状態遷移図の同定問題の一解法

若杉忠男
若杉情報技術コンサルタントオフィス　電話0466(23)4832

状態遷移図をパスに分解し，長さＬのパスの個数をＰＬと記すと，これで状態遷移図の複雑さが表される．また各リンクを試験したときのフォールト発見率の下限をｒとするとフォールト残存率上限はＰＬ×（１－ｒ）^{L}／Ｐ１で表される．これから試験によってフォールトの個数を０にできる条件が分かり，発見フォールト数から残存フォールト数の推定ができる．またパスＰＬを試験項目でどれだけカバーするかで試験の程度が評価できる．

1　状態遷移図の同定問題とは

状態遷移図の同定問題とは，状態遷移図で表された構造と試験対象の構造とが一致するかどうかを確認することである[1]．すなわち同定問題ではすべてのリンクについて次の確認をする．

Ａ→Ｂというリンクを考えると
- スタートノードが確かにＡか．
- 入力Ｉ１に対して期待されるＯ１が出力されるか．
- エンドノードが確かにＢか．

入力Ｉ１に対して出力Ｏ１が出ることを確認するのは，一般に難しくはない．しかしエンドノードが確かにＢかという確認はブラックボックス試験では難しい．その確認には，状態Ｂに到達後に続けてＩ２を入力し期待されるＯ２が出力されＣに到達することを確認する．しかしＣの確認がやはり確実ではない．Ｂであるという確認には“Ｂと思われる状態”に到達後，引き続いて何個かの入力データを与え，その一連の出力がＢから遷移した状態から期待される出力と一致することを確認し，かつ“Ｂではない”状態から期待されるものとは一致しないことを確認する[2][3]．しかし“Ｂと思われるがＢではない”状態はエラーを含む状態遷移図には無数にありうる．したがって同定問題では，そのＢに似ているがＢではないというような状態の数はあまり多くないという仮定のもとに同定試験をする．

本論文ではそのような仮定をはずした同定法を提案する．

2　状態遷移図の複雑度

状態遷移図をそれを構成するパスの個数で表すことをパス分解と呼ぶ．筆者は，そのパスの個数で状態遷移図の複雑度を表すことを提案した[4][5]．

状態遷移図の各ノードを出発点とした長さＬのパスの数をＰＬと表わすと，ＰＬは連結行列を使って求められる．連結行列とは状態遷移図のノードの数をＳ個とするとＳ×Ｓ行列で，その各ｉｊ要素はノードｉからノードｊへとＨ本のリンクがつながっているときにＨ，リンクのない場合は０とするものである．トランスポートプロトコルとＦＴＡＭのイニシエータ側については，ＰＬが次のような簡単な関数で近似できる[4]．この式の［］は括弧内の数値を整数に切り捨てることを意味する．

On a Conformance test method of state transition diagram.
Tadao Wakasugi
Wakasugi Information technology consultant office

トランスポート

クラス０　10+4L

クラス２　$6+8L+2L^2$

クラス４　$(54+62L+23L^2+10L^3+L^4)/6$

FTAM　$23.21\times1.63^{[(L-1)/2]}\times2.22^{[L/2]}$

ここでは，資料[4]とは少し違い，状態遷移図をアイドルのところで切り離して二つに分け，一方を状態遷移図のスタートノード，他方を状態遷移図の終了ノードとする連結行列を考える．図１～４にトランスポートクラス０，２，４およびＦＴＡＭ（ファイル・トランスファ・アクセス・マネジメント）の状態遷移図と，それらに対応する連結行列を示す．

こうして作った行列をＭとし，このマトリックスの要素をm_{ij}と表すとＭ×Ｍの各要素は，

$$\sum_{k=1}^{n} m_{ik}\times m_{kj} \qquad (1)$$

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．したがってこの行列の要素の合計は長さ２のパスの総計になる．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．

パスの個数が複雑度を表すとすれば，複雑度を減らすには連結行列をべき乗しても要素の合計が増えないようにすればよい．そのための条件はいろいろあるだろうが，簡単には次のようなことが考えられる．

（１）連結行列の０要素が多いこと．

（２）連結行列をうまく並べ変えると，左下（あるいは右上）半分が０要素ばかりになる三角行列にできること．三角行列はべき乗するとやはり三角行列となるので，要素の増加速度小さい．

複雑度を表す指標の一つにマッケイブのサイクロマチック数があるが，これは

サイクロマチック数
＝リンク数－ノード数＋２　(2)

という式で表される[4]．ノード数＝連結行列の行数であるから，サイクロマチック数が小さいということは，連結行列の各行の０要素の数が多いことを示すので，これは上記（１）の条件を示す指標であると考えられる．たとえば，もっとも単純な状態遷移図はＡ→Ｂ→Ｃ→・・というノードが一列に並んだものであると考えると，そのサイクロマチック数は１であり，連結行列では各行に一つの１という要素があるだけであり，これ以上少なくはできない．

また構造化プログラミングでは，ＧＯＴＯレス，ＩＦ　ＴＨＥＮ　ＥＬＳＥの使用などを推奨しているが，これはプログラムフローの後戻りを許さないという思想であると考える．連結行列の各行は遷移先を表すものであるから，行と列を適当に並べ変えるて左下三角要素を０にできるということは，後戻りがないように並べることができるということである．したがって上記（２）の条件と構造化プログラミングの目的は一致する．

したがって，パスの個数の増加によって複雑度を定義するということは，従来からの複雑度の考え方と矛盾しない．

図１～３に示すトランスポートプロトコルの連結行列は，資料[5]などに記載したものを，左下半分の三角要素がすべて０になるように並べ変えたものである．三角行列になるということがトランスポートプロトコルの特徴の一つであり，構造がシンプルであることを示している．一方，ＦＴＡＭは状態遷移図を見て分かるように，大きなループがあるので下三角行列を０とすることはできない．

Idle	0	1	1	1	0	1
WFNC	0	0	1	0	0	1
WFCC	0	0	0	0	1	1
WGTRESP	0	0	0	1	1	1
Open	0	0	0	0	1	1
Idle	0	0	0	0	0	0

サイクロマチック数＝13-6+2＝9

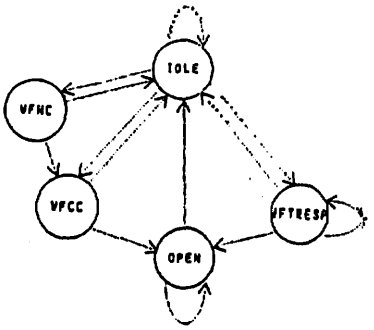


図1　トランスポートクラス0

Idle	0	1	1	1	0	0	0	1
WFNC	0	0	1	0	0	0	0	1
WFCC	0	0	0	0	1	1	0	1
WFTRESP	0	0	0	0	1	1	0	1
Open	0	0	0	0	0	1	1	1
WDCL	0	0	0	0	0	0	1	1
Closing	0	0	0	0	0	0	1	1
Idle	0	0	0	0	0	0	0	0

サイクロマチック数＝19-8+2＝13

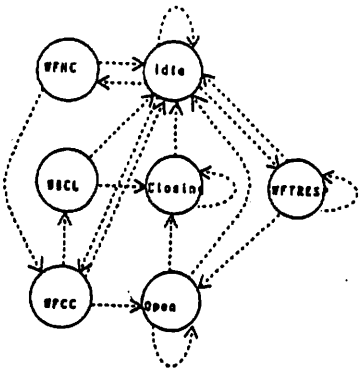


図2　トランスポートクラス2

Idle	0	1	1	0	0	0	0	1
WFT RESP	0	0	1	0	1	0	0	1
WFCC	0	0	0	1	1	0	0	1
WDCL	0	0	0	1	0	1	1	0
AK WAIT	0	0	0	0	1	1	1	0
Open	0	0	0	0	0	1	1	0
Closing	0	0	0	0	0	0	1	0
DCF WAIT	0	0	0	0	0	0	1	0
Idle	0	0	0	0	0	0	0	0

サイクロマチック数＝25-9+2＝18

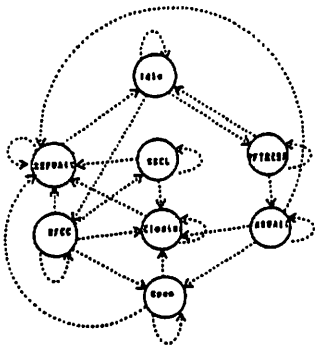


図3　トランスポートクラス4

Initialized	0	1	1	0	0	0	0	0	0	0	0	0
Create Pending	0	0	0	0	0	1	0	0	0	0	0	1
Select Pending	0	0	0	0	0	1	0	0	0	0	0	1
Deselect Pending	0	0	0	0	0	0	0	0	0	0	0	1
Delete Pending	0	0	0	0	0	0	1	0	0	0	0	1
Selected	0	0	0	1	0	1	1	0	0	0	0	0
Read-Attribute P.	0	0	0	0	1	0	0	0	0	0	0	0
Change-Attribute P.	0	0	0	0	0	1	0	0	0	0	0	0
Open Pending	0	0	0	0	0	1	0	0	0	1	0	0
Close Pending	0	0	0	0	0	1	0	0	0	0	0	0
Data Transfer Idle	0	0	0	0	0	0	0	0	1	0	1	0
Locate Pending	0	0	0	0	0	0	0	0	0	1	0	0
Erase Pending	0	0	0	0	0	0	0	0	0	0	0	0
Initialized	0	0	0	0	0	0	0	0	0	0	0	0

サイクロマチック数＝23-14+2＝11

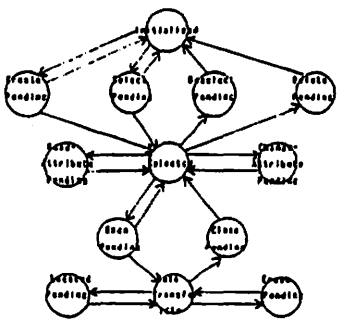


図4　ＦＴＡＭイニシエータ側
状態遷移図とそれに対応する連結行列

3　フォールトモデル

ここで状態遷移図のフォールトモデルを考える．フォールトモデルとは，フォールトの状態を定義したモデルである．フォールトモデルで表現されないフォールトは試験の対象にはならない．たとえば，状態遷移図がフォールトモデルならば，一般にはシステム効率などは試験対象ではない．

ここでは，フォールトとエラーという言葉を明確に区別する．システムの作成の間違い部分をフォールトと呼び，その結果生じた間違った出力／現象をエラーと呼ぶ．フォールトは原因でエラーはその結果である．状態遷移図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，下流で検出された病原菌がエラーである．

まず図5のような状態遷移図で考える．次のような前提条件を定める．

（1）試験は試験対象のパスの短い方から実施し，エラーが見つかったらその原因であるフォールトを除去し，フォールトを除いてから次のパスの試験を行う．これは資料[6]で述べたマルチトランジションカバレジ試験の考え方である．

（2）リンクにフォールトがあった場合に，その結果であるエラーはそのリンクを先頭にもったパスのどれかに現れる．どのパスからもエラーが発見されないフォールトはないと同じである．

（3）リンクA→Bのフォールトの下流にパスの枝別れがある場合，すなわち図5に示すように，B→C，B→Eというパスがある場合には，下流のパスの全てに等確率にエラーが起きる．

（4）エラーを見つけてフォールトを修正すると，そのエラー発見場所の下流のエラーは消えるが，下流以外のエラーの確率は変わらない．すなわち，A→Bのフォールトを B→Cで発見して修正するとC→D以降のパスではエラーは消えるが，B→Eでのエラーの起きる可能性は不変とする．

4　フォールト残存率

ここで，フォールトの残存率について考察する．

定理1

上記の前提条件で，リンク当り平均フォールト発見率をｒとする．０＜ｒ＜１であり，長さＬまでのパスのカバレジ試験を実施した場合のリンク当りの平均フォールト残存率ＺＬは次のようになる．

$$ZL = PL \times (1-r)^{L} / P1 \qquad (3)$$

したがって，Ｌ→∞のときにＺＬ→０が言えれば，試験によってフォールトの残存数を０に近付けることができる．

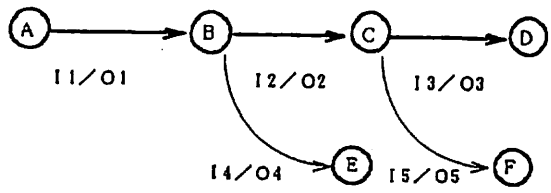
証明：

リンクを試験した場合のフォールト発見率を考える．その発見率にはリンクによって異なるだろうが，リンクの数は有限なので平均値が存在する．これを“リンク当りの平均フォールト発見率”と呼びｒと記述する．以降では平均という言葉を省くことがある．ここで０＜ｒ＜１である．

A→B→Cという長さ２のパスの試験のフォールト発見率は，A→Bにあるフォールトを見逃した上でB→Cで発見する確率で（１－ｒ）×ｒとなる．同様にして長さＬのパスでの発見率は（１－ｒ）$^{L-1}$×ｒとなる．

図5によって，長さ２までのパスをすべて試験した場合のフォールト残存率は，Bで見逃してかつCまたはEで見逃す確率で$(1-r)\times\{(1-r)+(1-r)\}=2(1-r)^2$となる．

一般にＮ本の枝別れをもつ一つのリンクについてパスの長さ２の試験がすんだ後でのフォールト残存率は，Ｎ×（１－ｒ）^{2}で



ﾊﾟｽの長さ	1	2	L
ﾊﾟｽ1本当り平均ﾌｫｰﾙﾄ発見率	r	$(1-r)\times r$	$(1-r)^{L-1}\times r$
ﾊﾟｽ本数	P1	P2	PL
ﾘﾝｸ当りﾌｫｰﾙﾄ平均残存率	r	$P2\times(1-r)^2/P1$	$PL\times(1-r)^L/P1$

図5　状態遷移図と試験の例

ある．これをすべてのリンクについて合計すると，$\Sigma N\times(1-r)^2=P2\times(1-r)$となる．ここでP2は，2節で考察した長さ2のパスの個数である．したがってこれをリンク当りで平均するとP2×$(1-r)^2$／P1となる．長さLまで試験すると，リンク当りの平均フォールト残存率ZLは，ZL＝PL×$(1-r)^L$／P1となる．

証明終了．

定理2

定理1と同じ条件のもとで，PLがLの多項式で表される場合には，パスの短い方からカバーするように試験を続けることにより，試験対象のリンク当りの平均フォールト残存率を0に近付けることができる．

証明：

リンク当りフォールト残存率ZLにおいて，P1は省略し，PLをLのK次多項式としてL^Kと表し，Lを実数とみなしてLで微分することをd/dLで表すと，

$d\{L^K\times(1-r)^L\}/dL$

$=KL^{K-1}\times(1-r)^L+L^K\times(1-r)^L\times\mathrm{Log}(1-r)$

$=L^K\times(1-r)^L\{K/L+\mathrm{Log}(1-r)\}$

$0<r<1$だからLog(1-r)＜0である．したがってLを充分大きくすれば{}内の第一項は0に近付き第二項はマイナスになり，結局$L^K\times(1-r)^L$は単調に減少する．$L^K\times(1-r)^L>0$であるから，あるC≧0に収束する．そのCを＞0と仮定して上式の最後の部分に代入すると，$C\times\{K/L+\mathrm{Log}(1-r)\}$となる．L→∞のときこの第一項は0に近付き，第二項はマイナスであるから，$L^K\times(1-r)^L$はさらに減少を続けることになる．これはC＞0に収束するという仮定に反する．したがってC＝0である．すなわち残存率は0に収束する．Kは任意の正整数であるから定理は証明された．

証明終了．

ZL＝PL×$(1-r)^L$／P1において，$(1-r)^L$の減少速度よりもPLの増加速度が大きければ，試験をしてもフォールトが減らないことになる．パスがべき乗で増加する場合がこれに相当する．状態爆発（State Explosion）という言葉があるが，そのようなことを表すものと思われる．

トランスポートプロトコルの場合には，PLはLの多項式で表される[4]．したがって，次のことが言える．

系

前定理の仮定のもとで，トランスポート

プロトコルは，試験によってフォールトを0に近付けることができる．

証明省略．

なお[4]で示したＦＴＡＭ（イニシエータ側）のプロトコルの残存フォールト率については，２節で示したように，

$PL = 23.21 \times 1.63^{[(L-1)/2]} \times 2.22^{[L/2]}$

となるので，これに$(1-r)^L$を掛けると，

定数$\times (1.63 \times 2.22)^{L/2} \times (1-r)^L$となる．これから，$r > 1-1/(1.63 \times 2.22)^{1/2} = 0.474\cdots$ならば，$L \to \infty$のとき0に収束することがいえる．

5　疑似カバレジ

試験スイートの品質を評価する指標として，資料[7]で疑似カバレジという考えを提案した．すなわち，状態遷移図をパス分解して得たベクトル｛P1，P2・・PL・・｝の各パスを試験スイートでカバーし，長さＬのパスをCL%カバーしたことをベクトルで表現し，カバレジベクトル｛C1，C2，・・CL・・｝と表す．これによって試験スイートの品質を評価する．これを一つの数値で表現するために

$$Y1 = 100 \times (C1/2 + C2/4 + C3/8 + \cdots) \quad (4)$$

という式を提案し，疑似カバレジと名付けた．ここに100は全部カバーすると100になるようにするための係数である．また1/2，1/4，・・は和を計算して１にするための重みであるが，ここで先のフォールトモデルとの結び付けて説明する．

長さ１のパスの試験によるリンク当りフォールトの発見率はｒである．パスの長さ２の発見率は長さ１のパスの試験で見逃したという条件のもとでの発見率で（１－ｒ）×ｒである．同様に長さＬのパスの試験による発見率は（１－ｒ）$^{L-1}$×ｒとなる．

したがって，カバレジベクトル｛CL｝の試験によるフォールト発見率は

$$\sum_{L=1}^{\infty} CL \times (1-r)^{L-1} \times r \quad (5)$$

となる．ｒの値は不明なので１と０の中間をとって，ｒ＝０．５を代入すると，

$$\text{試験効果} = \sum_{L=1}^{\infty} CL \times 0.5^L \quad (6)$$

となり，疑似カバレジの式となる．

この評価では，パスの長い方が試験効果は小さい．常識に反する評価のようであるが，これは一つの試験項目をパスの長さで分解して別々に評価しているからである．たとえば，一つのループを繰り返す試験項目を考える．１回目と２回目のループは必須であろう．３回目のループも実施した方がよいだろう．４回目のループは丁寧な試験だと感じるであろう．しかし１０回目のループとなるとしつこいという気がする．その感じを表現するために，｛CL｝の各項に，0.5^Lという重みをかけたと解釈する．またパスの長さが長くなるほど試験の価値が減るので，効率を考えると試験を適当な長さで打ち切ることが望ましい．これは自然な考えであり，0.5^Lという重みは試験の効果に対する実感を表わしていると考えられる．

なお，ＩＳＯで開発したトランスポート試験スイートの疑似カバレジはクラスによらずほぼ一定で６２点前後であった．ここで，疑似カバレジの性質を定理にまとめておく．

定理３

リンク当り平均フォールト発見率をｒ＝０．５とすると，状態遷移図がなんであろうと，疑似カバレジＹ1には次の性質がある．ただしCLは１以上の値になったときは切り捨てて１とする．

（1）Lが大になるほどカバレジの価値が減る．すなわちカバレジCLとCL+1の値が同じであっても，CL+1の価値はCLの半分になる．

（2）長さLまでのすべてのパスをカバーし，L＋1以上長さのパスはカバーしない場合のY1は次式で表される．

$$Y1(L) = (1-0.5^{L}) \times 100 \quad (7)$$

（3）状態遷移図のすべてのパスを試験項目が100%カバーしたとき，Y1は100になる．

証明省略

5　残存フォールト数の推定

この試験方法の特色は，残存フォールトの数を推定できることである．

パスの長さLまでのﾌｫｰﾙﾄ発見率
＝1－パス長さLでのﾌｫｰﾙﾄ残存率
＝1－ZL

であるから，フォールトの総数をF*とすると次式が成り立つ．

$$F^{*} \times (1-ZL) = F1 + F2 + \cdots + FL \quad (8)$$

特にL＝1，2のときは

$$F1 = F^{*} \times r \quad (9)$$

$$F1 + F2 = F^{*} \times (1-Z2)$$

$$= F^{*} \times \{1 - P2 \times (1-r)^{2} / P1\} \quad (10)$$

となり，未知数はrとF*の二つだから，(9)(10)式を連立させればF*を求められる．

試験の実施順序については，マルチトランジションカバレジ法では各ノードを出発点としてパスの長さの短い方からカバーしてゆくように試験するが，実際にはそのようにする必要はない．プロトコルのスタートを出発点として，エラーが見つかった場合には，そのフォールトを修正してから先に進む．そしてフォールトがエラーを発見したと同じトランジション内にあれば，長さ1のパスのフォールトとし，一つ前のトランジションにあれば長さ2のパスのフォールト，L－1個前ならば長さLのパスのフォールトと分類し，それらの個数をそれぞれF1，F2，FLとする．

これによって長さ1と2のパスをすべてカバーし終わったときに先の式(9)(10)を使ってF*を求めることができる．

詳細は別途発表する．

6　疑似カバレジとフォールト残存率

本方法で使用した品質評価の指標は二つある．一つはフォールト残存率ZLであり，もう一つは疑似カバレジY1である．前者は試験開始時から考えてどのくらいフォールトが残っているかという推定値であり，後者は試験対象のパスの長さ別のカバレジの程度である．疑似カバレジY1は，C1，C2などの多変数関数であるが，マルチトランジションカバレジ試験のもとではパスの短い方から試験してゆくので1変数関数となる．すなわち，定理3の(7)式が成立する．これをYLとおく．すなわち

$$Y1(L) = YL = (1-0.5^{L}) \times 100 \quad (11)$$

$$100 - YL = 0.5^{L} \times 100$$

ここで，(3)式にr＝0.5を代入し

$$YL + P1 \times ZL / PL \times 100 = 100 \quad (12)$$

が得られる．したがって長さの短いパスから試験してゆくという前提で，r＝0.5でY

LとZLの間には(12)式の関係が成り立つ．

A→B→C→というような枝別れのない単純な状態遷移図の場合，すなわち，P1＝P2＝・・PL＝1でパスの長さによる影響がない場合には，

$$YL + ZL \times 100 = 100 \qquad (13)$$

という簡単なものになり，YLとZLは一方が増加すれば，他方が減少するという関係にあることが分かる．

図6にトランスポートプロトコルのYLと100×ZLを示す．YLはクラスによらず共通なので一本しかないが，ZLはクラスによって異なるのでクラス0／2／4の3本の曲線となる．この図で見ると，クラス4の場合一時的にフォールト残存率が増加するという現象が起こる．

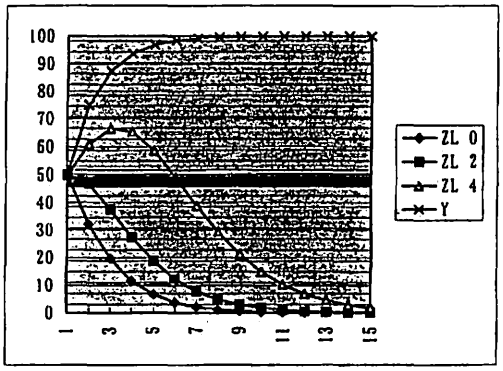


図6　ﾄﾗﾝｽﾎﾟｰﾄのYLとZL×100

7　まとめ

パス分解という方法を使い，

（1）状態遷移図の複雑度について，パス数とサイクロマチック数，構造化プログラミングとの関係を説明した．

（2）パスカバレジ試験の意味付けをし，試験でフォールトを0にできる条件を示した．また状態爆発という現象を説明できた．

（3）上記方法で，残存フォールト数の推定ができることを示した．

（4）試験の品質について，疑似カバレジとフォールト残存率の関係を示した．

今後は本理論を一般のソフトウェア／システムの試験に拡張し，ソフトウェア試験の理論の体系化と残存フォールト数推定法の実用化を目指す予定である．

参考文献

[1] Gerard J. Holzmann: "Design and Validation of Computer Protocols", Prentice Hall, Inc., 1991. 水野忠則他訳：“コンピュータプロトコルの設計法”，カットシステム（1994-11）．

[2] S.Fujiwara and Others:"Test selection Based on Finite States Models", IEEE Trans. of Software Eng. Vol.17, No.6, pp.591-603(June 1991).

[3] D.P.Siduh and Others:"Formal Methods for Protocol Testing:.Detailed Study", IEEE Trans. of Software Eng. Vol.15,No.4, pp.413-426(April 1989).

[4] 若杉忠男：“有限状態マシン(FSM)で表されるシステムの複雑度の評価について”，情報処理学会マルチメディア通信と分散処理研究会，(1995-9)．

[5] 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(1996-3)．

[6] 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”，日本電子情報通信学会論文誌，VOL. J72-BI No.4（1996-4）．

[7] 若杉忠男：“有限状態マシンで表されるシステムの試験スイートの必要度充足率について”，情報処理学会マルチメディア通信と分散処理研究会，pp.7-12，(1995-9)．