ソフトウェア工学 118－17
（1998．3．11）

残存フォールト数の推定が可能なソフトウエア試験法について（3）
－パスカバレジに基づいたソフトウエア信頼性モデル－

若杉忠男

若杉情報技術コンサルタントオフィス
連絡先：〒251　神奈川県藤沢市片瀬山3-11-1　電話0466-23-4832

抄録

フローグラフをパスに分解して解析するソフトウェア信頼性モデルを提案する．これを“パス分解法”と名付ける．プログラムに予めいくつかのチェックポイントを埋め込み，フォールトが混入されている位置から何回めのチェックで発見されたかによってエラーの長さを定義する．エラーはフローグラフのパスにそって伝わりチェックポイントで発見されるが，発見されるエラーの長さは幾何分布にしたがう．エラーを発見して除去すると，一般に発見場所以降のパスのそのエラーの発生確率は0になる．前提条件を変えていくつかの信頼性モデルを考え，それらのモデルのパスカバレジ試験のエラー発見率，エラー数の推定法などを述べる．

On a software test method capable of estimating the number of programming errors (3)
- software reliability models based on path coverage -

Tadao WAKASUGI, Member
WAKASUGI Information Technology Consultant Office
3-11-1 Kataseyama Fujisawa-city, 2510033 Japan

Abstract

New software reliability models on the basis of numbers of path of flowgraph named "path decomposition method" are proposed. In a program, some check points are set beforehand, and the length of the errors is defined as the number of check times the errors have been checked before they are detected. Probability of length of the detected errors becomes a geometric distribution. Based on the idea, some variations of the software reliability models are introduced, and error detection rate of path coverage test, and estimation method of the number of errors are discussed.

1　はじめに

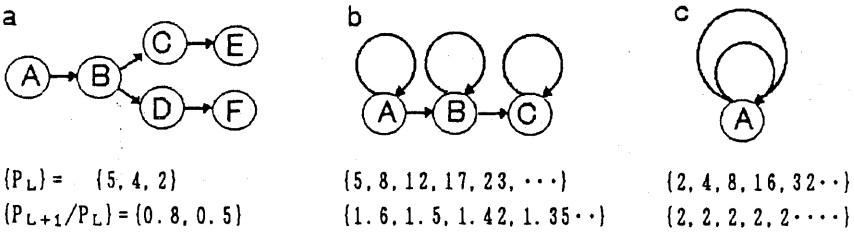
ソフトウェア信頼性モデルにはいろいろのアイデアがあるが，その多くは作業経過時間に対する発見バグ累積数に適当に曲線を当てはめるものである[1][2][3]．当てはめに使う曲線には，指数型，ゴンペルツ曲線，遅延S字，ワイブル曲線などがあり，またこれらを統合した理論も展開されているが[4]，独立変数である作業経過時間の計測があいまいなのが欠点である．そのためにテストインスタンスという作業単位を独立変数に採用した方法も提案されている[5]．

筆者はプログラム内に予めチェックポイントを配置し，発見されたエラーを，それがいくつのチェックポイントを経由してから発見されたかによってエラーの長さを定義し，チェックポイントでのエラー発見率が一定という前提で理論を構築した．これによりパスカバレジ試験に結び付いた信頼性モデルを考え，プログラムの複雑さや試験の質をモデルに

反映させた．以降では２章でこの理論の基礎となるパスベクトルの性質について述べ，３章でこの信頼性モデルの基本条件を説明し，４章と５章で信頼性モデルを紹介し，６章でこの考えに基づくエラー数の推定法を示し，７章でまとめを述べる．

２．フローグラフのパスベクトル表現

フローグラフはいくつかのノードとそれを結ぶリンクから構成される．連続するＬ個のリンクを長さＬのパスと呼び，長さＬのパスの個数をP_Lと記述する．これをベクトル形式で｛P_L｝と記述しパスベクトルと呼ぶ．このP_Lが大きいほどプログラム規模が大きくまた複雑であると考える．またP_{L+1}/P_Lをパスの増加率と呼び，同様にベクトル形式で表す．図１に，フローグラフとそれに対するパスベクトルなどの例を示す．



$\{P_L\}$ = {5, 4, 2}　　{5, 8, 12, 17, 23, ･･･}　　{2, 4, 8, 16, 32･･}

$\{P_{L+1}/P_L\}$ = {0.8, 0.5}　　{1.6, 1.5, 1.42, 1.35･･}　　{2, 2, 2, 2, 2････}

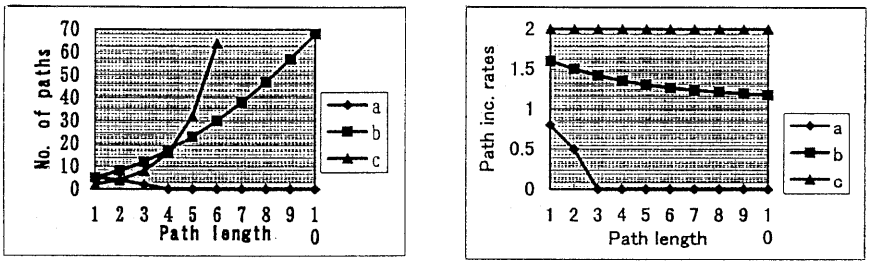


Fig. 1　Examples of flowgraphs, the graphs of their path vectors $\{P_L\}$ and increasing rate vectors $\{P_{L+1}/P_L\}$.

図１　フローグラフとそのパスベクトル，パス増加率の例

フローグラフの｛P_L｝は，次の３つの場合に分類される[6]．すなわち，Ｌ→大とすると，フローグラフが

（１）ループを含まない場合には$P_L \to 0$になる．図１ａの例がこれで，Ｌが４以上のとき$P_L = 0$である．

（２）ループを含むがループが互いに共通のノードをもたない場合にはP_LはＬの多項式になる．したがって$P_{L+1}/P_L \to 1$となる．図１ｂの場合がその例で$P_L = 0.5L^2 + 1.5L/2 + 3$となる．

（３）ループを含みそのループが互いに共通のノードをもつ場合にはP_LはＬの指数関数で増加する．したがってP_{L+1}/P_Lは１より大きな値となる．図１ｃの場合がその例で，$P_L = 2^L$で$P_{L+1}/P_L = 2$である．

これらの証明は別論文[6]などにゆずり，ここでは簡単な説明にとどめる．

（１）は自明である．（２）については，パスのうち最も多くのループを含むものを考え，そのループの数をＫ個とする．Ｋ個のループを含む長さＬのパスの個数は，ループの間をつなぐリンクを除いて考えると，Ｋ種の中から順序を考慮せずに重複を許してＬ個を

選択する組合せの数である．これは${}_{K}H_{L}$で，したがってＬのＫ－１次式となる．それ以外のパスはループの数がＫ－１個以下であるからパス数はＫ－２次式以下となる．また（３）については，一つのノードをＫ個のループが共有すると，そのノードで選択できるパスの数はＫ個となる．一般にフローグラフ全体のノードの数をＮ，一つのノードから出るリンクの本数の最大数をＫとすると，長さＬのパスの数は$N \times K^{L}$以下となりＬの指数関数となる．

3　信頼性モデルとパスカバレジ

ここではプログラムの間違った作り方をフォールトと呼び，その結果生じた予期しない現象をエラーと呼び区別する．反対の呼び方をする資料もあるが[2]，本論文ではフォールトは原因でエラーはその結果とする．試験項目のパスを水道にたとえると，上流での異物の混入がフォールトで，下流で検出された汚染がエラーである．

まず，前提条件として次の仮定をする．

前提１：プログラムをフローグラフで表し，そのノード点をチェックポイントとする．チェックポイントとは，たとえば変数のダンプリストの出力などの仕掛をした位置で，ここでデータをチェックしてエラーを見つける．各チェックポイントにおけるエラー発見率（チェックポイントを通過するエラーのうち，そこで発見できるエラーの割合）の平均値をｒと表し，以降では**試験能力**と呼ぶ．$0 \leq r \leq 1$である．

前提２：リンク内にフォールトがあった場合，その結果であるエラーはそのリンクの下流のパスのチェックポイントのどこかに現れる．

以上の仮定から，発見されるエラーの長さは，たとえばサイコロを振って６の目が何回目に出るかという確率と同じく幾何分布にしたがうことが言える．

ここでパスカバレジ試験を考える．パスカバレジ試験の定義として，ここではすべてのリンクを一通り試験をすることを長さ１のパスカバレジ試験と呼び，連続するＬ個のリンクの組合せすべてを試験することを長さＬのパスカバレジ試験という．長さ１のパスカバレジ試験がステートメントカバレジ試験で，長さ２がブランチカバレジ試験にあたる[6][7][8]．

フローに枝別れのある場合の試験について，エラーの出現と除去の関係によって次の３つのモデルを考え，前提３とする．

前提３：

モデル１：あるリンクに存在するフォールトは下流のパスのすべてにエラーを生じ，それを見つけて修正するとそのフォールトによるエラーはすべて消える．

モデル２：あるリンクに存在するフォールトは下流のパスのどれか一本にだけエラーを生じ，それを見つけて修正すると，そのフォールトのエラーはすべてなくなる．

モデル３：あるリンクに存在するフォールトは下流のパスのすべてにエラーを生じ，それを見つけて修正すると，そのエラー発見場所の下流については発見したフォールトによるエラーはすべて消える．しかしエラー発見場所の下流以外についてはフォールトの影響は消えない．図１のａの例で，Ａ→Ｂ間にあるフォールトをＣで発見して修正するとＣ以降のパスからはエラーは除去されるが，Ｄ以降にはエラーが残っているとする．あるパスでエラーを除去しても，その下流以外のパスのエラーは除かれないと考える．

水道にたとえて説明すると，（１）は上流で病原菌が混入したのを下流で検出し，混入場所を突き止めてふさぐことに当たる．（２）は上流で１個の異物が混入したのを下流で網をはって取り除くという例であり，これを浮遊機雷モデル，略して機雷モデルと呼ぶ．

（３）は病原菌を検出したので検出場所で消毒剤を混ぜて下流への汚染を防いだが他の支流には消毒剤はまかないというような例である．これを水道汚染モデル，略して汚染モデルと呼ぶ．

図１ａのＡ→Ｂ間にフォールトがあったとして，長さ２までのパスをすべて試験した場合の長さ３のパスに与える影響，すなわち長さ３のパスへのエラーの見逃し率を考える．

モデル１のエラーの見逃し率は，Ｂ，Ｃ，Ｄの３地点すべてで見逃す確率で，$(1-r)^3$となる．モデル２の機雷モデルの場合は，Ｂで見逃しかつＣまたはＤでまた見逃す確率でエラーがＢ→ＣまたはＢ→Ｄで生ずる確率をそれぞれ1/2として，$(1-r)\times\{1/2\times(1-r)+1/2\times(1-r)\}=(1-r)^2$となる．モデル３の汚染モデルの場合はＢ→ＣとＢ→Ｄの両方に確率1で伝播するとして，$(1-r)\times\{(1-r)+(1-r)\}=2\times(1-r)^2$となる．したがって長さＬのパスの数を$P_L$とすれば，見逃し確率は$P_L\times(1-r)^L$となり，このモデルでは，エラーの数は枝別れの数に比例して増加する．

モデル１は楽観的過ぎて現実と合わないと考えられるので，以降では機雷モデルと汚染モデルについてのみ考える．

4　機雷モデル

このモデルはフォールトとエラーは１対１なので単純で扱いやすい．この仮定のもとで次の定理が成り立つ．

定理１

機雷モデルの条件のもとで，

（１）長さＬまでのすべてのパスを試験項目でカバーすると，長さＬのエラーの発見率Z_Lおよびその見逃し率S_Lは次の式で求められる．

$$Z_L=(1-r)^{L-1}\times r \qquad (1)$$

$$S_L=(1-r)^{L} \qquad (2)$$

$$S_{L-1}=S_L+Z_L \qquad (3)$$

（２）長さＬのパスを試験項目で100%カバーして発見したエラー数をE_L，全フォールト数をＦ個とすると，次の式が成立つ．

$$E_L=F\times Z_L=F\times(1-r)^{L-1}\times r \qquad (4)$$

$$\text{エラー累積数}\ Y_L=E_1+E_2+\cdots+E_L=F\times(1-S_L) \qquad (5)$$

（３）全フォールト数Ｆと試験能力ｒとは次の式で求められる．

$$r=1-E_2/E_1 \qquad (6)$$

$$F=E_1/r \qquad (7)$$

（４）試験でカバーするパスの長さを増やしてゆくと発見エラー数は単調に減少し，残存エラー数は０に近付く．すなわち，

$$E_1>E_2>E_3>\cdots\rightarrow 0 \qquad (8)$$

証明

(6)式は，(4)式のＬに１と２を代入して連立させてＦを消去して得られる．(7)式は(5)式でＬ＝１として得られる．また，(8)式は，(4)式で$F\times(1-r)^{L-1}\times r=E_L$と，$F\times(1-r)^{L}\times r=E_{L+1}$を連立させて，$0<1-r=E_{L+1}/E_L<1$から得られる．他の証明は省略する．

ＱＥＤ．

(5)式でＬ→∞とすると$\Sigma E_L=F$となるが，これはフォールトの個数と発見したエラーの個数が等しいということを示し，このモデルの前提条件から当然である．

5　汚染モデル

機雷モデルより悲観的と思われる汚染モデルを考察する．このモデルではエラーはフローの分岐点があると増殖する．まず定理１に対応する汚染モデルの性質を考察する．

定理２

汚染モデルの条件のもとで，次の式が成立する．

（１）長さＬまでのすべてのパスを試験項目でカバーすると，エラー発見率Z_Lとエラー見逃し率S_Lは，P_Lを長さＬのパスの個数として

$$Z_L = P_L \times (1-r)^{L-1} \times r / P_1 \quad (11)$$

$$S_L = P_L \times (1-r)^{L} / P_1 \quad (12)$$

$$S_{L-1} = S_L + Z_L \quad (13)$$

となる．

（２）長さＬのパスを100%カバーして発見したエラー数E_Lについて，全フォールト数をＦとすると，次式が成り立つ．

$$E_L = F \times Z_L = F \times P_L \times (1-r)^{L-1} \times r / P_1 \quad (14)$$

（３）Ｆとｒは次式から求められる．

$$r = 1 - (E_2 / E_1) / (P_2 / P_1) \quad (15)$$

$$F = E_1 / r \quad (16)$$

証明

汚染モデルでは，エラーの数はパスの個数に比例するので，機雷モデルのZ_Lにフローグラフのパス数P_Lを掛けて(11)(12)式が得られる．(15)(16)式は，(14)式で特にＬ＝１，２として，$E_1 = F \times r$，$E_2 = F \times P_2 \times (1-r) \times r / P_1$から得られる．

ＱＥＤ．

(11)(12)式でP_1で割ったのは，定理１の(1)(2)のＬ＝１の場合と揃えるためである．発見エラー数E_Lの(14)式を少し変形して$E_L = (F / P_1) \times P_L \times (1-r)^{L-1} \times r$とすると，$F / P_1$は試験前のフォールトの密度を表し，$P_L$はフローグラフの複雑さを表し，残りの項は試験能力を表す．一方機雷モデルの発見エラーの式(4)にはP_Lの項がなく，したがってプログラムの複雑さが考慮されていない．

(15)式で，ｒが一定のときは，P_2 / P_1が大きいほど，すなわちパスの増加率が大きいほどE_2 / E_1すなわちエラーの発見増加率が下がる．もし$P_2 / P_1 < E_2 / E_1$，すなわちパス数の増加率以上に発見されるエラーの増加率が多い場合には，ｒはマイナスとなる．これは何か異常が発生したと解釈すべきである．また特に$P_{L+1} / P_L = 1$が常に成立するときは，ｒは機雷モデルの(6)式と同じになる．

次に，汚染モデルによる見逃しエラーの収束条件を考察する．機雷モデルでは無条件に収束するが，汚染モデルでは無条件ではない．

系２．１

汚染モデルのパスカバレジ試験でＬ→∞のときに$P_{L+1} / P_L \rightarrow C$とすると，見逃しエラーを０にするためには試験能力ｒは

$$r > (1 - 1 / C) \quad (17)$$

でなければならない．$0 < C \leq 1$の場合には，任意のｒで収束する．

証明

(12)式のエラー見逃し率S_Lについて，正項級数ΣS_Lが収束するにはダランベールの定理によるとあるＢが存在して$S_{L+1} / S_L \rightarrow B < 1$であればよい．$S_L = P_L \times (1-r)^{L} /$

P_1であるから，この条件は$0 \leqq P_{L+1} \times (1-r) / P_L = C \times (1-r) < 1$となる．これから式(17)が得られる．

(17)式で$0 < C \leqq 1$の場合には，$1 - 1/C$は0またはマイナスとなり，$r \geqq 0$であるから無条件で成立する．

ＱＥＤ．

２章に述べたように，フローグラフは次の３つのクラスに分類でき，したがって汚染モデルでは系２．１から次の系が導かれる．

系２．２

汚染モデルについては，$L \to$大の場合，フローグラフが

（１）ループを含まない場合には$P_L \to 0$であり，したがって100%パスカバレジ試験ができる．

（２）ループを含むがループが互いに共通のノードをもたない場合には，$P_{L+1} / P_L \to 1$であり，したがってパスカバレジ試験によりエラー数は無条件で0に収束する．

（３）ループを含みそれが共通のノードをもつ場合には，$P_{L+1} / P_L \to C > 1$である．したがって試験能力$r > (1 - 1/C)$でなければ試験してもエラーは0に収束しない．

証明省略．

これによって，図１ａの場合には１００％カバレジ試験ができ，図１ｂの場合には試験でエラーを0にでき，図１ｃの場合は，$P_L = 2^L$で$C \to 2$なので，$r > 1/2$でなければ試験してもエラーを0にはできないことになる．

6　モデルの適用について

実際にこれらのモデルでエラー数を推定するには，E_1とE_2を求めなければならないが，次の仮定をすればE_1とE_2を一部のパスをカバーするだけで推定できる．

前提４：長さＬのエラーの発見数は，長さＬのパスを試験項目がカバーする割合に比例する．

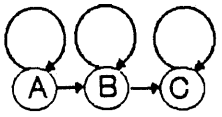
たとえば，長さＬのパスを試験項目がａ％カバーしていれば，

E_Lの推定値＝試験項目で得られた長さＬのエラー数×100／ａ　　(18)

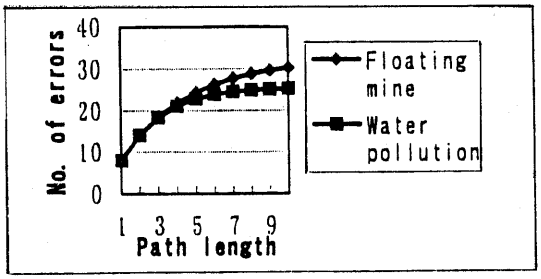
となる．これによってE_1，E_2を推定し，それを使って機雷モデルの(4)(6)(7)式や汚染モデルの(14)(15)(16)式から$\{E_L\}$全体を推定できる．

ここで具体的な例でエラー数の推定方法を説明する．図２に示すフローグラフに，Ａ→Ａ→Ｂ→Ｂ→Ｃ→Ｃという長さ５の試験項目を適用する場合を考える．このフローグラフとそれをカバーする試験項目ののパスベクトルは，それぞれ図２のように表される．そのカバレジを考えると，フローグラフの長さ１のパスは100%カバーしているが，長さ２のパスは８個のうち試験項目はＡ→Ｂ→Ｃというパスなどを試験していないので，４／８＝50％のカバレジである．したがってこの試験項目による各長さ別のパスカバレジは，図２のカバレジベクトルのようになる．

いま仮にこの試験項目によって，長さ別のエラーが｛８，３，・，・｝のように得られたとすると，前提４により，長さ１のエラーは100%カバーしているからエラーはもうないが，長さ２のエラーは50%カバーして３個見つけたからもう３個あるだろうと推定する．こうして得られた推定値E_1，E_2から，長さ３以上のエラーを推定する．機雷モデルでは(6)(7)式から$r = 0.25$，$F = 32$となる．エラー数はフォールト数と同じ32である．同様にして汚染モデルでは，$r = 0.53$，$F = 15.1$，エラー総数は$L = 15$までE_Lを加えると25.4となる．



Test case
A → A → B → B → C → C



Path vector of the flowgraph: A {5, 8, 12, 17, 23, ・・}
Path vector of the test case: B {5, 4, 3, 2, 1, ・・}
Coverage of the test case: C=B/A {1, 0.5, ・, ・, ・・・・}
Detected error numbers: D {8, 3, ?, ?, ?, ・・}
Estimated error No. by Prep.4: D/C {8, 6, ?, ?, ?, ・・}
Estimated error No. by mine model {8, 6, 4.5, 3.4, 2.5, ・・} Total 32
Estimated error No. by pol. model {8, 6, 4.2, 2.8, 1.8, ・・} Total 25.4

Fig. 2 A sample of path coverage testing about the flowgraph of fig. 1b

図2　図1 bのフローグラフについてのパスカバレジテスト適用例

Fとエラー数の差約10は枝別れによるエラーの増殖である．汚染モデルではエラーが枝別れにしたがって増殖するので機雷モデルよりもエラーの数が多くなると考えられるが，この例ではかえって少ないという結果になった．図2にこれらの比較グラフを示す．

汚染モデルでP_LがLの指数関数となる場合は，$P_L = a \times b^L$とすると，(14)式から

$$E_1 = F \times r$$

$$E_2 = F \times r \times b \times (1 - r) = E_1 \times b \times (1 - r)$$

となり，一般に

$$E_L = E_1 \times \{b \times (1 - r)\}^{L-1} = E_1 \times (E_2 / E_1)^{L-1} \qquad (19)$$

となる．

(19)式はすべてのE_LがE_1とE_2で決まることを示す．また$P_L = 1 = 1^L$も指数関数であるから，(19)式は機雷モデルでも成立する．したがって次の定理が成り立つ．

定理3

機雷モデルおよびP_LがLの指数関数となる場合の汚染モデルについては，長さLのエラー数について次の関係が成立する．

$$E_L = E_1 \times (E_2 / E_1)^{L-1} \qquad (19)$$

したがってエラー数が0に収束するためには，$E_2 / E_1 < 1$でなければならず，その場合には数列E_Lは等比級数となり，その和（＝エラー総数）は次の式で表される．

$$\text{エラー総数} = E_1^2 / (E_1 - E_2) \qquad (20)$$

証明省略

図2の例について，$E_1 = 8$，$E_2 = 6$を(20)式に代入するとエラー総数＝32となり，図2の機雷モデルで得た値と一致する．E_LはE_1とE_2のみで決まるので，Lを横軸にエラー数をたて軸にした信頼度成長曲線を描くと，機雷モデルの場合，また汚染モデルでP_Lが指数関数で近似できる場合には，E_1とE_2さえ決まれば，フォールト数Fや試験能力rやフローグラフの形に関係なく一本の同じカーブとなる．

こうして，本理論によれば，わずかな試験項目で無限のループを含むカバレジ試験のエラー数の推定ができることになる．

7　まとめ

従来の信頼性モデルは統計的分析によるアプローチであるのに対し，ここで述べた信頼性モデルはソフトウェア試験のシミュレーション的アプローチであり，これを“パス分解法”と名付ける．一般に信頼性モデルでは横軸を作業時間とするが，このモデルではパスの長さを横軸とし，たとえていえば，波形を時系列としてではなく振動数に分解するようなものである．

その成果は次のとおりである．

（１）エラー数の見積とパスカバレジ試験とを結び付け

（２）作業時間というあいまいな数値をパス数という具体的なものに変え

（３）信頼性モデルにプログラムの複雑さを反映させ

（４）したがって少ない試験項目でエラー数の精度のよい見積ができることが期待でき

（５）エラー数の簡単な見積式を導いた．

この方法を実際の試験に適用するためには，ｒが一定になるようにチェックポイントをどう設定するか，いくつかのフォールトの複合で起きるエラーについてはその長さをどう数えるか，またエラー数の見積の精度はどのくらいかなど多くの問題が残されている．今後はこれらの問題を解決して，理論の具体例による検証を行う予定である．

参考文献

[1] 大場充：ソフトウェア信頼性モデル入門，情報処理，Vol.31,No.12,pp1623-1630,(1990).

[2] 尾崎俊治：非定常ポアソン過程モデル，情報処理，Vol.31,No.12,pp1631-1640,(1990).

[3] 山田茂：ソフトウェア信頼性モデル－基礎と応用，P202，日本科学技術連盟（1994）.

[4] 古山恒夫：ソフトウェア信頼性成長モデルに関する統合モデルの解析的パラメータ推定法，情報処理学会論文誌，Vol.37,No.12,pp2326-2333(1996).

[5] Tohma Y. et al: Structural Approach to the Estimation of the Number of Residual Software Faults Based on the Hyper-Geometric Distribution, IEEE Soft. Eng. SE-15,No.3, pp345-355(1989).

[6] 若杉忠男：残存エラー数の推定が可能なソフトウェア試験法について（２）－パス分解法と構造化プログラミングとの関係－，情報処理学会ソフトウェア工学研究会報告97-SE-114,PP57-64(1997年).

[7] Beizer,B.: Software Testing Techniques, P.442,小野間,山浦訳,日経ＢＰ出版センター（1994）.

[8] 若杉忠男：ＯＳＩ適合性試験スイートの評価法－マルチトランジション試験，電子情報通信学会論文誌，VOL.J72-BI No.4,pp159-155（1996）.

[9] 山田，尾崎：ソフトウェア信頼度成長モデルとその比較，電子通信学会論文誌,Vol.J65D, No.7, pp 906-912(1982).