック項目

5.2 測定結果

サンプリングの有効性の確認のため, 提案したサンプリング法を適用し全エッジについて測定した場合と比較することにより, 精度, サンプル数等の比較を行った。

(1) ランダム・サンプリング

表 3.3-1 (式1) に示したサンプル数の導出式から, 信頼度を95%としてサンプルエッジ数を求め (プログラムA: 873, プログラムB: 995) エッジ網羅率を測定した結果, 全エッジについて測定した場合と比較して, 0.5%未満の誤差であった (図5.2-1, 図5.2-2)。

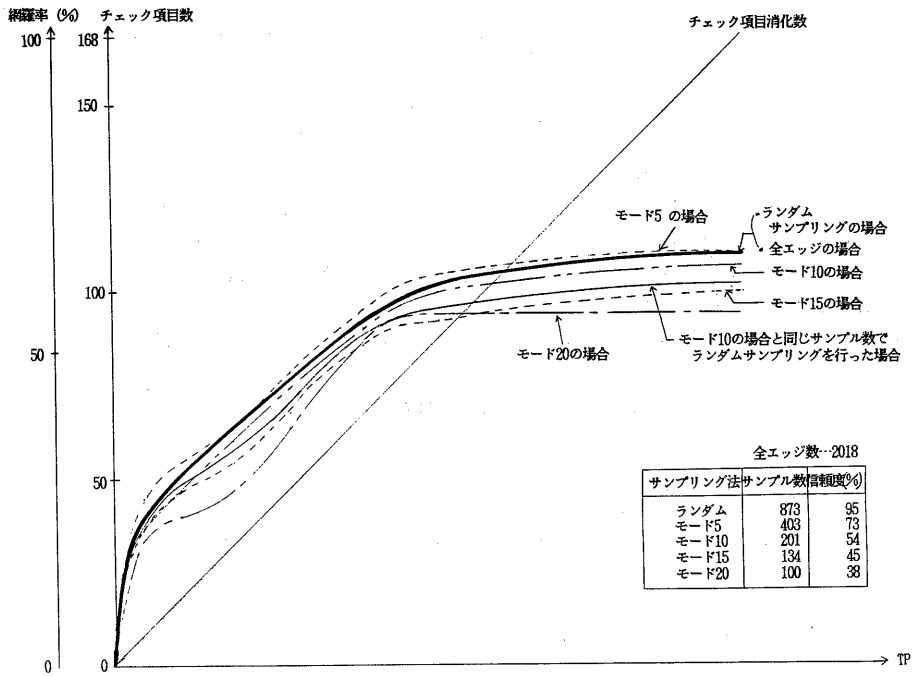


図 5.2-1 網羅率測定結果 (プログラムA)

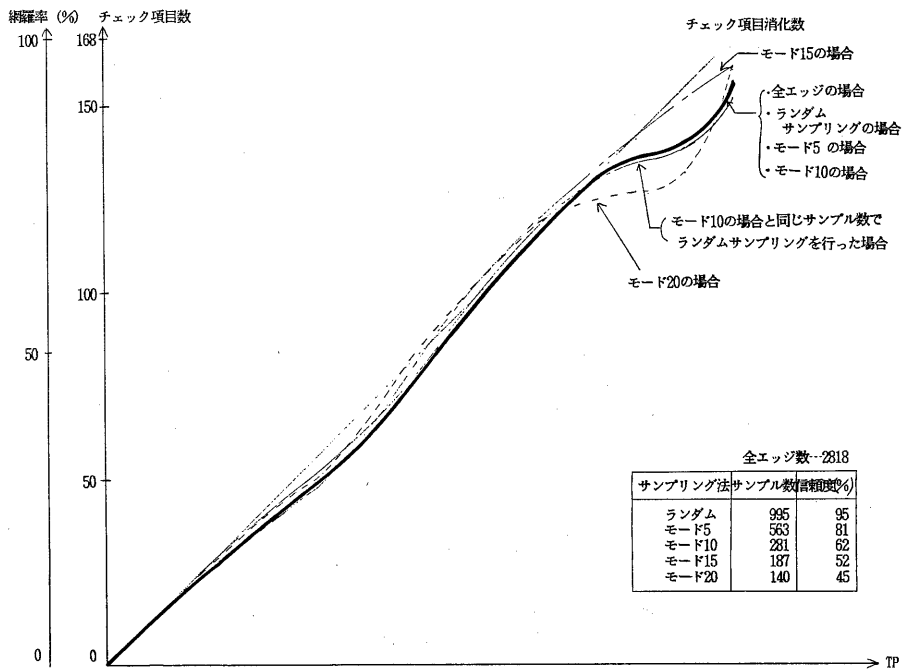


図 5.2-2 網羅率測定結果 (プログラムB)

(2) モードによるサンプリング

各プログラムについて、モードを変えて測定した。モード10程度であれば、2~3%程度の誤差であった。このときのサンプル数は、ランダム・サンプリングに比べて1/2 ~ 1/3程度である(図5.2-1, 図5.2-2)。

(3) 層別サンプリング

層別サンプリングを行うためには、まず、プログラムを適切な層に分割する必要がある。プログラムの最小構成単位であるルーチンに着目し、各ルーチン毎の網羅率とプログラムの種々の属性(CALL文の数、エッジ数、Cyclomatic Number⁽⁶⁾、CASE文の数等)との相関を調べることで網羅率を左右する要因(層の分割基準)を明らかにすることを試みた。

今回測定したプログラムについて、プログラムの特性を表すと考えられる項目を列挙し、網羅率との相関係数を求めた(図5.2-3)。

この結果、CALL文の数、Cyclomatic Number等と弱い相関があるが、層を識別できる程強い相関のあるものは見つからなかった。

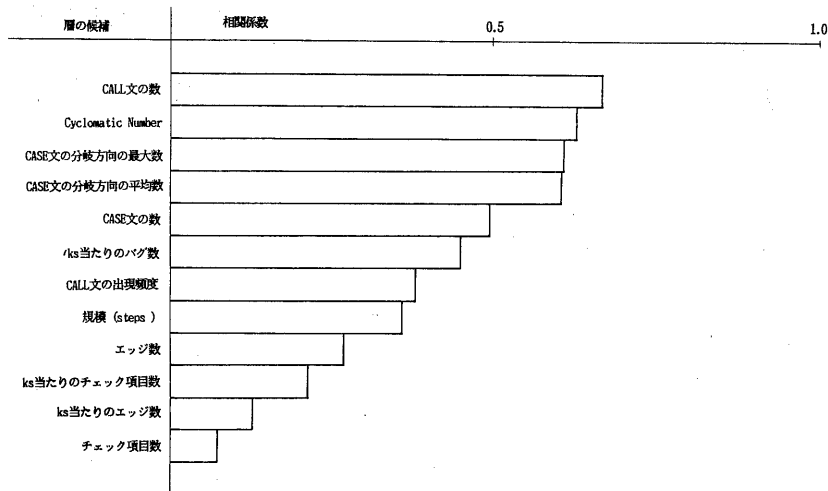


図 5.2-3 網羅率と層の候補との相関

6. 評価・考察

6.1 サンプリング法

(1) サンプリング法の適用性

統計上保証されるサンプル数を用いて測定を行った結果、サンプリングしない場合と比較して0.5%未満の誤差という高い精度が得られた。今回の実験により、サンプリングという統計手法がプログラムについても十分適用性のある有効な手法であるとの見通しを得た。

(2) サンプル数

ランダム・サンプリング、モードによるサンプリングをそれぞれ用いた場合のサンプル数を図6.1-1に示す。モードによるサンプリングでは、今回の試行においてランダム・サンプリングに比べサンプル数を1/2 ~ 1/3にしても同程度の精度で測定できた。この原因としては以下のように考察できる。

エッジには、独立性の高いエッジと、一つのエッジを通過した際そのエッジに従属して通過するエッジが存在する⁽⁷⁾。モードを用いて、エッジを一定間隔に抽出することにより、結果的にこの従属するエッジをある程度排除して互いに独立性の高いエッジをサンプリングしているものと考えられる。(つま

りエッジ間の従属性により実質的に母集団が全エッジより小さくなっており、少ないサンプル数で高い精度が得られた。)

同様に、同じサンプル数でランダム・サンプリングがモードによるサンプリングより 2~3 %精度が悪くなったのは、ランダム・サンプリングでは隣接したエッジ等従属関係の強いエッジを選択する可能性があり、モードによるサンプリングに比べサンプルに偏りが生じるためと推定される。この点に関しては、今後プログラムの構造・エッジの従属性とサンプルの取りかたの関係を明らかにし、必要なサンプル数/サンプルの抽出法を確立する必要がある。

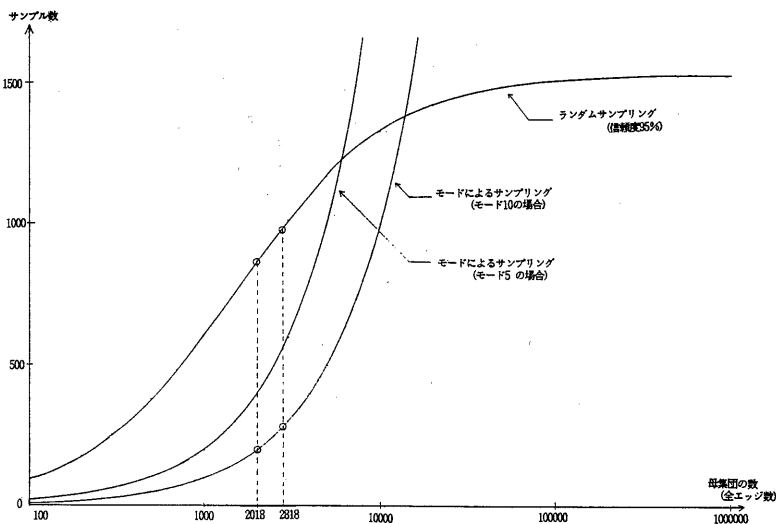


図 6.1-1 ランダム・サンプリングとモードによるサンプリングのサンプル数の比較

6.2 エッジ網羅率測定ツール (ROUTE-C) の効果

(1) 測定準備に要した工数の評価

本ツールを用いる際の測定準備に必要な工数の測定を行った。プログラムA, Bいずれの場合も、ツールの登録、不当命令の埋め込み、再立ち上げ、あわせて5分以内で収まり、実際のテスト工程においてもオーバーヘッドとなるには至らない。

(2) 測定工数の評価

本ツールを用いてエッジ網羅率の測定を行った際に要したのペマシンの評価を行った。この結果測定を行わない場合に比較して数%増で網羅率が測定できることがわかり、実際のテスト工程においても十分使用できると考えられる。

(3) DS評価

網羅率測定アルゴリズム、及びサンプリング法導入の効果を評価するために、ROUTE-CのDSの机上算定を行い、測定を行わない場合の対象プログラムのDSを1とし、従来方式との比較を行った。評価のモデルとして、100ks, 10ks のプログラムで 2.5steps に1個エッジが出現することを仮定した(図6.2-1)。

この結果、従来方式に比較して 1/20~ 1/50程度にDSを削減でき、本測定方法の有効性を確認できた。

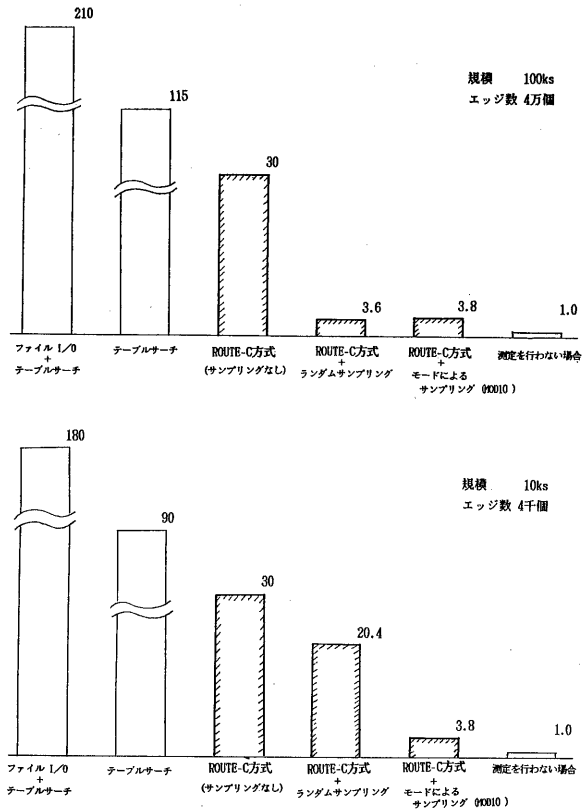


図 6.2-1 網羅率測定方式の DS 評価

(4) メモリ評価

エッジ 1 個につき 8 バイトのテーブル・エントリを必要とする。サンプリングを用いることにより、通過状況記録テーブルは 32 KB に収まり、測定時に特に問題にならないと考えられる。

7. おわりに

大規模プログラムのテスト網羅率の実用的な測定方法を確立することをねらいに、①サンプリング法の適用による測定対象エッジの削減法、②測定準備削減、測定時間短縮を実現する測定方法、のそれぞれについて提案し、実験による有効性の確認を行った。その結果、2つの手法を組み合わせることで、測定工数を大幅に削減できることが明らかになった。今回の実験ではモード10程度のサンプル数であれば網羅率を正確に測定できることがわかったが、さらに効率良くテストを進めるためにはモードの値をより大きくし、サンプル数をさらに削減しても網羅率を精度良く測定できる方法についての検討が必要である。

今後は以下の項目について留意しながら検討を進める予定である。

(1) サンプル数の削減法

- ① モードによるサンプリングの有効性を確認し、最適なモードの値を求めるためにプログラムの構造を解析し、独立なエッジの出現頻度について調査を進める。
- ② 今回の検討においては、複数のルーチンを束ねたものを層として分割することを試みたが、分割基準となる属性を明らかにすることはできなかった。今後は、網羅率と弱い相関のあった属性を幾つか組み合わせた分割基準、或いはエッジの到達難易度⁽⁸⁾等に着目し、分割基準/層のとりかたについて考察を進める。

(2) プログラムの変更管理法の検討

今回実現した測定方法の実用性をさらに高めるためには、プログラムのテスト開始時点から測定を開始しテストの進捗管理を可能とする必要がある。このために、ソース・プログラムとエッジの対応表或いは変更部分識別のためのアルゴリズム、テスト・データ管理法等の問題の解決が必要である。

(参考文献)

- (1) 杉田, 高村 「C1によるソフトウェアの品質保証」ソフトウェアシンポジウム80 論文集
- (2) 岡, 山口他 「テスト網羅性によるソフトウェア品質保証法の評価」第24回情処全国大会
- (3) 中屋 「プログラムの実行網羅性検証方法」第27回情処全国大会
- (4) 寺島, 細谷他 「システム製造用言語SYSL」研究実用化報告 vol.24 No.1 1975
- (5) 林, 伊土他 「ソフトウェアの品質推定のための Capture&Recapture 法支援ツール CREDIT-Q」
第27回情処全国大会
- (6) T.J.McCABE 「A Complexity Measure」IEEE Trans.on SE vol.SE-2, No.4 December 1976
- (7) 中所 「バステストに本質的な分岐に着目した網羅率尺度の提案」
情処学会論文誌 1982 .9月 vol.23, No.5
- (8) 伊土, 馬場 「バグ発見過程に着目したバグ抽出難易度に関する尺度」第26回情処全国大会