

フローグラフや状態遷移図の特性のパス数による分析

若 杉 忠 男†

フローグラフや状態遷移図の L 個の連続したリンクを長さ L のパスと定義し、長さ L のパスの個数を P_L 、それをベクトル状に並べた $\{P_L\}$ をパスベクトル、 P_{L+1}/P_L を P_L の増加率と呼ぶ。パスベクトル $\{P_L\}$ は連結行列によって求められる。これはフローグラフや状態遷移図の性質を表し、構造化プログラミングやサイクロマチック数と関連付けられる。またパスベクトルのうち P_1 はフローグラフの規模、 P_2 は複雑さ、また増加率の極限值はパスカバレッジ試験の困難さを表すと考えられる。フローグラフや状態遷移図はそれに含まれるループの性質によって次の3つに分類できる。 L を大にしたとき、(1) ループを含まない場合には P_L は 0 となり、(2) ループを含むがそのループが互いに共通のノードを持たない場合には P_L は L のべき乗のオーダーで増加し、(3) 共通のノードを持つループを含む場合は L の指数のオーダーで増加する。逆に、 L を大にしたときに P_L の増加率が 1 に近付けばループは共通のノードを持たず、1 より大な数値に近付けば共通のノードを持つループを含む。パスベクトルの応用例として、プログラムの構造を単純化するためのガイドラインを導き、構造化プログラミングの理論と比較する。またトランスポートプロトコルの複雑度を評価する。

On Analysis of Characteristics of Flowgraphs and State Transition Diagrams by Numbers of Paths

TADA0 WAKASUGI†

Concerning the flowgraphs and state transition diagrams, a sequence of consecutive links of L is defined as a path of length L , and the number of paths of length L is described as P_L . Path vector $\{P_L\}$ means sequence of P_L , and the increasing rate of P_L means P_{L+1}/P_L . Path vector $\{P_L\}$ is calculated by connection matrixes. P_L represents the characteristics of flowgraphs and state transition diagrams. For example, P_1 represents volume of them, P_2 represents the degree of complexity, and the limit of increasing rate represents the difficulty of path coverage testing. Flowgraphs and state transition diagrams are classified in 3 classes depending on loops contained. When L increases, (1) if no loop is contained then P_L becomes 0, (2) if loops have no common node then P_L increases at a rate proportional to a power of L , and (3) if loops have common nodes then P_L increases at a rate proportional to an exponential of L . To show the usefulness of the concepts of path vector, some samples of program structure, and the communication protocols are discussed.

1. はじめに

プログラムの複雑度の指標については多くの論文が発表されている。最も単純な指標としてステートメント数があるが、これはプログラムのボリュームを表すだけで複雑さは表現していないと考えられる。プログラムの構造の複雑さを示す尺度として、McCabe¹⁾のサイクロマチック数がある。これは式(1)のようにリンクとノードの差だけが評価の対象となり、プログラムの規模は示さない。

$$\text{サイクロマチック数} = (\text{リンク数}) - (\text{ノード数}) + 2 \quad (1)$$

より詳細に分析した指標として、プログラムをオペレータとオペランドに分類して数える指標などが提案されている^{2),3)}。

これらのほかにもプログラムの性質を表す指標は多く、31種もの指標を列挙し比較している文献もあり⁴⁾、開発必要工数や発見されたエラー数などと関連付けてその妥当性の裏付けが試みられているが、人の体形を表すのに身長や体重、胸囲など多くの数値が必要のように、プログラムの特性を表すにも工数やエラー数の予測など、目的別にいくつかの数値が必要であり、まずプログラムのメトリックとはどうあるべきか、そのフレームワークから考え直そうという検討もなされて

† 若杉情報技術コンサルタントオフィス
WAKASUGI Information Technology Consultant Office

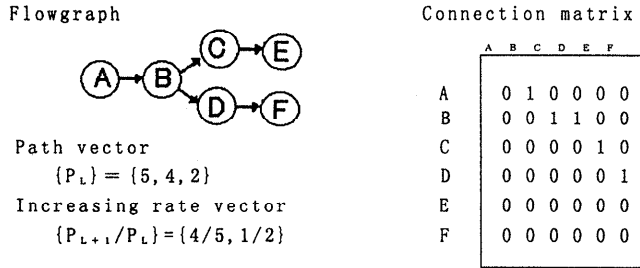


図1 ループを持たないフローグラフとその連結行列, $\{P_L\}$, $\{P_{L+1}/P_L\}$ の一例
 Fig. 1 An example of a flowgraph contains no loop, its connection matrix, path vector $\{P_L\}$ and increasing rate vector $\{P_{L+1}/P_L\}$.

いる^{5),6)}。

ここでは、プログラム（またはプロトコル）の特性をフローグラフ（または状態遷移図）のパスの数で表すことを考える。パスの数にこだわる理由はパスカバレッジテストと結び付けることにある。パスカバレッジテストは古くからよく使用されており^{7),8)}、プログラムのすべてのパスを試験項目でカバーして確認すればエラーはなくなるとする。したがって、パスの数は試験の手間やエラーの多さに関係すると考える。

しかしプログラムにループがあればパスの長さは無限になり、ブランチが多ければパスの数は急激に増加する。本論文は、パスの数がプログラムの構造によってどう変わるかについて研究し、パスベクトルという評価指標を提案し、実際問題へ適用する。ただし、ここではパスと試験の関係の考察は別の論文にゆずり⁹⁾、フローグラフとパス数の関係の分析に限定する。

まず、2章でパスベクトルの定義とその基本的な性質を説明し、3章ではパスベクトルとサイクロマチック数などとの関係を述べ、4章でループとパスベクトルの関係について分析し、5章でパスベクトルの応用例を示し、6章でまとめと今後の展望を述べる。

2. パスとフローグラフ

フローグラフはノードとそれをつなぐリンクからなる。状態遷移図では、ノードは1つの状態、リンクは状態の遷移を表す。一方プログラムの場合には、リンクはセグメント（分岐点や合流点を含まない一連のステートメント）を表し、ノードはセグメントとセグメントの間の分岐点や合流点を表す。プログラムのセグメントをリンクでなくノードで表現するフローグラフもよく使われるが、本論文では文献⁷⁾に従い前者をとる。その方が状態遷移図の書き方と一致するからである。

まず前提として次の定義をする。

前提：フローグラフのパスとは、任意のノードから出

発し任意のノードで終わる連続したリンクのシーケンスをいう。パスを構成するリンクの個数をパスの長さといい、長さ L のパスの個数を P_L と記述する。また P_{L+1}/P_L を P_L の増加率と呼ぶ。

図1のフローグラフのパス数 P_L は、ノードAからノードBに行くパスをA→Bと表すと、

- 長さ1のパス：A→B, B→C, B→D, C→E, D→Fの5本
- 長さ2のパス：A→B→C, A→B→D, B→C→E, B→D→Fの4本
- 長さ3のパス：A→B→C→E, A→B→D→Fの2本

となる。上記のフローグラフのパスの個数をベクトルの形で $\{P_L\} = \{5, 4, 2\}$ と記述し、パスベクトルという。

フローグラフはまた連結行列でも表される。連結行列とは図1に示すように、フローグラフのノードの数が N 個の場合 $N \times N$ のマトリックスで、各 i, j 要素はノード i からノード j へのリンクの数を表す。 i, j 要素が0ということとは、 i と j がリンクでつながっていないことを示す。こうして作った行列を M とし、このマトリックスの i, j 要素を m_{ij} と表すと、行列 $M \times M$ の各要素は、

$$\sum_{k=1}^N m_{ik} \times m_{kj} \tag{2}$$

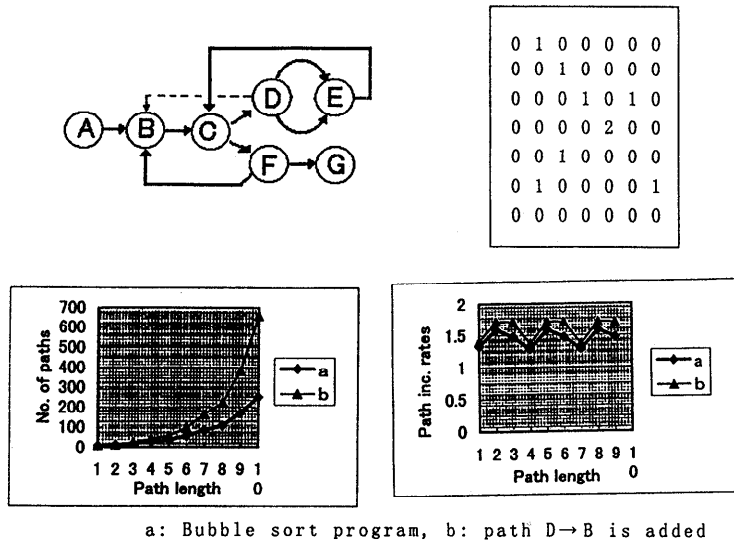
となり、これはノード i からノード j に長さ2で遷移するパスの個数を表す。同様にして行列 M^L の要素の合計は長さ L のパスの総数を表す。

P_L については次の定理が成立する。

定理1：フローグラフについて、ノードの数を N 、1つのノードから出るリンクの最大本数を K とすると、次の式が成り立つ。

$$P_L \leq N \times K^L \tag{3}$$

証明：1つのノードから出るリンクの数の最大を K



a: Bubble sort program, b: path D→B is added
 図2 バブルソートのフローグラフとその連結行列, $\{P_L\}$, $\{P_{L+1}/P_L\}$ の例
 Fig. 2 An example of a flowgraph of bubble sort, its connection matrix, path vector $\{P_L\}$ and increasing rate vector $\{P_{L+1}/P_L\}$.

とすると、1つのノードから出る長さ L のパスは K^L 個以下である。したがって、長さ L のパスの全体は $N \times K^L$ 以下である。 QED

3. フローグラフの特性とパス

パスベクトル $\{P_L\}$ の第1項 P_1 はセグメントの総数である。セグメントが多ければステートメント数も多いので P_1 はプログラムのボリュームを示すと考える。またブランチをノードから出るリンクの総数と定義すれば、 P_2 はまん中にノードを持つパスの数であるからすべてのブランチを含み、プログラム構造の複雑さを表すと考える。 P_1 と P_2 はそれぞれパスカバレジ試験でいうステートメントカバレジとブランチカバレジ試験に必要なチェック項目の数を表す指標となる。

パスベクトルの実例としてバブルソートを考える。そのフローグラフは図2の実線の部分である。パスベクトルは、定理1から $P_L \leq 7 \times 2^L$ となるがこの見積値は大きすぎる。より正確に求めるために P_{L+1}/P_L を計算し、 $L \rightarrow \infty$ としたときの値を求めると、図2のグラフの曲線 a に示すようにサイクル3で振動する。その3つの値の幾何平均は1.44となる。それを使って $P_L = P_1 \times 1.44^{L-1} = 9 \times 1.44^{L-1} = 6.25 \times 1.44^L$ として連結行列で求めた値と比較すると、 $L = 1 \sim 10$ の範囲ではその誤差は10%以内である。

図2のフローグラフはGO_TO レスプログラミングの条件を満たしているが¹⁰⁾、ここで図2の点線に

示す D→B というリンクを入れると GO_TO レスプログラミングでなくなる。そうすると図2のグラフの曲線 b に示すように P_L も P_{L+1}/P_L の値も増加する。先と同じように連続する3つの P_{L+1}/P_L の幾何平均を求めると1.59となる。これによりプログラムにGO_TO 文を1本挿入するだけでパス数が増加することが分かる。増加率の幾何平均は、これが大きいと試験しなければならない条件の組合せの数が増え、したがって、プログラムの試験のしにくさ、たちの悪さを示す数値であると考えられる。ここで次の定義をする。

定義: P_L の増加率 P_{L+1}/P_L の極限值を“ P_L の増加率の極限”と呼ぶ。誤解を招かない場合には単に極限と呼ぶ。ただし増加率がサイクリックな値をとる場合にはその一連の極限值の幾何平均値とする。

先に述べたサイクロマチック数の式を連結行列で表現すると、

$$\begin{aligned} & \text{サイクロマチック数} \\ &= \text{連結行列の要素の和} - \text{連結行列の行数} + 2 \end{aligned} \tag{4}$$

となる。この値が小さいということは、行列の0要素が多いことである。0要素の多い行列はべき乗してもその要素の数の増加が少ないから、 P_L の増加率の極限の値は小さくなる。図2の例で P_L の増加率の極限とサイクロマチック数を比較すると、D→Bのパスのない場合にはそれぞれ1.44と4となり、D→Bのある場合には1.59と5になる。

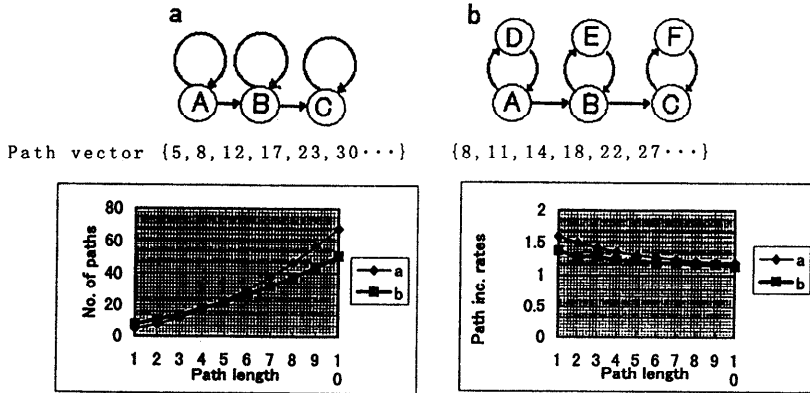


図3 共通のノードのないループを含むフローグラフとそのパス数 $\{P_L\}$ の例
 Fig. 3 Examples of flowgraphs which have loops without common nodes, and graphs of their path vector $\{P_L\}$.

4. パスの個数

各種のフローグラフについて P_L の評価を試みる。ここではループとは無限に反復可能なものとする。

4.1 ループと P_L の関係

定理2: フローグラフがループを含まない場合には、 N をノード数として $L \geq N$ のとき $P_L = 0$ となる。逆に $L \geq N$ となる長さ L のパスはループを含む。

証明: フローグラフのノードの個数を N とすると、長さ N 以上のパスは必ず同じノードを2度通ることになり、そこにループができる。したがって、ループがなければ $L \geq N$ となるパスはない。 QED

図1がループのないフローグラフの例である。この連結行列はスタートノードからエンドノードへと番号付けすると、図1の例に示すように左下半分がすべて0要素である三角行列にすることができる。三角行列は少なくとも N 回べき乗すると0行列になる。以降ではノードAに始まりBに到達する長さ L のパスの個数を $P_{AB}(L)$ などと表す。

4.2 共通のノードを持たないループがある場合

定理3: フローグラフがループを含みそのループが互いに共通のノードを持たない場合は、全体のノード数を N 、1つのパスでつながっているループの最大数を I 、 $L \geq N$ とすると、長さ L のパス数 P_L は L^{I-1} 次の多項式 ($I=1$ のときは振動または一定値) になる。

また、その各ループに含まれるノードの数をそれぞれ J_1, J_2, \dots, J_I 。その最小公倍数を J とすると、 J 個の連続する増加率の幾何平均は1に近づく。すなわち、

$$\{(P_{L+1}/P_L) \times (P_{L+2}/P_{L+1}) \times \dots (P_{L+J}/P_{L+J-1})\}^{1/J} \rightarrow 1 \quad (5)$$

証明: 図3のフローグラフを使い4段階に分けて証明する。

(1) まず図3aのように各ループに1つしかノードがない場合を考える。これは $I=3, J=1$ の例である。最も多くのループを含むパス、 $A \rightarrow B \rightarrow C$ という長さ2のパス $P_{AC}(2)$ がループを1つまわると長さ3のパスになる。ループは3個あるから長さ3のパスは、 $P_{AC}(3) = P_{AC}(2) \times 3$ である。パスの長さが K でループが I 個ある場合、 $P_{AC}(K+1) = P_{AC}(K) \times I$ となる。同様にして $P_{AC}(K+2) = P_{AC}(K) \times (I$ 個の中から2個を順序を考慮せずに選んだ場合の数 $= {}_I H_2)$ となる。したがって I 個のループを含む長さ $L = K + (L - K)$ のパス数は、 K を固定して L を大としたとき、

$$\begin{aligned} P_{AC}(L) &= P_{AC}(K) \times I {}^H_{L-K} \\ &= P_{AC}(K) \times (L - K + I - 1) \\ &\quad (L - K + I - 2) \dots (L - K + 1) / (I - 1)! \\ &= P_{AC}(K) \\ &\quad \times \{(L - K) \text{ の } I - 1 \text{ 次多項式}\} \\ &\quad \rightarrow L \text{ の } I - 1 \text{ 次式} \end{aligned}$$

となり、 $P_{AC}(K)$ は $L \geq N$ の場合、 L の $I - 1$ のべき乗のオーダーで増加する。つながっているループの数が I より少ないパスの個数は、 $I - 2$ 次以下の多項式になるから無視でき、結局 P_L は L の $I - 1$ のべき乗で増加する。

(2) 各ループがノードを2つ以上持つ場合、すなわち図3bの $P_{AC}(L)$ について考える。この図では $J=2$ であるが、ループA, B, Cのノード数を J_A, J_B, J_C 、その最小公倍数を J とし、パス $P_{AC}(L)$ を長さ J だけ伸ばすと、 I 個のループのどれでも選択で

きるから、 $P_{AC}(L+J) \geq P_{AC}(L) \times I$ となる。したがって(1)と同じ論法によって、 $P_{AC}(L)$ は $L \geq N$ で L が J の倍数のとき L/J の $I-1$ 次多項式より大きい。

図3aとbの $P_{AC}(L)$ を比較すると、図3aの $P_{AC}(L)$ のうち各ループを0または偶数回まわるパスの数と図3bの $P_{AC}(L)$ のパスの数とが等しい。しかし図3aのパスのうち各ループを奇数回まわるパスは図3bには対応するパスがなく、それだけ図3bのパスの方が数は少ないので、図3aの $P_{AC}(L) >$ 図3bの $P_{AC}(L)$ となる。したがって、一般に図3bの $P_{AC}(L)$ は、 L が J の倍数のとき、 L の2つの $I-1$ 次式にはさまれる。

(3) L が J の倍数でない場合は、図3bの両端のループA、Cにそれぞれ J_A, J_C 個のノードがあるとすると、パス $A \rightarrow B \rightarrow C$ を含むパスは $J_A \times J_C$ 組だけある。それ以外のパスは含むループの数が少ないので無視できる。図3bの例では $J_A = J_C = 2$ だから、ループを3個含むパスは、 $P_{AC}, P_{DC}, P_{AF}, P_{DF}$ の4組である。 P_{DF} を考えると、これは P_{AC} から $A \rightarrow D$ と $F \rightarrow C$ という2つのパスを除いたものと考えられるから、 $P_{DF}(L) = P_{AC}(L-2)$ という関係がある。同様に $P_{DC}(L) = P_{AF}(L) = P_{AC}(L-1)$ であり、図3bの P_L は、

L が偶数ならば、

$$\begin{aligned} P_L &= P_{AC}(L) + P_{DF}(L) \\ &= P_{AC}(L) + P_{AC}(L-2) \\ &\rightarrow L \text{ の } I-1 \text{ 次式} \end{aligned}$$

L が奇数ならば、

$$\begin{aligned} P_L &= P_{DC}(L) + P_{AF}(L) \\ &= 2 \times P_{AC}(L-1) \\ &\rightarrow L-1 \text{ の } I-1 \text{ 次式} \end{aligned}$$

となり、パス数はすべての L で L の $I-1$ 次式となる。

P_L はすべての $L > N$ についてAからCに行くパスを含むパスを含み、またAからCに行くパスを含むパスの数は $P_{AC}(L)$ の整数倍で表され、 $P_{AC}(L)$ は(2)により L の $J-1$ 次式で表されるので、結局 P_L はすべての $L \geq N$ で L の $J-1$ 次式となる。

(4) 式(5)は、 P_L がサイクル J で(L の $I-1$ 次式)となるので、それらを $\alpha_L L^{I-1}, \alpha_{L+1} L^{I-1}, \dots, \alpha_{L+J-1} L^{I-1}$ と表すと、 $\alpha_L = \alpha_{L+J}$ で、任意の L から $L+J-1$ について、式(5)は

$$\begin{aligned} &\{\alpha_{L+1}(L+1)^{I-1} \\ &\quad \times \dots \alpha_{L+J-1}(L+J-1)^{I-1} \\ &\quad \times \alpha_{L+J}(L+J)^{I-1}\}^{1/J} \\ &\quad / \{\alpha_L L^{I-1} \times \alpha_{L+1}(L+1)^{I-1} \times \dots \\ &\quad \times \alpha_{L+J}(L+J-1)^{I-1}\}^{1/J} \\ &= \{(L+J)^{I-1} / L^{I-1}\}^{1/J} \\ &= (1+J/L)^{(I-1)/J} \rightarrow 1 \end{aligned}$$

となる。

QED

P_L は連結行列をべき乗してその要素を合計すれば数値として得られる。 $L \geq N$ なるパスの P_L を式で求めるには、すべてのノードの組合せについてパス数を L の関数として求め合計すればよい。このような求め方は面倒で実際的ではないが、図3aのフローグラフについて実例を示す。

$$\begin{aligned} P_{AC}(L) &= {}_3 H_{L-2} = L \times (L-1)/2 \\ & \quad (2H_{L-2} \text{ の } -2 \text{ は、} A \rightarrow B \text{ と } B \rightarrow C \text{ のリンクを示す}) \end{aligned}$$

$$P_{AB}(L) = P_{BC}(L) = {}_2 H_{L-1} = L$$

$$P_{AA}(L) = P_{BB}(L) = P_{CC}(L) = 1$$

したがって、これらを合計して

$$\begin{aligned} P_L &= P_{AC}(L) + P_{AB}(L) + P_{BC}(L) + P_{AA}(L) \\ &\quad + P_{BB}(L) + P_{CC}(L) \\ &= L \times (L-1)/2 + L + L + 1 + 1 + 1 \\ &= 0.5L^2 + 1.5L + 3 \end{aligned}$$

となる。

4.3 共通ノードを持つループがある場合

ループが互いに共通のノードを持つ場合、 P_L には次のような性質がある。

定理4: フローグラフに、同じノードを共有しているループのグループがある場合、同じノードを共有するループの数を $I > 1$ 、 N をフローグラフのノードの総数とすれば、 $L \geq N$ なるときに、 P_L は I^L のオーダーで増加する。

また、 I 個のループに含まれるノードの数をそれぞれ J_1, J_2, \dots, J_I とし、その最小公倍数を J とすると、 $L \rightarrow$ 大のとき

$$\begin{aligned} &\{(P_{L+1} \times P_{L+2} \times \dots \times P_{L+J}) / (P_L \times P_{L+1} \\ &\quad \times \dots \times P_{L+J-1})\}^{1/J} \rightarrow C > 1 \quad (6) \end{aligned}$$

すなわち連続する J 個の P_L の増加率の幾何平均は、1より大なる数値に収束する。

証明: 図4で議論を進める。aの例では $I=2, J=1$ 、bでは $I=2, J=2$ である。定理3と同じく4段階に分けて証明する。

(1) 定理1によって、 $L \geq N$ なる L について、 $(J_1 + J_2 + \dots + J_I - I + 1) \times I^L \geq P_L$ である。

(2) 図4bのようにループが2つ以上のノードを

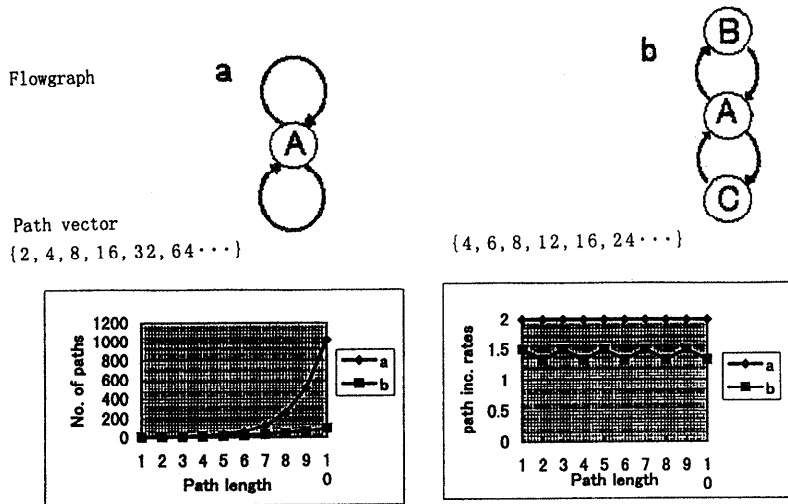


図4 共通のノードを持つループを含むフローグラフの例
 Fig. 4 Examples of flowgraphs which have loops with common nodes.

持つ場合、ループの持つノード数の最小公倍数を J とすれば、 $P_{AA}(J) \geq I$, $P_{AA}(2J) \geq I^2$, 同様にして $P_{AA}(N \times J) \geq I^N$ で、 L が J の倍数のときに $P_L \geq P_{AA}(L) \geq I^{L/J}$ となる。したがって、(1)とあわせて、 $L \geq N$ かつサイクル J のとき、 P_L は L の2つの指数関数に挟まれる。

(3) $L \geq N$ なる任意の L について P_L のパスはノード A を通り、したがって $P_{AA}(L)$ を含むパスを含む。 $P_{AA}(L)$ は L の指数関数で増加するので、 P_L は L の指数関数で増加する。

(4) 式(6)については、 P_L はサイクル J で指数関数で増加するので、それらを $\beta_L \times \alpha_L^L$, $\beta_{L+1} \times \alpha_{L+1}^{L+1}$, \dots , $\beta_{L+J-1} \times \alpha_{L+J-1}^{L+J-1}$ と表す。 $I > 1$ であるから $\alpha_L > 1$ である。また $\alpha_{L+J} = \alpha_L$, $\beta_{L+J} = \beta_L$ である。したがって任意の L から $L+J-1$ について式(6)→

$$\begin{aligned} & \{\beta_{L+1} \times \alpha_{L+1}^{L+1} \times \beta_{L+2} \\ & \times \alpha_{L+2}^{L+2} \times \dots \times \beta_L \times \alpha_L^{L+J}\}^{1/J} / \\ & \{\beta_L \times \alpha_L^L \times \beta_{L+1} \times \alpha_{L+1}^{L+1} \\ & \times \dots \times \beta_{L+J-1} \times \alpha_{L+J-1}^{L+J-1}\}^{1/J} \\ & \rightarrow \alpha_L > 1 \end{aligned}$$

となる。 QED

図4の a, b について P_{L+1}/P_L を計算すると、a では $\alpha = 2$ でループの数となる。b についてはサイクルが2で $\alpha_1 = 1.333$ と $\alpha_2 = 1.5$ という値に収束し、その積はちょうど2となり、サイクル2で2倍ずつ増加していることを示す。また図2の例ではノード C を共有している2つのループ $C \rightarrow D \rightarrow E$ と $B \rightarrow C \rightarrow F$

がそれぞれ3つのノードを持っているので最小公倍数 J は3となり、 P_L の増加率はサイクル3で振動している。

5. バスベクトルの応用例

バスベクトルの概念の有用性を示すために、前章で得た結果の応用例を示す。

5.1 構造の“単純な”プログラムの作り方

構造の単純なプログラムとは、バスベクトルのパスの数が少ないプログラムと考え、そういうプログラムを作成するためのガイドラインを示し、構造化プログラミング¹⁰⁾の考えと比較する。本ガイドラインと構造化プログラムの思想とは一致している点もあるが違う点もある。

構造化プログラミングの定義は人によってやや異なるが、ここでは次のように考える。

[シーケンス, IF_THEN_ELSE, WHILE_DO, REPEAT_UNTIL だけを使用し, GO_TO 文を使用しないプログラムである。特にループ内への飛び込みやループからの飛び出しは禁止する.]

一方、パス数の少ないプログラムの作成方針は次のようなものである。

(1) ループの削減

ループはない方がよい。したがってフローグラフ上で後戻りする GO_TO 文は悪い。

理由：定理2によりループがなければパスの数は有限となり、100%のパスカバレッジテストが可能である。構造化プログラミングと違い、フローの先の方に進む GO_TO 文はよい。

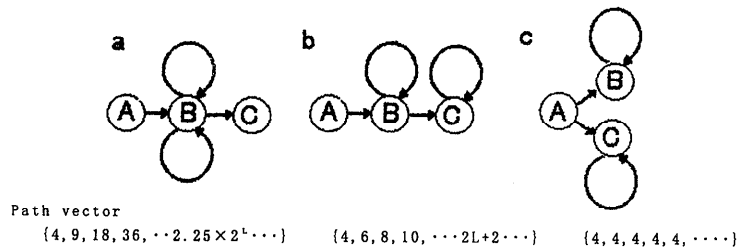


図5 フローグラフの単純さの比較
Fig. 5 The comparison of simplicity of flowgraphs.

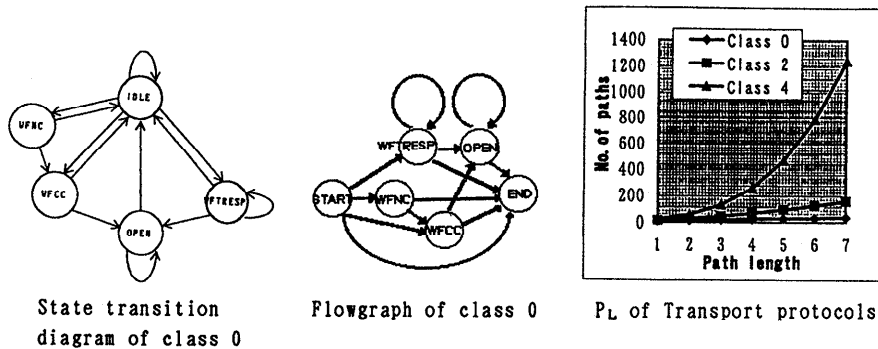


図6 トランスポートプロトコルのパス数 P_L .
Fig. 6 Path numbers of ISO transport protocols.

(2) プログラムのスタートからエンドまでのパスの長さの短縮化

長いパスはできるならば分断して短いパスの並列にする。特に、ループがある場合は、1つのパスに多くのループが並ばないようにする。

理由：プログラムは短い方がテストしやすいことは自明であろう。特にループがある場合、定理3により、1つのパスに並んだループの数が多いほどパス数が増加する。したがって、ループは直列より並列に並べる方がよい(図5b, cを参照)。

(3) ノードを共有するループの削減

共通のノードを持つループは作らないようにする。したがって、①ループの中のループ、②ループの中の判定はよくない。

理由：定理4により、2つのループに共通のノードがあるとパス数は L の指数関数で増加する。ループへの飛び込みや飛び出しも、一般に①や②と同様に悪い。しかしループの途中でプログラムの進行方向へ抜け出すのは、出口が2個になるのでパス数が2倍になるだけであり、指数関数で増加するのに比べれば悪くない。これが構造化プログラミングの考えと違う点である。

図5のa, b, cは、みなリンクが4個でループが2個であるが、aよりb, bよりcが望ましい。

5.2 トランスポートプロトコルの評価

トランスポートプロトコルには3つのクラスがあるが、それをパスベクトルで評価比較してみる。

状態遷移図のアイドルノードを2つに分割し、一方をスタートノード、他方をエンドノードとするとフローグラフになる。図6にトランスポートプロトコルのクラス0のフローグラフと、連結行列から得た P_L の値を示す。このフローグラフのループは共通のノードを持たないので、定理3により L が大のとき P_L は $L \geq N$ で L の $I-1$ 次多項式で表される。その多項式は連結行列から得た数値と一致するように係数を定めれば求められる。他のクラスも同様で、クラス0, 2, 4はそれぞれ $P_L = 10 + 4L$, $6 + 8L + 2L^2$, $(54 + 62L + 23L^2 + L^4)/6$ となる¹¹⁾。クラスが上がるほどパス数が増加することが、 P_L の式の次数が増えることで示される。このことはすでに発表済みであるが¹¹⁾、本論文によって一般化と完全な証明を紹介した。

一般に近似式とは独立変数の小さいところで近似がよく、大きくなると誤差が増えるが、定理3の場合には逆に $L \geq N$ で正確な値となり、 L が小さいところでは正しさは保証されない。トランスポートの場合には、 $L > 2$ で連結行列から得られる値と正しく一致し

ている。一般のフローグラフでは指数関数と多項式が混在するので、このように単純な式の形で求めることは難しい。

6. まとめと展望

フローグラフのパスの個数を長さ順に並べたパスベクトル $\{P_L\}$ という概念を提案し、これがプログラムの構造の複雑さを表す指標となりうるかを検討し、次の結果を得た。

(1) P_1 はプログラムの規模、 P_2 はブランチの数、 P_L の増加率は試験のしにくさを表すと考えられ、これによりプログラムの構造の性質を表すことができることを例によって示した。

(2) フローグラフはそれに含まれるループによって、 $L \rightarrow$ 大のときに、次の3つのタイプに分類できることを証明した。

- (i) ループのない場合には、 $P_L \rightarrow 0$
- (ii) ループを含みそのループが共通のノードを持たない場合には、パス数は L の多項式で表現され、その増加率の幾何平均 $\rightarrow 1$
- (iii) 2つ以上のループが同じノードを共有する場合には、パス数は L のべき乗で増加し、その増加率の幾何平均 $\rightarrow C > 1$

(3) パスベクトルによる評価の応用として、単純なプログラムを設計するためのガイドラインを示して構造化プログラミングの思想と比較した。またトランスポートプロトコルの3つのクラスのパスベクトルを比較した。

今後は、パスベクトルとパスカバレッジ試験とを結び付け、残存エラー数を推定する方法を研究する予定である。

参考文献

- 1) McCabe, T.J.: A Complexity Measure, *IEEE Trans. Softw. Eng.*, 2-4, pp.308-320 (1976).
- 2) Halstead, M.H.: *Elements of Software science*, elsevier, North Holland (1977).
- 3) Lipow, M.: Number of faults per line of code,

IEEE Trans. Softw. Eng., pp.437-439 (1982).

- 4) Li, H.L. and Chung, W.K.: An Empirical Study of Software Metrics, *IEEE Trans. Softw. Eng.*, Vol.SE-13, No.6 (1987).
- 5) Fenton, N.: Software Measurement: A Necessary Scientific Basis, *IEEE Trans. Softw. Eng.*, Vol.20, No.3, pp.199-206 (1994).
- 6) Kitchenham, B., et al.: Towards a framework for software measurement validation, *IEEE Trans. Softw. Eng.*, Vol.21, No.12, pp.929-943 (1995).
- 7) Beizer, B., 小野間, 山浦 (訳): ソフトウェアテスト技法, p.442, 日経出版 (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カバレッジ, 電子通信学会論文誌, Vol.J72-BI, No.4, pp.159-155 (1996).
- 9) 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, 情報処理学会ソフトウェア工学研究会報告, 96-SE-112, pp.9-16 (1996).
- 10) Dijkstra, E.W., et al., 野下ほか (訳): 構造化プログラミング, p.249, サイエンス社 (1975).
- 11) 若杉忠男: ISO で開発したトランスポートプロトコルの適合性試験スイートの質の評価, 情報処理学会論文誌, Vol.37, No.3, pp.451-459 (1996).

(平成 9 年 11 月 5 日受付)

(平成 10 年 11 月 9 日採録)



若杉 忠男 (正会員)

昭和 8 年生。昭和 36 年東京大学理学部数学科卒業。三菱電機 (株) にて一貫してソフトウェア開発に従事。平成元年退職後、技術コンサルタント、また湘南工科大情報工学科、文教短大、埼玉短大、東京高専の非常勤講師を兼任。この 10 年ほどはソフトの品質管理や試験方法等に興味を持ち、最近では「ソフトウェア試験の残存エラー数の推定」について、情報処理学会ソフトウェア工学研究会などで発表。昭和 63 年に技術士 (情報処理部門)。平成 10 年 3 月に「OSI 適合性試験の品質評価に関する研究」によって工学博士 (東京大学)。電子情報通信学会会員。