

システム故障診断に関する研究(1)内部テスト端子数の下限および最少内部テスト端子の決定アルゴリズム

メタデータ	言語: Japanese 出版者: 公開日: 2011-12-14 キーワード (Ja): キーワード (En): 作成者: 松本, 忠, 飛田, 義孝, MATSUMOTO, Tadashi, TOBITA, Yoshitaka メールアドレス: 所属:
URL	http://hdl.handle.net/10098/4652

福井大学
工学部 研究報告
第22巻 第1号
昭和49年3月

システム故障診断に関する研究 (1)

内部テスト端子数の下限および最少内部テスト端子の決定アルゴリズム

松 本 忠* ・ 飛 田 義 孝*

STUDIES ON THE SYSTEM DIAGNOSIS

Part 1 The Greatest Lower Bounds on the Number of the Internal Test-Terminals and the Algorithm of Determination for the Minimum Internal Test-Terminals in System Diagnosis.

Tadashi MATSUMOTO, Yoshitaka TOBITA

(Received Oct. 11, 1973)

In this paper, based on the graph theoretical approach, some considerations on the system diagnosis with the internal test-terminals (edges or vertexes) are given. The system under considerations is modeled fundamentally as the a-cyclic SEC graph (a-cyclic single-entry single-exit connected graph) with a single fault (that is, a single malfunction).

Under these assumptions, the following important articles are considered for the three types (U, L and UL methods) of system diagnosis.

(1) In order to be 1-distinguishable any a-cyclic SEC graph by the internal test-terminals, the alternative necessary and sufficient conditions for 1-distinguishability used the new concept of the dummy edges are given for each method of system diagnosis. These conditions make easy and simple determine the minimum internal test-terminals.

(2) The unified necessary and sufficient conditions for the absolute internal test-terminals—the ones to be selected without the regards of system structures for all possible a-cyclic SEC graphs—are shown in chapter 2 for each method of system diagnosis.

(3) Furthermore, in order to investigate the effectiveness of these system diagnosis, the greatest lower bounds and the least upper bounds on the number of internal test-terminals are derived for each system diagnosis. The necessary and sufficient conditions for the fault table of 1-distinguishable graph and the important properties of the vertex-range defined later are also shown.

(4) Based on the above results (1), (2) and (3), the effective algorithm of determination for the minimum internal test-terminals is shown in chapter 4 and the FORTRAN coded program is also given in appendix 1.

(5) The mutual relationship between the system diagnosis with internal test-edges and the one with internal test-vertexes are discussed in detail in chapter 5.

*電気工学科

1. まえがき

マイクロ・エレクトロニクス技術における大規模集積化 (large scale integration) の利用がますます普及していること、ならびに非常に複雑な電子計算機システムの開発、利用が進展していることなどは、システム故障診断ならびに保守のより有効な手法を開発する必要性をさらに増大させている。システムの故障診断には、システムの入出力関係のみによる方法* とさらにシステム内に入出力機能をもった端子 (すなわち、内部テスト端子) を導入して診断する方法 [1], [8], [9] とに大別される。

本論文は、後者の立場の故障診断法にグラフ理論的考察を適用することにより、故障診断の有効さを最大にすることとシステム構造の関係を明らかにするための第一段階の検討を行なったものである。[10]～[13], [14]～[16]

機能的に結合されたユニットから構成されているシステムのユニットを節点に、各ユニット間の情報または信号の流れを有向枝に対応させることによってシステムを a-cyclic SEC グラフ (a-cyclic single-entry single-exit connected graph の略称) であらわすことができる。ここでは対象とするシステムに、さらに、つぎの3つの基本的な機能を仮定する。すなわち、各ユニットに対して、

(i) 規定入力が入っているとき、規定出力を出すならば、そのユニットは正常状態にある。

(ii) 規定入力が入っているのに、規定出力を出さないならば、そのユニットは故障状態にある。

(iii) ユニットが正常状態でも規定入力が入らなければ、規定出力を出さない。

このようにモデル化された一部の a-cyclic SEC グラフを対象にして Mayeda 氏等は単一の故障節点を確定的に抽出するために内部テスト端子の備えるべき必要十分条件を与えた[1]。その後、中野および中西両氏は一般的な a-cyclic SEC グラフに対する必要十分条件を明らかにした[9]。しかし、最少内部テスト端子をデジタル計算機を用いて実際に求めることを考えると、上述のそれら必要十分条件は必ずしも適当でない。そこで本論文では、dummy edge なる新し

*: この一部には従来のデジタルシステム故障診断における最小入力系列の設計問題があり[4]、その他にはアナログシステムをも含めたシステム内に積極的に信号の流れをブロックする機能をもったユニットを設けて行なう場合がある。[2], [3]

い概念を導入してデジタルシミュレーションに適した必要十分条件の別表現を与えている。

さらに、最少内部テスト端子を求めるとき、グラフ構造の如何に関係なく必ず選定されるべき端子を前もって明らかにしておくことは、その探索手順および探索時間の両面からみても大切なことである。本論文では、可能なすべてのテスト法に対して、この絶対内部テスト端子のもつ諸性質を明らかにしている。

他方、高信頼度をもつシステムを開発するためにも、また、本論文で述べる故障診断法を最大限有効に活かしてゆくためにも、システム設計の段階にはじめから故障診断および保守を考慮に入れておくことが重要である。このためには、まず、節点数 n が与えられたもとでの可能なすべての a-cyclic SEC グラフの内部テスト端子数の下限およびそのグラフの構造や生成法などを明らかにする必要がある。従来、下限に関しては一部のテスト法に対する推測式が与えられていたにすぎなかったが、本論文の第3章ではこれを厳密に論じて下限式を与えている。さらに、付録2では、特別なグラフ構造に対する下限式をのべている。

さらに、第4章では、第2章および第3章において得られた結果をもとにした最少内部テスト端子の決定アルゴリズムを与えている。(また、FORTRAN IV によるプログラム例を付録1に与えてある。)

第4章までは内部テスト端子を枝に設けた場合 (枝レベル) の詳細な検討であるが、内部テスト端子を節点に設ける立場 (節点レベル) も考えられる。第5章では、両者における諸概念および手法の相互関係を明示して、枝レベルでの諸結果および手法がある種の条件のもとで節点レベルに適應できることをのべている。

2. システムの故障診断に関する基本定理^[10]

2.1 諸定義 (その1)

ここでは以下の議論に必要な諸定義をのべる。有向グラフは、節点の集合 $V(\neq \emptyset)$ と有向枝 (以下では枝と略する) の集合 $E(\neq \emptyset)$ および定義域が E であり値域が V である関数 f, s によって記述される。すなわち、 $v \in V, e \in E$ とすれば、 $f(e)$ は枝 e の始まる点 (始点) を示し、 $s(e)$ は枝 e の終わる点 (終点) を示す。

また、 $f^{-1}(v)$ は節点 v から出る枝の集合を示し、 $s^{-1}(v)$ は節点 v に入る枝の集合を示す。 $f^{-1}(v)$ の元の数を節点 v の out-degree と呼び $od(v)$ で示し、 $s^{-1}(v)$

*: ただし、 $ff^{-1}(v) = v$, $f^{-1}f(e) \supseteq e$ ($ss^{-1}(v) = v$, $s^{-1}s(e) \supseteq e$) である。

の元の数をもつ節点 v の in-degree と呼び $id(v)$ で示す。 $u, v \in V$ に対して、節点 u から節点 v に到達可能であるということは u から v に至る有向道 (directed path) が存在することであり、 $u \geq v$ と示す。(ただし、 $u \geq u$) $u \neq v$ とは u から v に到達可能でないことを示す。 V の部分集合 v においてその任意の 2 つの節点が互いに到達可能であるとき、節点部分集合 v を strong component という。 $s^{-1}(v) = \phi$ となる節点 v を入力節点 (entry vertex)、 $f^{-1}(v) = \phi$ となる節点 v を出力節点 (exit vertex) という。

1 個の入力節点と 1 個の出力節点をもち、cycle を含まない連結グラフを a-cyclic SEC グラフと呼ぶ。入力節点および出力節点をそれぞれ 2 個以上有し、cycle を含むグラフをつぎのようにすることによって a-cyclic SEC グラフに変換することができる。すなわち、strong component を 1 つの節点に置き換えたのち、新たな入力節点を導入して、その節点から以前の入力節点に向う枝を挿入する。さらに新たな出力節点を導入して以前出力節点から新たな出力節点に向う枝を挿入する。

以下では、a-cyclic SEC グラフ G を対象にして考える。そのとき、 G の入力節点を v_1 、出力節点を v_n とする。

定義 1 グラフ G の枝 $e \in E$ による vertex range $V(v_1, f(e))$ および $V(s(e), v_n)$ をそれぞれつぎのように定める、

$$V(v_1, f(e)) \triangleq \{v' \mid v' \in V, v_1 \geq v', v' \geq f(e)\},$$

$$V(s(e), v_n) \triangleq \{v' \mid v' \in V, s(e) \geq v', v' \geq v_n\}.$$

また、 $E\{V_1 \times V_2\}$ を一般に節点集合 V_1 および V_2 間を結ぶ枝集合、すなわち、カットセットとする。(定義終)

すると、 $V_1 \triangleq V(v_1, f(e))$ 、 $V_2 \triangleq V(s(e), v_n)$ とするときの $E\{V_1 \times V_2\}$ は有向カットセットになるが、以後これを枝 e による上方有向カットセットと呼び、 $U\text{-}DC\langle e \rangle$ と記す。ただし、 $V = V_1 \cup V_2$ である。

$V_1 \triangleq \overline{V(s(e), v_n)}$ 、 $V_2 \triangleq V(s(e), v_n)$ とするときの $E\{V_1 \times V_2\}$ は有向カットセットになるが、以後これを枝 e による下方有向カットセットと呼び、 $L\text{-}DC\langle e \rangle$ と記す。ただし、このときも、 $V = V_1 \cup V_2$ である。

2.2 システムの故障診断法

まえがきにのべたシステムモデルに、さらに、つぎの 2 つの事項を仮定する。

(1) 1 つのシステム内に故障は 1 つのユニット (節点) にしか生じない。そして、それは保守されない限り恒

久的である。(2)* 任意の枝 $e (\in E)$ に対してつぎのいずれかが行なえる。(i) $f(e)$ の出力を枝 e で代表して観測できる。(ii) $s(e)$ への入力を枝 e から代表して行なえる。(iii) (i) および (ii) をともに行なえる。

2.2.1 故障の検出

上述の仮定から、入力節点 v_1 に規定入力を入れたとき、出力節点 v_n に規定出力が現われれば、そのシステムは正常状態にあり、規定出力が現われなければ故障状態にあると定められる。以後に扱うシステムは故障状態にあるものとする。

2.2.2 故障の location 法

上述の仮定を考慮に入れると故障節点の location 法としてつぎの 3 つが考えられる。

U 法 入力節点 v_1 からテスト信号を入れて任意の枝 e によって節点 (すなわち、ユニット) $f(e)$ の出力を観測した結果によって、それが規定出力ならば故障節点は $\overline{V(v_1, f(e))}$ 内に存在し、規定出力でないならば故障節点は $V(v_1, f(e))$ 内に存在する、と判断できる。

L 法 任意の枝 $e (\in E)$ から節点 $s(e)$ にテスト信号を入れて出力節点 v_n の出力を観測した結果によって、それが規定出力ならば故障節点は $\overline{V(s(e), v_n)}$ 内に存在し、規定出力でないならば故障節点は $V(s(e), v_n)$ 内に存在する、と判断できる。

UL 法 任意の枝 e に対してつぎの (i)、(ii) および (iii) のように判断できる、(i) 入力節点 v_1 にテスト信号を入れて、枝 e によって節点 $f(e)$ の出力を観測した結果、その出力が規定出力でないならば、故障節点は $V(v_1, f(e))$ 内に存在する、(ii) 枝 e から節点 $s(e)$ にテスト信号を入れて出力節点 v_n の出力を観測した結果、それが規定出力でないならば故障節点は $V(s(e), v_n)$ 内に存在する、(iii) (i) および (ii) のように出力を観測した結果、出力がともに規定出力ならば故障節点は $\{\overline{V(v_1, f(e))} \cap \overline{V(s(e), v_n)}\}$ 内に存在する。

以上の各 location 法に用いられる枝 e をとくに内部テスト端子 (internal test-edge or test-terminal) と呼び、この集合をそれぞれのテスト法に対して E_U^e 、 E_L^e 、 E_{UL}^e と記すことにする。

2.3 1-識別可能であるための必要十分条件

ここでは、上述の各テスト法に対して故障節点を確

*：以下この機能により行なう故障診断を第 5 章の場合と対比させて、“枝レベルの故障診断”と呼ぶ。

定的に抽出するための内部テスト端子の条件を調べる。

定義 2 UL法の内部テスト端子集合 $\mathbf{E}_{\text{UL}}$ によって a-cyclic SEC グラフ G (すなわち, システム) が 1-識別可能 (1-distinguishability) であるとは, つぎのような枝 $e (\in \mathbf{E}_{\text{UL}})$ が少なくとも 1 つ存在することである。すなわち, G の任意の節点对 $v_i, v_j (\in \mathbf{V})$ が枝 e によって区分される節点集合 $V(v_1, f(e)), V(s(e), v_n), \overline{V(v_i, f(e)) \cap V(s(e), v_n)}$ のうちの同一集合に含まないような枝 e である。(定義終)

定義 3 a-cyclic SEC グラフ G のすべての節点对 $v_i, v_j (\in \mathbf{V})$ に対して, $v_i \neq v_j$ かつ $v_i \neq v_j$ のとき, v_i と v_j の間を結ぶ双方向枝を挿入して得られるグラフを G_a と示す。このとき新たに挿入された双方向枝を dummy edge と呼び, その集合を $\mathbf{E}_a$ と記す。(定義終)

これらの定義を用いてあらわされる 1-識別可能の必要十分条件をつぎに示す。

定理 1 a-cyclic SEC グラフ G が UL 法の内部テスト端子 $\mathbf{E}_{\text{UL}}$ によって 1-識別可能であるための必要十分条件は次式が成立することである,

$$\bigcup_{\mathbf{E}_{\text{UL}}} \{(\text{U-DC} \langle e_i \rangle) \cup (\text{L-DC} \langle e_i \rangle)\} = \mathbf{E} \cup \mathbf{E}_a \quad \dots (1)^*$$

ここで, $e_i \in \mathbf{E}_{\text{UL}} \subseteq \mathbf{E}$ である。

<証明> 必要性; G が UL 法の意味で 1-識別可能ならば定義 2, 3 よりグラフ G の任意の節点对 v_i, v_j は少なくとも 1 つのテスト端子 $e' \in \mathbf{E}_{\text{UL}}$ による節点集合 $V(v_1, f(e')), V(s(e'), v_n), \overline{V(v_1, f(e')) \cap V(s(e'), v_n)}$ の同一節点集合に含まれない。それゆえ, v_i と v_j を結ぶ path に含まれる枝**の一つは e' の上方または下方有向カットセットに必ず含まれることになる。***すなわち, 式(1)が成立する。

十分性; 定義より G_a の任意の節点对 v_i, v_j 間には必ず path が存在する。いま, 式(1)が成立するとその path の枝は内部テスト端子の上方または下方有向カットセットに必ず含まれているから, とくに内部テスト端子 e' の上方有向カットセットに含まれているとする, すなわち

$$v_i \in V(v_1, f(e')), \quad v_j \in V(v_1, f(e'))$$

$$\text{または } v_j \in V(v_1, f(e')), \quad v_i \in V(v_1, f(e'))$$

*: 文献[1]の必要十分条件は定理 1 で $\mathbf{E}_a = \emptyset$ としグラフを対象にしたものに相当する。

** : このとき, v_i と v_j 間を結ぶ枝は dummy edge であることも存在する。

*** : ただし, G_a の dummy edge は $\text{U-DC} \langle e' \rangle$ and /or $\text{L-DC} \langle e' \rangle$ に含ませて考えるものとする。

となる。すると, v_i, v_j は常に $f(e')$ の出力を観測することによって識別可能となる。他方, v_i と v_j 間の枝が e' の下方有向カットセットに含まれる場合も同様である。以上のことがらが G_a の任意の節点对 v_i, v_j に対して成立することにより, グラフ G は $\mathbf{E}_{\text{UL}}$ で 1-識別可能となることがいえる。(証明終)

定義 4 節点数が n である a-cyclic SEC グラフ G の節点に任意に番号付けを行なってその 1 つを v_i とあらわす。(ただし, 入力節点は v_1 , 出力節点は v_n であるとする) そのとき, $i > j$ ならば $v_i \neq v_j$ が成立すれば, G を節点の ordering がなされた a-cyclic SEC グラフであるという。さらに, その ordering がただ 1 通りしかないとき, それを unique ordering と呼ぶ*。

unique ordering な a-cyclic SEC グラフの入力節点 v_1 から出力節点 v_n への最長有向道の枝集合を $\mathbf{E}_c$ であらわす。**(定義終)

すると, unique ordering な a-cyclic SEC グラフ G においては $G = G_a$ (すなわち $\mathbf{E}_a = \emptyset$) となる。このときは dummy edge を挿入する必要がなく, 定理 1 はさらにつぎのように簡単化される。

定理 2 unique ordering な a-cyclic SEC グラフ G が内部テスト端子 $\mathbf{E}_{\text{UL}}$ によって 1-識別可能であるための必要十分条件は次式が成立することである,

$$\bigcup_{\mathbf{E}_{\text{UL}}} \{(\text{U-DC} \langle e_i \rangle) \cup (\text{L-DC} \langle e_i \rangle)\} \supseteq \mathbf{E}_c \quad \dots (2)$$

ただし, $e_i \in \mathbf{E}_{\text{UL}} \subseteq \mathbf{E}$ である。

<証明> 必要性は定理 1 から明らかである。十分性; グラフ G の任意の節点对 v_i, v_j は必ず枝集合 $\mathbf{E}_c$ のつくる path 上にある。しかも $\mathbf{E}_c$ は式(2)によって $\mathbf{E}_{\text{UL}}$ の上方または下方有向カットセットに含まれている。したがって, v_i と v_j は定理 1 の十分性の証明と同様に処理されて識別可能となつて, 最終的には結論が得られる。(証明終)

つぎに, U(L)法に対する条件を与える。

定義 5 U(L) 法の内部テスト端子 $\mathbf{E}_{\text{UL}}^t (\mathbf{E}_{\text{L}}^t)$ によって a-cyclic SEC グラフ G が 1-識別可能であるとは, 任意の節点对 $v_i, v_j (\in \mathbf{V})$ が少なくとも 1 つの内部テスト端子 $e' \in \mathbf{E}_{\text{UL}}^t (\mathbf{E}_{\text{L}}^t)$ によって区分される節点集合 $V(v_1, f(e')), \overline{V(v_1, f(e')) \cap V(s(e'), v_n)}, \overline{V(s(e'), v_n)}$ のうちの同一集合に含まれないことである。

(定義終)

*: unique ordering が可能である必要十分条件は第 4 章, 4.2 節にあるアルゴリズムの step 4 を参照されたい。

** : このとき, $\dim \mathbf{E}_c = n-1$ となる。

定理 3 a-cyclic SEC グラフ G が $U(L)$ 法の内部テスト端子 $\mathbf{E}^{U^t} (\mathbf{E}^{L^t})$ によって 1-識別可能であるための必要十分条件は次式が成立することである,

$$\bigcup_{e_i \in \mathbf{E}^{U^t}} \{U\text{-DC} \langle e_i \rangle\} = \mathbf{E} \cup \mathbf{E}_d \left(\bigcup_{e_i \in \mathbf{E}^{L^t}} \{L\text{-DC} \langle e_i \rangle\} \right) = \mathbf{E} \cup \mathbf{E}_d \quad \dots(3)$$

ただし, $e_i \in \mathbf{E}^{U^t} \subseteq \mathbf{E}$ ($e_i \in \mathbf{E}^{L^t} \subseteq \mathbf{E}$) である。

(証明略)

定理 4 unique ordering な a-cyclic SEC グラフ G が $U(L)$ 法の内部テスト端子 $\mathbf{E}^{U^t} (\mathbf{E}^{L^t})$ によって 1-識別可能であるための必要十分条件は次式が成立することである,

$$\bigcup_{e_i \in \mathbf{E}^{U^t}} \{U\text{-DC} \langle e_i \rangle\} \supseteq \mathbf{E}_o \left(\bigcup_{e_i \in \mathbf{E}^{L^t}} \{L\text{-DC} \langle e_i \rangle\} \right) \supseteq \mathbf{E}_o \quad \dots(4)$$

ただし, $e_i \in \mathbf{E}^{U^t} \subseteq \mathbf{E}$ ($e_i \in \mathbf{E}^{L^t} \subseteq \mathbf{E}$) である。

(証明略)

2.4 絶対内部テスト端子の性質

定義 6 絶対内部テスト端子とは κ -テスト法によって 1-識別可能となるために, グラフの構造の如何を問わずに一意的に内部テスト端子とすべき枝をいい, $e_{k^{at}}$ と記す。また, その集合を $\mathbf{E}_{k^{at}}$ とあらわす。
($\kappa = U, L, UL$)

定義 7 $R(v)$ および $R^{-1}(v)$ をそれぞれつぎのように定める,

$$R(v) \triangleq \{v' \mid v' \in \mathbf{V}, v' \geq v\}$$

$$R^{-1}(v) \triangleq \{v' \mid v' \in \mathbf{V}, v \geq v'\}$$

絶対内部テスト端子の性質をのべるまえに, つぎの 2 つの補題を与えておく。

補題 1 $e \in U\text{-DC} \langle e_0 \rangle$ と,

$$e_0 \in f^{-1}\{R^{-1}(f(e)) - R^{-1}(s(e))\}, \quad e, e_0$$

$\in \mathbf{E}$ は等価である。

($e \in L\text{-DC} \langle e_0 \rangle$ と, $e_0 \in s^{-1}\{R(s(e)) - R(f(e))\}$, $e, e_0 \in \mathbf{E}$ は等価である。)

<証明> $\mathbf{V}_R \triangleq R(f(e_0))$ とすると定義 1, 7 より $U\text{-DC} \langle e_0 \rangle = f^{-1}(\mathbf{V}_R) - s^{-1}(\mathbf{V}_R)$ となる。他方, $f^{-1}(\mathbf{V}_R) \supset s^{-1}(\mathbf{V}_R)$ であるから $U\text{-DC} \langle e_0 \rangle \ni e$ となるのは $f^{-1}(\mathbf{V}_R) \ni e$ かつ $s^{-1}(\mathbf{V}_R) \not\ni e$ の場合である。それゆえ $\mathbf{V}_R \ni f(e)$, $\mathbf{V}_R \not\ni s(e)$ となる。したがって, $R(f(e_0)) \ni f(e)$ かつ $R(f(e_0)) \not\ni s(e)$ であるから, これらを変形すれば結論を得る。この逆も成立する。

(証明終)

補題 2 $e \in U\text{-DC} \langle e_0 \rangle$ かつ, $\dim[f^{-1}\{R^{-1}(f(e)) - R^{-1}(s(e))\}] = 1$ ならば $e = e_0$ である。($e \in L\text{-DC} \langle e_0 \rangle$ かつ, $\dim[s^{-1}\{R(s(e)) - R(f(e))\}] = 1$ ならば $e = e_0$ である。)

<証明> 条件より $\{R^{-1}(f(e)) - R^{-1}(s(e))\}$ には 1 つの節点要素しかないから, それを v_0 とすれば $R^{-1}(f(e)) = e_0 \cup R^{-1}(s(e))$ となる。ゆえに, $v_0 = R^{-1}(f(e)) - R^{-1}(s(e)) = f(e)$ となる。また, $\dim\{f^{-1}(f(e))\} = 1$ であるから $od(f(e)) = 1$ となって, $e_0 \in f^{-1}(f(e)) = e$ となる。したがって, $e = e_0$ となる。

(証明終)

定理 5 枝 e が $U(L)$ 法の絶対内部テスト端子であることと, つぎの 3 つの事柄は等価である。

(i) 枝 e を被覆する上(下)方有向カットセットはその枝自身による上(下)方有向カットセットのみである。

(ii) $od(f(e)) = 1$ ($id(s(e)) = 1$)

(iii) $\dim[f^{-1}\{R^{-1}(f(e)) - R^{-1}(s(e))\}] = 1$
($\dim[s^{-1}\{R(s(e)) - R(f(e))\}] = 1$)

<証明> (iii) ならば補題 2 によって (ii) が成立することがいえる。(ii) ならば補題 1 において $od(f(e)) = 1$ を代入すれば $e_0 \in e$ となり明らかに (i) が成立する。そして, (i) ならば補題 1 において $e = e_0$ とおいたことになるから (iii) が成立する。(証明終)

UL 法の絶対内部テスト端子についてはつぎの定理が成立する。

定理 6 枝 e が UL 法の絶対内部テスト端子であることと, つぎの 3 つの事柄は等価である。

(i) 枝 e を被覆する上方または下方有向カットセットはその枝自身による上方および下方有向カットセットのみである。

(ii) $od(f(e)) = id(s(e)) = 1$

(iii) $\dim[f^{-1}\{R^{-1}(f(e)) - R^{-1}(s(e))\}] = \dim[s^{-1}\{R(s(e)) - R(f(e))\}] = 1$

<証明> UL 法の絶対内部テスト端子は U 法かつ L 法に対して絶対内部テスト端子となるもののみである。したがって, 証明は定理 5 のそれから明らかである。(証明終)

以上, 本章の結果は第 4 章でのべる最少内部テスト端子の決定アルゴリズムを考えるための基礎となるものである。

3. 内部テスト端子数の上, 下限 [11], [14]~[16]

本章では, 節点数 n をもったすべての a-cyclic SEC グラフを考えたもとで, 第 2.2 節でのべた各故障診断

法によってグラフが1-識別可能となるための内部テスト端子数の上, 下限を求めている。従来, この下限に関しては明確にされておらず, わずかに UL 法の推測式が与えられているにすぎない。[1]

3.1 vertex range の諸性質

まず, 第2.1節で定義した vertex range のいくつかの性質を考察する。

補題 3 a-cyclic SEC グラフ G の任意の枝 e に対して

$$V(v_1, f(e)) \cap V(s(e), v_n) = \phi$$

となる。

<証明> いま, $V(v_1, f(e)) \cap V(s(e), v_n) \ni v_0 \neq \phi$ とすると $V(v_1, f(e)) \ni v_0$ かつ $V(s(e), v_n) \ni v_0$ であるから $f_0 \geq f(e)$ かつ $s(e) \geq v_0$ となる。他方, $f(e) \geq s(e)$ であるから, 全体として $v_0 \geq f(e) \geq s(e) \geq v_0$ となって cycle を形成する。このことは G が a-cyclic SEC グラフであったことに反するから命題が成立する。(証明終)

定義 8 UL 法の内部テスト端子 $E_{tL} = \{e_1^t, e_2^t, \dots, e_N^t\}$ によって得られる D-partition とは, 次式によって与えられる節点集合の分割をいう,

$$\bigcap_{i=1}^N \left[\left\{ V(v_1, f(e_i^t)) \text{ or } \overline{V(v_1, f(e_i^t))} \right\} \cap \left\{ \overline{V(s(e_i^t), v_n)} \text{ or } V(s(e_i^t), v_n) \right\} \right] \quad \dots\dots(5)$$

(定義終)

すると, この UL 法の D-partition にはつぎの重要な性質がある。

定理 7 UL 法の内部テスト端子 $\{e_1^t, e_2^t, \dots, e_N^t\}$ によって得られた D-partition が $\{V_1, V_2, \dots, V_k\}$ であったとする。このとき, さらに新たな内部テスト端子 e_{N+1}^t を加えて得られる D-partition において, もとの $V_i (i=1, 2, \dots, k)$ は3つ以下に分割され, 3つに分割されるもとの V_i は1つ以下である。

<証明> $\{e_1^t, e_2^t, \dots, e_N^t\}$ による D-partition は式(5)によって

$$\bigcap_{i=1}^N \left[\left\{ V(v_1, f(e_i^t)) \text{ or } \overline{V(v_1, f(e_i^t))} \right\} \cap \left\{ \overline{V(s(e_i^t), v_n)} \text{ or } V(s(e_i^t), v_n) \right\} \right] \\ = \{V_1, V_2, \dots, V_k\} \quad \dots\dots(6)$$

となる。すると, 式(5)および(6)によって $\{e_1^t, e_2^t, \dots, e_N^t, e_{N+1}^t\}$ の D-partition は

$$V_i \cap \left\{ V(v_1, f(e_{N+1}^t)) \text{ or } \overline{V(v_1, f(e_{N+1}^t))} \right\}$$

$$\cap \left\{ \overline{V(s(e_{N+1}^t), v_n)} \text{ or } V(s(e_{N+1}^t), v_n) \right\} \quad \dots\dots(7)$$

$$(l=1, 2, \dots, k)$$

で与えられる。さらに補題 3 によって, 式(7)は

$$\{V_i \cap V(v_1, f(e_{N+1}^t)) \cap \overline{V(s(e_{N+1}^t), v_n)}, \\ V_i \cap \overline{V(v_1, f(e_{N+1}^t))} \cap V(s(e_{N+1}^t), v_n), \\ V_i \cap \overline{V(v_1, f(e_{N+1}^t))} \cap \overline{V(s(e_{N+1}^t), v_n)}\} \quad \dots\dots(8)$$

と変形されるので, V_i と vertex range の包含関係を考慮すれば, V_i が3つ以下に分割されることがいえる。($l=1, 2, \dots, k$)

つぎに, 3つに分割されるもとの V_j は1つ以下であることを示すために, $f(e_{N+1}^t), s(e_{N+1}^t)$ および V_j の包含関係によって以下の (i) および (ii) の場合に分けて検討する。

(i) $V_j \supset \{f(e_{N+1}^t), s(e_{N+1}^t)\}$ の場合: このとき, V_j は内部テスト端子 e_{N+1}^t によってさらにつぎのように分割される,

$$V_j \cap V(v_1, f(e_{N+1}^t)) \cap \overline{V(s(e_{N+1}^t), v_n)} \ni f(e_{N+1}^t) \neq \phi \\ \overline{V_j \cap V(v_1, f(e_{N+1}^t)) \cap V(s(e_{N+1}^t), v_n)} \ni s(e_{N+1}^t) \neq \phi \\ \overline{V_j \cap V(v_1, f(e_{N+1}^t)) \cap \overline{V(s(e_{N+1}^t), v_n)}} = \phi \text{ or } \neq \phi$$

したがって, V_j は3つ以下に分割される。さらに, e_{N+1}^t によって3つに分割される旧の D-partition の要素が高々1つ (すなわち, 高々 V_j だけ) であることを示すために, 対偶をとる。すなわち, e_{N+1}^t によ

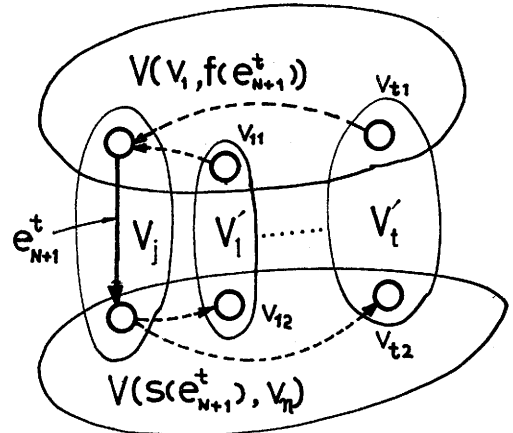


図1 定理7を証明するための説明図

って V_j の他に3つに分割される旧の D-partition の要素を $V_1', V_2', \dots, V_k'$ と仮定する。(図1参照) すると

$$V_s' \cap V(v_1, f(e_{N+1}^t)) \ni v_{s1} \neq \phi,$$

$$V_s' \cap V(s(e_{N+1}^t), v_n) \ni v_{s2} \neq \phi$$

なる v_{s1} および v_{s2} が存在して

$$v_{s1} \geq f(e_{N+1}^t) \quad s(e_{N+1}^t) \geq v_{s2}$$

となる ($s=1, 2, \dots, t$)。したがって、 v_{s1} から $f(e_{N+1}^t)$ に向う path に含まれる枝集合のうちで、 V_s' から $\overline{V_s'}$ に向う枝が必ず存在するからその枝を e_s とする。また、 V_s' なる D-partition の要素が存在することから e_s を上方または下方有向カットセットにもつ内部テスト端子が存在するので、その端子を e_s^t と記すことにする ($s=1, 2, \dots, t$)。そこで、まず、 e_s^t の上方有向カットセットに e_s が含まれる場合を考える。このとき、 $v_{s2} \neq v_{s1}$ であり ($\therefore$ cycle をつくらないことを要するから)、 $v_{s1} \geq f(e_s^t)$ であるから、 $V(v_1, f(e_s^t)) \ni v_{s1}$ 、 $V(v_1, f(e_s^t)) \ni v_{s2}$ となる。すると、 V_s' が内部テスト端子 $\{e_1^t, e_2^t, \dots, e_N^t\}$ によって得られた D-partition の要素でないことになるが、これは前述の仮定に反する。したがって、 e_{N+1}^t によって3つに分割される可能性のある旧の D-partition の要素は V_j だけであるという結論を得る ($s=1, 2, \dots, t$)。他方、 e_s が e_s^t の下方有向カットセットに含まれる場合も同様に証明される。

(ii) (i) 以外の場合： $V_i \ni f(e_{N+1}^t)$ かつ $V_j \ni s(e_{N+1}^t)$ なる要素 V_i と $V_j (i \neq j)$ 、または $V_k \ni (f(e_{N+1}^t), s(e_{N+1}^t))$ なる要素 V_k が e_{N+1}^t によってどのように分割されるかが残っている。まず、前者の場合を考える。すると、 V_i, V_j は e_{N+1}^t によってつぎのように分割されることがいえる、

$$\left. \begin{aligned} V_i \cap V(v_1, f(e_{N+1}^t)) \cap \overline{V(s(e_{N+1}^t), v_n)} \\ \quad \ni f(e_{N+1}^t) \neq \phi \quad \dots(a) \\ V_i \cap \overline{V(v_1, f(e_{N+1}^t))} \cap V(s(e_{N+1}^t), v_n) \\ \quad = \phi \quad \dots(b) \\ V_i \cap \overline{V(v_1, f(e_{N+1}^t))} \cap \overline{V(s(e_{N+1}^t), v_n)} \\ \quad = \phi \text{ or } \neq \phi \quad \dots(c) \end{aligned} \right\} \quad (9)$$

$$\left. \begin{aligned} V_j \cap V(v_1, f(e_{N+1}^t)) \cap \overline{V(s(e_{N+1}^t), v_n)} \\ \quad = \phi \quad \dots(a) \\ V_j \cap \overline{V(v_1, f(e_{N+1}^t))} \cap V(s(e_{N+1}^t), v_n) \\ \quad \ni s(e_{N+1}^t) \neq \phi \quad \dots(b) \\ V_j \cap \overline{V(v_1, f(e_{N+1}^t))} \cap \overline{V(s(e_{N+1}^t), v_n)} \\ \quad = \phi \text{ or } \neq \phi \quad \dots(c) \end{aligned} \right\} \quad (10)$$

式(9 b)および式(10 a)以外は式(5)より明らかである。したがって、式(9 b)、(10 a)が成立することをいえば、この場合の V_i および V_j は高々2個に分割されることになる。ここでは、式(9 b)の成立することを代表

として示すが、それには式(9 b)の十分条件である次式が成立することを示せばよい

$$V_i \cap V(s(e_{N+1}^t), v_n) = \phi$$

いま、対偶をとって $V_i \cap V(s(e_{N+1}^t), v_n) \ni v_j \neq \phi$ とする。 $V_i \ni f(e_{N+1}^t)$ 、 $V_j \ni s(e_{N+1}^t)$ の場合には e_{N+1}^t は $e_1^t, e_2^t, \dots, e_N^t$ のいずれかの上方または下方有向カットセットに含まれている。そこで、まず、 e_{N+1}^t が e_j^t の上方有向カットセットに含まれていたとする。すると、 v_j は $V(v_1, f(e_j^t)) \ni v_j$ または $\ni v_j$ のいずれかであるから、まず $V(v_1, f(e_j^t)) \ni v_j$ とする。他方、 $V_i \cap V(s(e_{N+1}^t), v_n) \ni v_j$ なる仮定から $s(e_{N+1}^t) \geq v_j$ であるから

$$V(v_1, f(e_j^t)) \ni v_j, f(e_{N+1}^t), s(e_{N+1}^t)$$

となって、 e_j^t の上方有向カットセットに e_{N+1}^t が含まれなくなる。したがって、 $V(v_1, f(e_j^t)) \ni v_j$ とならなければならないので、以上の場合には存在しないことになる。

つぎに、 $V(v_1, f(e_j^t)) \ni v_j$ の場合を考える。このときも次式が成立する。

$$V(v_1, f(e_j^t)) \ni f(e_{N+1}^t), \quad V_i \ni v_j, f(e_{N+1}^t)$$

すると、 V_i は $e_j^t (j=1, 2, \dots, N)$ によって得られる D-partition の要素でなくなる。これは V_i が旧の D-partition の要素であったことに反するから $V_i \cap V(s(e_{N+1}^t), v_n) = \phi$ とならなければならない。したがって結論を得る。他方、 e_{N+1}^t が e_j^t の下方有向カットセットに含まれている場合も同様に証明される。なお、式(10 a)についても上記と同様に証明される。

最後に、 V_i, V_j 以外の D-partition の要素、すなわち、 $V_k \ni (f(e_{N+1}^t), s(e_{N+1}^t))$ なる V_k が e_{N+1}^t によって3つに分割されないことは(i)の場合の V_s' なる要素について考えたと同様に示される。

以上ですべての場合が尽された。(証明終)

つぎの定理8は vertex range の定義より明らかである。

定理8 a-cyclic SEC グラフ G において

$$V(v_1, v_i) \supseteq V(v_1, v_j) \quad (V(v_i, v_n) \supseteq V(v_j, v_n))$$

となるための必要十分条件は

$$v_j \geq v_i \quad (v_i \geq v_j)$$

である。

(証明略)

定理9 $v_i \neq v_j$ ならば次式が成立する

$$\{V(v_1, v_j)\} \cap \{V(v_i, v_n)\} = \phi$$

<証明> $V(v_1, v_j) = \{v | v_1 \geq v, v \geq v_j\}$ 、 $V(v_i, v_n) = \{v | v_i \geq v, v \geq v_n\}$ であるから、もし $\{V(v_1, v_j)\} \cap \{V(v_i, v_n)\} \ni v_0 \neq \phi$ とすると、 $v_0 \geq v_j, v_i \geq v_0$ となり、

$v_i \geq v_j$ となるから、命題が成立する。(証明終)

3.2 故障表とグラフ構造の関係

故障診断のすべての情報を内在する表、すなわち故障表とグラフ構造の関係を調べる。まず、故障表の定義を明示する。

定義9 a-cyclic SEC グラフ G において内部テスト端子 $\mathbf{E}^t = \{e_1^t, e_2^t, \dots, e_N^t\}$ の故障表とは、行に全節点 $\mathbf{V}$ を、列に内部テスト端子 $\mathbf{E}^t$ をそれぞれ対応させたとき、要素 a_{ij} がつぎのように定められる行列 ($\kappa = \text{U, L}$ のとき $n \times N$ 次, $\kappa = \text{UL}$ のとき $n \times 2N$ 次の行列) である。

$$\begin{aligned} \text{U法}(\kappa = \text{U}) \quad & \begin{cases} v_i \in V(v_1, f(e_j^t)) \text{ ならば } a_{ij} = 1 \\ v_i \notin V(v_1, f(e_j^t)) \text{ ならば } a_{ij} = 0 \end{cases} \\ \text{L法}(\kappa = \text{L}) \quad & \begin{cases} v_i \in V(s(e_j^t), v_n) \text{ ならば } a_{ij} = 1 \\ v_i \notin V(s(e_j^t), v_n) \text{ ならば } a_{ij} = 0 \end{cases} \\ \text{UL法}(\kappa = \text{UL}) \quad & \begin{cases} v_i \in V(v_1, f(e_j^t)) \text{ ならば } a_{ij} = 1 \\ v_i \notin V(v_1, f(e_j^t)) \text{ ならば } a_{ij} = 0 \\ v_i \in V(s(e_j^t), v_n) \text{ ならば } a_{i, N+j} = 1 \\ v_i \notin V(s(e_j^t), v_n) \text{ ならば } a_{i, N+j} = 0 \end{cases} \\ & i=1, 2, \dots, n \quad j=1, 2, \dots, N \end{aligned}$$

なお、故障表の第 i 番目の行ベクトルを $\mathbf{a}_i$ で示す (定義終)

たとえば、図2のグラフで、 e_1^t, e_2^t および e_3^t をUL法の内部テスト端子としたときの故障表を求めると表1に示すようになる。

つぎに、0, 1 要素からなる任意の $n \times N$ 次行列がU(L)法の故障表であるための必要十分条件を与える。

定理10 同じ行および列ベクトルをもたない0, 1要素からなる n 行 N 列行列 $[\mathbf{a}_1^T, \mathbf{a}_2^T, \dots, \mathbf{a}_n^T]^T$ がU(L)

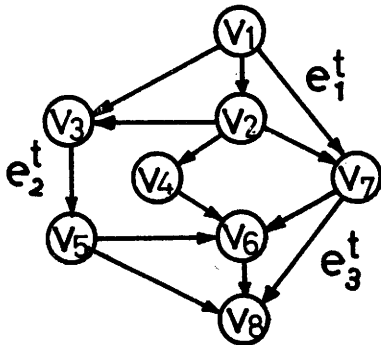


図2 a-cyclic SEC グラフ G

	e_1^t	e_2^t	e_3^t	e_1^t	e_2^t	e_3^t
V_1	1	1	1	0	0	0
V_2	0	1	1	0	0	0
V_3	1	0	1	0	0	0
V_4	0	0	0	0	0	0
V_5	0	0	0	0	1	0
V_6	0	0	0	1	1	0
V_7	0	0	1	1	0	0
V_8	0	0	0	1	1	1

表1 図2に対するUL法の故障表

法の故障表であるための必要十分条件は次式をみたす行ベクトル $\mathbf{a}_1^0, \mathbf{a}_2^0, \dots, \mathbf{a}_N^0$ が存在することである。

$$\begin{pmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_n \end{pmatrix} \cdot \begin{pmatrix} \mathbf{a}_1^0 \\ \mathbf{a}_2^0 \\ \vdots \\ \mathbf{a}_N^0 \end{pmatrix} = \begin{pmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_n \end{pmatrix} \quad \dots\dots(11)$$

ただし、行列の積演算において和は二値論理和および積は二値論理積として行なわれるものとし、さらに、

$$\mathbf{a}_1 \triangleq [1, 1, \dots, 1], \mathbf{a}_n \triangleq [0, 0, \dots, 0]$$

$$\mathbf{a}_j^0 \in \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n\} \quad j=1, 2, \dots, N$$

とする。以後 $\mathbf{a}_1^0, \mathbf{a}_2^0, \dots, \mathbf{a}_N^0$ を故障表の基底と呼ぶ。

<証明> この定理で同じ行ベクトルをもたないとするのは1-識別可能な a-cyclic SEC グラフを考えているからであり、同じ列ベクトルをもたない行列を考えているのはつぎの理由による。列ベクトルに同じものが k 個 ($k \geq 2$) 存在する行列を故障表としてもつ a-cyclic SEC グラフが存在する場合と存在しない場合がある。しかし、グラフが存在する場合においても必ず $(k-1)$ 個の内部テスト端子は冗長である。ここでは、このような冗長性のある内部テスト端子ははじめから考えていないからである。したがって、同じ行および列ベクトルを有しない行列だけを対象にすれば十分である。

十分性: $[\mathbf{a}_1^T, \mathbf{a}_2^T, \dots, \mathbf{a}_n^T]^T$ の基底が $\mathbf{a}_1^0, \mathbf{a}_2^0, \dots, \mathbf{a}_N^0$ であるから、いま $\mathbf{a}_i^0$ を内部テスト端子 e_i^t の始点(終点)に対応する行ベクトルとする。他方、式(11)は行列 $[\mathbf{a}_1^T, \mathbf{a}_2^T, \dots, \mathbf{a}_n^T]^T$ の各行ベクトル $\mathbf{a}_j$

がいくつかの基底の論理和であらわされることを示している。このことから行列 $[a_1^T, a_2^T, \dots, a_n^T]^T$ の各行を節点に対応させれば $v_i^t \triangleq f(e_i^t)$ ($v_i^t \triangleq s(e_i^t)$) と全節点 $\mathbf{v}$ との到達関係が一意に定まる。ここで、 $i=1, 2, \dots, N$ 。また、入力節点 v_1 および出力節点 v_n はそれぞれ a_1 および a_n (a_n および a_1) に対応する節点であり、いま考えている行列においてはすべての行ベクトルが異なっているからグラフは cycle を含まない。したがって、 $[a_1^T, a_2^T, \dots, a_n^T]^T$ を故障表とする a-cyclic SEC グラフが存在する。

必要性：同一行および列ベクトルをもたない $n \times N$ 次の故障表が与えられるとき、グラフ G では内部テスト端子 $e_1^t, e_2^t, \dots, e_N^t$ が定理3または4により既知となっている。また、その故障表は $f(e_j^t)$ ($s(e_j^t)$) に対応する N 個の行ベクトルをも含んでいる。他方、故障表の定義9から故障表の第 i 行目の行ベクトルの意味することはつぎの2つであることがいえる、(1)その行ベクトルに対応する節点 v_i がどの $f(e_j^t)$ に到達可能であるか？（その行ベクトルに対応する節点 v_i にどの $s(e_j^t)$ から到達可能であるか？）(2)さらに、 v_i から (に) 到達可能な $f(e_j^t)$ ($s(e_j^t)$) には他の内部テスト端子の始点（終点）を経ないで直接に到達可能なものと間接的に到達可能なものがある。すなわち、 v_i から (に) 直接到達可能な $f(e_p^t)$ ($s(e_p^t)$) があるとき、 $f(e_p^t)$ から ($s(e_p^t)$ に) さらに $f(e_q^t)$ に ($s(e_q^t)$ から) 到達可能ならば、当然 v_i は $f(e_q^t)$ にも ($s(e_q^t)$ からも) 到達可能である。

つぎに目を式(II)に転ずる。すなわち、 $f(e_j^t)$ ($s(e_j^t)$) に対応する行ベクトルを a_j^0 とする*。すると、上記(1)の意味するところは、式(II)左辺で v_j に相当する $[a_1^T, a_2^T, \dots, a_n^T]^T$ の行ベクトルと $[a_1^0, a_2^0, \dots, a_n^0]^T$ との論理積をとることに相当し、(2)の意味するところは式(II)の左辺で(1)の結果零行ベクトルとならなかった $[a_1^0, \dots, a_n^0]^T$ の行論理和をとることに相当する。したがって、式(II)が成立する。ただし、 $j=1, 2, \dots, N$ (証明終)

つぎに、UL法の故障表について考えると、それはつぎの2つの型に分類される。

A型：故障表の行ベクトル a_j ($1 \times 2N$ 次) が第1から第 N 要素まですべて零であるか、または第 $(N+1)$ から第 $2N$ 要素まですべて零であるもの。 $j=1, 2, \dots, n$ (表2-(a)参照)

* $f(e_j^t)$ ($s(e_j^t)$) に対応する故障表の行ベクトルは、独立な行ベクトルであることによる。このことは故障表が列 full rank であることからいえる。

B型：A型の行ベクトルの他に、第1から第 N 要素までに非零要素を含むとともに第 $(N+1)$ から第 $2N$ 要素までも非零要素を含む行ベクトルが存在するもの。(表2-(b)参照)

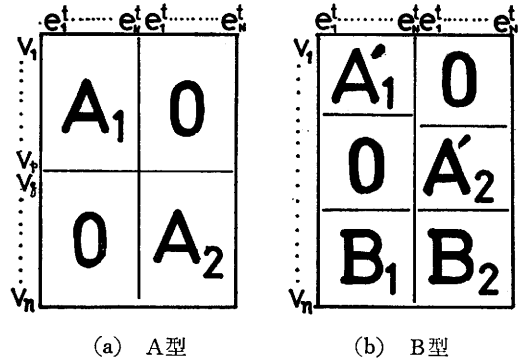


表2 UL法の故障表

UL法の故障表であるための必要十分条件はA型に対して求めると定理11のごとくなるが、B型に対しては現在のところ明らかにされていない。

定理11 0, 1要素からなる同一行および列ベクトルを有しない ($n \times 2N$) 次のA型の行列がUL法の故障表であるための必要十分条件は表2-(a)における A_1 および A_2 なる部分行列にそれぞれ定理10でのべた基底が存在することである。($a_n = [0, 0, \dots, 0]$ は含まなくてもよい。)

<証明> A型の行列の行に表2-(a)のごとく節点 $v_1, v_2, \dots, v_p, v_q, \dots, v_n$ を対応させると、 v_1 から v_p はテスト法Uによって、 v_q から v_n まではテスト法Lによってそれぞれ識別される。したがって、定理10より命題が成立することは明らかである。(証明終)

これらの定理によって、故障表が与えられたときその内部テスト端子の始点および終点が求められ、さらには後述の内部テスト端子数の下限を与えるグラフ構造ならびに生成法などを議論することが可能であるが、詳細は別報にゆずる。[14]~[16]

3.3 カットセット閉空間の定義

つぎに、内部テスト端子数の下限、すなわち、つぎの問題を考える。

問1 節点数 n が与えられたとき、可能なすべての a-cyclic SEC グラフを1-識別可能にするために必要な内部テスト端子数の最小値 N_k はいくらか？ 換言すれば、 N_k 個の内部テスト端子によって1-識別可能と

なるグラフの節点数の最大値はいくらか？ただし、 $\kappa=U, L, UL$ である。

この問の答を求めるために、まずカットセット閉空間なるものを定義する*。

カットセット閉空間 $S_U(i)$ および $S_L(i)$ はグラフの vertex range に、素空間 S_j はグラフの D-partition の各要素に対応づけられた概念である、すなわち、

$$\left. \begin{array}{ll} \text{全閉空間 } S & \Leftrightarrow \text{全節点 } V \\ S_U(i) & \Leftrightarrow V(v_1, f(e_i^t)) \\ S_L(i) & \Leftrightarrow V(s(e_i^t), v_n) \end{array} \right\} \dots (12)$$

補題3 および式(12)の対応関係を考慮に入れて $S_U(i)$ と $S_L(i)$ につぎの性質を付与する。

定義10 カットセット閉空間 $S_U(i)$ と $S_L(i)$ は $S_U(i), S_L(i) \subset S, S_U(i) \cap S_L(i) = \phi, i=1, 2, \dots, N$ なる性質をもつ。

定義11 $S_U(1), S_U(2), \dots, S_U(N) (S_L(1), S_L(2), \dots, S_L(N))$ の素空間とは次式によって得られる意味ある空間**をいう。

$$\left\{ \bigcap_{i=1}^N \{S_U(i) \text{ or } \overline{S_U(i)}\}, \overline{S_U(i)} \triangle S - S_U(i) \right\} \\ \left\{ \bigcap_{i=1}^N \{S_L(i) \text{ or } \overline{S_L(i)}\}, \overline{S_L(i)} \triangle S - S_L(i) \right\}$$

また、 $S_U(i)$ および $S_L(i), i=1, 2, \dots, N$ の素空間とは次式によって得られる意味ある空間をいう。

$$\bigcap_{i=1}^N \left(\{S_U(i) \text{ or } \overline{S_U(i)}\} \cap \{S_L(i) \text{ or } \overline{S_L(i)}\} \right) \text{ (定義終)}$$

この定義から素空間 $S_p (p=1, 2, \dots, n)$ は必ず $S_i \cap S_j = \phi$ (ただし、 $i \neq j$), $\bigcup_{p=1}^n S_p = S$ が成立することがわかる。

さらに、定理7 および式(12)の対応関係を考慮して、 $S_U(i)$ と $S_L(i) i=1, 2, \dots, N$ につぎの性質を付与する。

*U(L)法に対する答は故障表の基底を用いても求められる。しかし、UL法の答を求めるためにカットセット閉空間なる概念を用いた方が都合がよいので、ここでは統一してU(L)法の場合も含めてカットセット閉空間を用いて論ずることにする。

**： $\bigcap_{i=1}^N \{S_U(i) \text{ or } \overline{S_U(i)}\}$ の計算において結果が空に

なる場合がある。意味ある空間とはこのような場合を除いたものをいう。

定義12 $S_U(i)$ および $S_L(i) i=1, 2, \dots, N$ のつくる素空間はつぎの性質をもつ。 $S_U(i)$ および $S_L(i) i=1, 2, \dots, N$ のつくる素空間を $S_1, S_2, \dots, S_n$ とするとき、さらに新たなカットセット閉空間 $S_U(N+1)$ および $S_L(N+1)$ を加えて得られた新たな素空間において、旧の素空間 $S_p, p=1, 2, \dots, n$ はそれぞれ3つ以下の素空間に分割され、さらに、3つに分割される旧の素空間は高々1つである。(定義終)

定義13 $S_U(i)$ と $S_U(j) (S_L(i) \text{ と } S_L(j))$ が包含関係をもつとはつぎの関係が成立することをいう。

$$S_U(i) \cap S_U(j) = S_U(i) \text{ or } S_U(j), (S_L(i) \cap S_L(j) = S_L(i) \text{ or } S_L(j)) \text{ (定義終)}$$

このように定義されたUおよびLカットセット閉空間 $S_U(i), S_L(i)$ においてつぎの問題を考える。

問II $S_U(i)(S_L(i)) i=1, 2, \dots, N$ によって構成される素空間数の最大値はいくらか。

問III $S_U(i)$ かつ $S_L(i) i=1, 2, \dots, N$ によって構成される素空間数の最大値はいくらか。

補題4 $S_U(i) (S_L(i)) i=1, 2, \dots, N$ の素空間が $\{S_1, S_2, \dots, S_\alpha\}$ であったとする。新たなU(L)カットセット閉空間 $S_U(N+1) (S_L(N+1))$ を加えた $S_U(i) (S_L(i)) i=1, 2, \dots, N, N+1$ によってつくられる素空間数は 2α 以下である。また 2α になるための必要十分条件は、

$$S_j \cap S_U(N+1) = \phi, S_j \cap \overline{S_U(N+1)} = \phi, j=1, 2, \dots, \alpha \\ (S_j \cap S_L(N+1) = \phi, S_j \cap \overline{S_L(N+1)} = \phi, j=1, 2, \dots, \alpha) \text{ である。 (証明略)}$$

補題4によって明らかなようにつぎの定理が得られ、それは問IIの解でもある。

定理12 $S_U(i) (S_L(i)) i=1, 2, \dots, N$ による素空間数の最大値は 2^N である。(証明略)

つぎに問IIIの答はつぎの定理によって与えられる。

定理13 $S_U(i)$ および $S_L(i), i=1, 2, \dots, N$ による素空間数 β_N の最大値は $2^{N+1}-1$ である。また $\beta_N = 2^{N+1}-1$ となるときは $S_U(i)$ および $S_L(i) i=1, 2, \dots, N$ が包含関係をもたない場合である。

<証明> $N=1$ のときは明らかに $\beta_1 \leq 2^{1+1}-1=3$ が成り立つ。いま、 $\beta_{N-1} \leq 2^{(N-1)+1}-1$ が成立したとする。定義12によって $S_U(i)$ と $S_L(i) i=1, 2, \dots, j$ によって構成される素空間数 β_j は

$$\beta_j \leq 2\beta_{j-1} + 1, j=2, 3, \dots, N \dots (13)$$

である。したがって、式(13)を用いて β_N を計算すると、

$$\beta_N \leq (2^{(N-1)+1}-1) \times 2 + 1 = 2^{N+1}-1$$

となる。ここで等号の成立する場合は式(13)において $j=2, 3, \dots, N$ のすべてに対して等号が成立する場合

であるから $S_U(i)$ および $S_L(i)$ $i=1, 2, \dots, N$ が互いに包含関係をもたない場合である。(証明終)

3.4 U(L)法における内部テスト端子数の下限

$S_U(i)$ ($S_L(i)$) による素空間 S_j につきのように N 次元ベクトル $\mathbf{b}_j$ を対応させる。

$S_j \in S_U(i)$ ($S_j \in S_L(i)$) ならば $b_{ji}=1$

$S_j \notin S_U(i)$ ($S_j \notin S_L(i)$) ならば $b_{ji}=0$

ただし, $\mathbf{b}_j = [b_{j1}, b_{j2}, \dots, b_{jN}]$, $i=1, 2, \dots, N$ および $j=1, 2, \dots, 2^N$ である。

S_j は素空間であるから $j \neq k$ ならば必ず $\mathbf{b}_j \neq \mathbf{b}_k$ である。したがって, $\mathbf{b}_j$ を表に示すと表3になる*。

前にのべた式(12)の対応関係を表3に適用すれば表4を得る。表4は定義9によって, 内部テスト端子 $\mathbf{E}^t(\mathbf{E}^t) = \{e_1^t, e_2^t, \dots, e_N^t\}$ による故障表に対応することになる。表4において, $\dim V_j > 1$ ならば V_j に含まれる節点は, $\mathbf{E}^t(\mathbf{E}^t)$ による故障診断に対して同一情報を与えるから, 互いに1-識別可能とはならない。

*: 表3の S_i および表5の v_i はともに1行を意味するが, 表4の V_i は一般に1行以上を意味する。

$\mathbf{E}^t(\mathbf{E}^t)$ によって1-識別可能であるためには, $\dim V_j = 1$, ($j=1, 2, \dots, 2^N$) でなければならない。すると表4は表5になる。しかし, カットセット閉空間の定義は, それに対応する vertex range をもつ a-cyclic SEC グラフが存在するための必要条件であって十分条件ではない。したがって, 一般には表5を故障表としてもつ a-cyclic SEC グラフの存在することを確認しなければならない。ところが, 定理10によって表5を故障表としてもつ a-cyclic SEC グラフが存在することは明らかである。なぜなら, 故障表の基底を $v_{(2^N-N-1)}$ から $v_{(2^N-1)}$ に対応する行ベクトルに選べば式(11)をみたすからである。したがって, このことと定理13からテスト法U(L)の場合の問Iの答を得る。

定理14 n 個の節点をもつ可能なすべての a-cyclic SEC グラフにおいて, それをU(L)法によって1-識別可能にするための内部テスト端子数 N はつぎのようになる。

$$n \leq 2^N \dots\dots\dots (14)$$

$$\text{すなわち, } N \geq \lceil \log_2 n \rceil \dots\dots\dots (15)$$

ここで, $\lceil x \rceil$ は x 以上の最小の整数を示す。(定理終)

	$S_1, S_2, \dots, S_N$ ($S_1, S_2, \dots, S_N$)
S_1	1 1 1
S_2	0 0 1
$\vdots$	$\vdots$
S_{N+1}	1 0 0
$\vdots$	$\vdots$
$S_{\alpha i}$	1 1 1 0 0 1 1 1 0 0
$\vdots$	$\vdots$
$S_{\beta i}$	0 0 0 1 1
$\vdots$	$\vdots$
S_{2^N}	1 1 0 0 0 1
	0 0 0
	$\beta_i - \alpha_i + 1 = N C_i$

表 3

	$e_1^t, e_2^t, \dots, e_N^t$
V_1	1 1 1
V_2	0 0 1
$\vdots$	$\vdots$
V_{N+1}	1 0 0
$\vdots$	$\vdots$
$V_{\alpha i}$	1 1 1 0 0 1 1 1 0 0
$\vdots$	$\vdots$
$V_{\beta i}$	0 0 0 1 1
$\vdots$	$\vdots$
V_{2^N}	1 1 0 0 0 1
	0 0 0
	$\beta_i - \alpha_i + 1 = N C_i$

表 4

	$e_1^t, e_2^t, \dots, e_N^t$
v_1	1 1 1
v_2	0 0 1
$\vdots$	$\vdots$
v_{N+1}	1 0 0
$\vdots$	$\vdots$
$v_{\alpha i}$	1 1 1 0 0 1 1 1 0 0
$\vdots$	$\vdots$
$v_{\beta i}$	0 0 0 1 1
$\vdots$	$\vdots$
v_{2^N}	1 1 0 0 0 1
	0 0 0
	$\beta_i - \alpha_i + 1 = N C_i$

表 5

3.5 UL法における内部テスト端子数の下限

式(12)の対応関係から、UおよびLカットセット閉空間の包含関係をグラフに対応させると、つぎようになる。

$$S_U(i) \supseteq S_U(j) \Leftrightarrow V(v_1, f(e_i^t)) \supseteq V(v_1, f(e_j^t))$$

$$S_L(i) \supseteq S_L(j) \Leftrightarrow V(s(e_i^t), v_n) \supseteq V(s(e_j^t), v_n)$$

定理13によって $S_U(i)$ と $S_L(i)$ ($i=1, 2, \dots, N$) による素空間数が最大値 $(2^{N+1}-1)$ であるときは、それぞれUおよびLカットセット閉空間には包含関係がない。したがって上記の対応関係によって、

$$V(v_1, f(e_i^t)) \supset V(v_1, f(e_j^t)),$$

$$V(s(e_i^t), v_n) \supset V(s(e_j^t), v_n)$$

$$(i \neq j \quad i, j=1, 2, \dots, N)$$

となる。このことと定理8によれば、素空間数が最大値 $(2^{N+1}-1)$ のときのカットセット閉空間に対応するグラフにおいては、内部テスト端子 $\{e_1^t, e_2^t, \dots, e_N^t\}$ の間に到達関係がないことになる。すなわち、

$$f(e_i^t) \neq f(e_j^t) \quad s(e_i^t) \neq s(e_j^t) \quad (i \neq j, \quad i, j=1, 2, \dots, N)$$

である。したがって、 $f(e_i^t) \geq s(e_j^t)$ または $f(e_j^t) \geq s(e_i^t)$ は許される。以上の準備をもとにつぎの補題を与える。

補題5 UL法の内部テスト端子 $\{e_1^t, e_2^t, \dots, e_N^t\}$ の間に $s(e_i^t) \geq f(e_j^t)$ ($i \neq j$) なる関係をもつ i, j が存在しないならば、 $\{e_1^t, e_2^t, \dots, e_N^t\}$ による故障表はA型になり、存在するならばB型になる。逆も成り立つ。

<証明> 必要性：故障表がA型ならば、

$$V(v_1, f(e_i^t)) \supset v_p, \quad V(s(e_i^t), v_n) \supset v_p$$

$$\text{or } V(v_1, f(e_j^t)) \supset v_p, \quad V(s(e_i^t), v_n) \supset v_p$$

である。したがって、前者の場合、 $s(e_i^t) \neq v_p \geq f(e_j^t)$ 、であり、後者の場合 $s(e_i^t) \geq v_p \neq f(e_j^t)$ となる。ゆえに $s(e_i^t) \neq f(e_j^t)$ となる。

故障表がB型の場合は、

$$V(v_1, f(e_j^t)) \supset v_p, \quad V(s(e_i^t), v_n) \supset v_p$$

となるから、 $s(e_i^t) \geq f(e_j^t)$ となる。

十分性：対偶をとると上の証明において、A型とB型の立場が逆になり証明される。(証明終)

定理9と補題5を考慮して表6のように、組み合わせ的に $2N$ 次の行ベクトルを求めたものより $n \times 2N$ 次の行列をつくる。U(L) 法の場合と同様に表6を故障表としてもつ a-cyclic SEC グラフの存在を確認しなければならない。しかし、これは定理11によって存在が保証されている。したがって、このことから UL 法の

	$e_1^t, e_2^t, \dots, e_N^t$	$e_1^t, e_2^t, \dots, e_N^t$
v_1	1 1 1	0
$\vdots$	$\vdots$	
v_2	1 1 1	0
$\vdots$	$\vdots$	
v_{2^N}	0 0 0	0
$\vdots$	$\vdots$	
$v_{2^{N+1}}$	1 1 1	0
$\vdots$	$\vdots$	

表6 UL法の下限を与える故障表

場合の間Iの答が得られる。

定理15 n 個の節点をもつ可能なすべての a-cyclic SEC グラフ G において、テスト法ULによって1-識別可能にするための内部テスト端子数 N_{UL} はつぎのようになる

$$n \leq 2^{(N_{UL}+1)} - 1 \quad \dots\dots\dots (16)$$

すなわち、 $N_{UL} \geq [\log_2(n+1) - 1] \dots\dots\dots (17)$ (定理終)

定理15は W. Mayeda 氏らが文献[1]で推測した値に一致する。今までの議論は内部テスト端子を枝に設けた場合(枝レベル)[6]であるが、節点に端子を設けた場合(節点レベル)[6]を第5章で論じることにする。

また、内部テスト端子数の上限については定理16のようになる。

定理16 節点数 n の可能なすべての a-cyclic SEC グラフ G がテスト法 κ の内部テスト端子によって1-識別可能となるための端子数 N_κ は $(n-1)$ より多い必要はない。すなわち、

$$N_\kappa \leq (n-1), \quad \kappa = U, L, UL \quad (\text{証明略})$$

以上で、各テスト法に対する故障表の性質および内部テスト端子数の上、下限が明らかになった。

4. 最少内部テスト端子の決定アルゴリズムおよびデジタル・シミュレーション[12]

第2章において、a-cyclic SEC グラフで表現されるシステムのユニットが内部テスト端子によって1-識別

可能となるための必要十分条件を各テスト法について明らかにした。しかし、実際には1-識別可能となる最小数の内部テスト端子集合を見出すことが重要である。このような問題は被覆問題と呼ばれ、その解法として論理関数の単純化に用いられる prime-implicant 法が有効であることが知られている。〔4〕

ここでは、故障診断のデジタル・シミュレーションの立場から上述の事項をできるだけ簡単に実行できるように工夫している。

4.1 諸定義（その2）

ここでは、本章で用いる記号の定義を述べる。

$A \triangleq [a_{ij}] \in \mathbb{R}^{n \times n}$ 節点間接続行列 (adjacency matrix) :

$$a_{ij} \triangleq \begin{cases} 0 : v_i \text{ から } v_j \text{ に向う枝が存在しない場合} \\ 1 : v_i \text{ から } v_j \text{ に向う枝が存在する場合} \end{cases}$$

ただし、 $a_{ii}=0$ とする。

$A_0 \triangleq [a_{ij}^0] \in \mathbb{R}^{n \times n}$ ordered adjacency matrix : 節点の ordering を完了したグラフの A 行列であって、 A_0 は上三角行列 (ただし、 $a_{ii}^0=0$) となる。

$R \triangleq [r_{ij}] \in \mathbb{R}^{n \times n}$ 到達可能行列 (reachability matrix):

$$r_{ij} \triangleq \begin{cases} 0 : v_i \neq v_j \\ 1 : v_i \geq v_j, \text{ ただし、} r_{ii}=1 \text{ とする。} \end{cases}$$

節点の ordering を完了したグラフの R を $R_0 \triangleq [r_{ij}^0]$ と記すと、 R_0 は上三角行列 (ただし、 $r_{ii}^0=1$) となる。

$A_0^* \triangleq [a_{ij}^{0*}] \in \mathbb{R}^{n \times n}$: 節点の ordering を完了したグラフ G_d の節点間接続行列である。すなわち、 R_0 の $r_{ij}^0=0 (i < j)$ となるすべての要素に対して、節点 v_i と v_j の間に双方向枝である dummy edge を挿入して得られるグラフ G_d の節点間接続行列である。ここで、 G の vertex rang と G_d のそれと同じくするためにグラフ G と G_d の到達可能行列は同じであるとする。

$C_k \triangleq [c_{ij}]_k \in \mathbb{R}^{m \times \dim(\mathbf{E} \cup \mathbf{E}_d)}$ κ -DC $\langle e \rangle$ の被覆行列 (covering matrix) :

$$c_{ij} \triangleq \begin{cases} 0 : \kappa\text{-DC}\langle e_i \rangle \ni e_j \\ 1 : \kappa\text{-DC}\langle e_i \rangle \supset e_j \end{cases}$$

ただし、 $e_i \in \mathbf{E}$, $e_j \in (\mathbf{E} \cup \mathbf{E}_d)$, $m \triangleq \dim(\mathbf{E})$ であり、 C_k の行順および列順は第 i 行、第 i 列が同一の枝 $e_i (\in \mathbf{E})$ に対応するようにとられるものとする。また、 $j > m$ なる C_k の第 j 列には $e_d (\in \mathbf{E}_d)$ を対応させるものとする。ただし、 $\kappa = \text{U, L, UL}$ 。(以後、とくにことわらないときは同様の意味をもつとする。)

$C_k \triangleq [c_{ij}']_k \in \mathbb{R}^{m \times \dim[\mathbf{E}_0]}$: 第2章における定理

2, 4 に対応する unique ordering 可能なグラフの被覆行列であって、各要素の定義は C_k と同じである。

$C_k^{ab} = [c_{ij}^{ab}]_k \in \mathbb{R}^{\dim[\mathbf{E} - \mathbf{E}_k^{at}] \times \dim[\mathbf{E}' - (\mathbf{E} - \mathbf{E}_k^{at})]} : C_k$ からつぎのような行および列を除去して得られる行列である。すなわち、絶対内部テスト端子 $\mathbf{E}_k^{at}$ に対応する行および、 $\kappa\text{-DC}\langle e_i^{at} \rangle$ (ただし、 $e_i^{at} \in \mathbf{E}_k^{at}$) によって被覆される枝に対応する列である。ここに、 $\mathbf{E}'$ とは上記の操作によって除去されずに残った列に対応する枝集合である。

4.2 最少内部テスト端子の決定アルゴリズム

定義14 グラフ G が内部テスト端子集合 $\mathbf{E}_k^{at}$ によって1-識別可能であって、かつ、 $\dim[\mathbf{E}_k^{at}]$ が最小となる $\mathbf{E}_k^{at}$ を最小内部テスト端子集合と呼び、 $\mathbf{E}_k^{mt}$ で表わす。また、 $\mathbf{E}_k^{mt}$ の要素を e_k^{mt} で表わし、 $\mathbf{E}_k^{mt}$ の全集合を $\hat{\mathbf{E}}_k^{mt}$ で表わす。(定義終)

$\hat{\mathbf{E}}_k^{mt}$ の決定アルゴリズムの流れ図は、それぞれのテスト法とグラフ構造を考慮に入れると図3のようになる。以下では最も一般的な “non-unique ordering a-cyclic SEC グラフを UL 法によって故障診断するアルゴリズム” を示す。その他の場合には、図3の流れ図にしたがってこれの一部を実行すれば目的が達成される。

step 1. システムを a-cyclic SEC グラフ G に変換し、節点集合 $\mathbf{V}$ の番号付けを任意に行なう。そしてグラフ G を A 行列で表わす。

step 2. 節点集合 $\mathbf{V}$ の ordering を実行する。

step 3. A_0 行列および R_0 行列を求める。

step 4. $\mathbf{V}$ の ordering が non-unique であることを確認する。($a_{i,i+1}^0=1, i=1, 2, \dots, (n-1)$ でないこと、または、 R_0 が要素をすべて1とした上三角行列でないことを確認する。)

step 5. A_0^* 行列を求める。

step 6. テスト法が UL であることを確認する。

step 7. C_{UL} 行列を求め、 $\mathbf{E}_{UL}^{at}$ を決定する。

$\mathbf{E}_{UL}^{at} = \phi$ ならば $C_{UL} = C_{UL}^{ab}$ として step 9 へ移る。

step 8. C_{UL}^{ab} 行列を求める。

step 9. C_{UL}^{ab} のすべての列を被覆する最小数の行集合を求め $\mathbf{E}_{UL}^{mt'}$ とし、そのすべての組である $\hat{\mathbf{E}}_{UL}^{mt'}$ を求める。そして、 $\hat{\mathbf{E}}_{UL}^{mt} = \hat{\mathbf{E}}_{UL}^{mt'} \cup \mathbf{E}_{UL}^{at}$ とすれば最少内部テスト端子集合族が求まる。

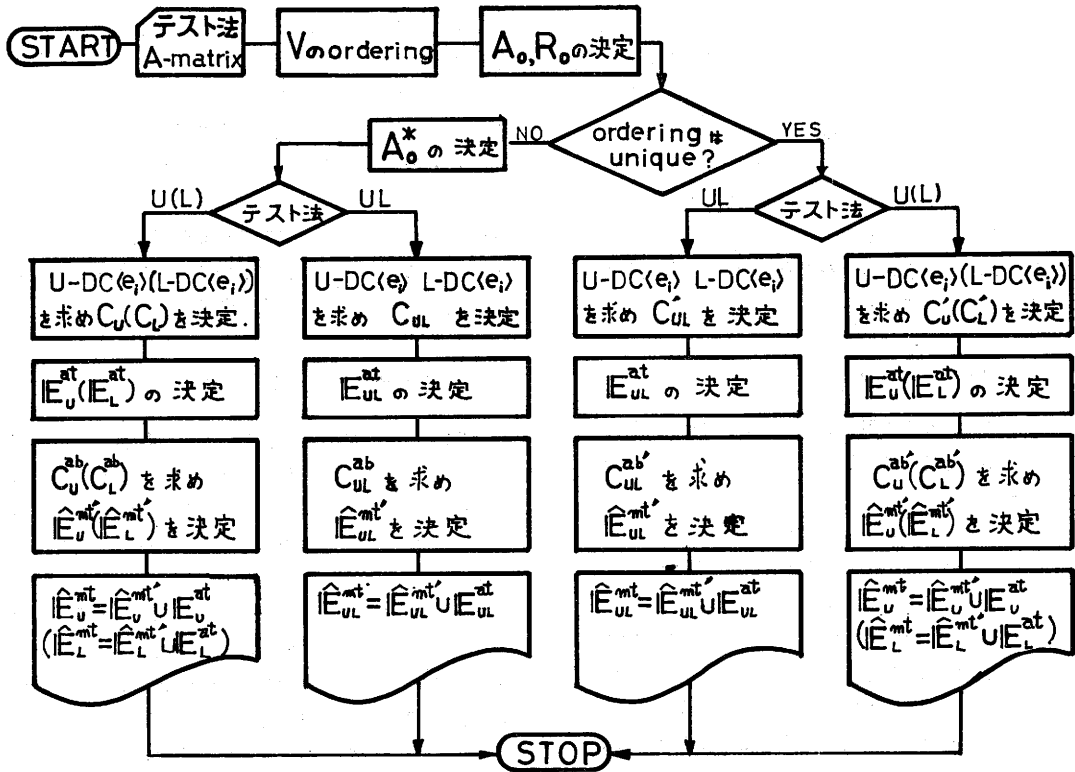


図3 最少内部テスト端子の決定アルゴリズム

4.3 $\hat{E}_k^{mt}$ 決定のデジタル・シミュレーション

上述のアルゴリズムにしたがって実際に作成したプログラムの概略を説明する。(付録にFORTRAN IVを用いたプログラムを掲載してある。)

4.3.1 入力

このプログラムの入力グラフのA行列とテスト法のみである。

4.3.2 A_0 行列の求め方

節点 $\hat{v}_i$ を ordering 完了後の order i の節点, $\hat{v}_i$ が入力時に有した節点番号を $ORD(\hat{v}_i)$ とする。

step 1. G を G' と表現し, $i=1$ に設定する。

step 2. order i となり得る節点集合 $\mathbf{v}_i$ は G' において $s^{-1}(v_i) = \phi$ ($v_i, v_j \in \mathbf{v}_i$) なる性質を有するから, このことを用いて $\mathbf{v}_i$ を求め, その中の1つの節点 $\hat{v}_j$ を v_i とする。したがって $ORD(\hat{v}_i) = v_j$ である。

step 3. $\hat{v}_i$ の出力枝 $f^{-1}(\hat{v}_i)$ を G' から除去して得られるグラフを新たに G' とする。

step 4. $i=n$ ならば, ordering が完了し, $ORD(\hat{v}_i)$, $i=1, 2, \dots, n$ が求まる。したがって, step 5 へゆく。 $i < n$ ならば, i を $i+1$ として, step 2 にもどる。

step 5. A行列の行と列の順序を $\{v_1, v_2, \dots, v_n\}$ から $\{ORD(\hat{v}_1), ORD(\hat{v}_2), \dots, ORD(\hat{v}_n)\}^*$ に入れ替えを行なうと A_0 行列となる。

4.3.3 R_0 行列の求め方

有向グラフのR行列は一般につきの式から求まる。
[5]

$$R = I + A + A^2 + A^3 + \dots + A^{n-1}$$

ここで, I は $(n \times n)$ 次の単位行列である。

しかし, この式を用いると n (節点数) が大きくなるほど計算時間が長くなり実用的でない。S. H. Unger氏は, 最近, 計算時間を短縮できるアルゴリズムを報告した[6]。本論文では先にA行列の ordering が完了: $ORD(\hat{v}_i)$ は $\hat{v}_i$ に対応している。 $ORD(\hat{v}_i)$ は入力時の情報を保存するため v_i を旧の節点番号で示したものである。

了されているために、S. H. Unger 氏のアルゴリズムを用いてもそのうちの pass II の手順だけで R_0 行列が求まり計算時間が $\frac{1}{2}$ に短縮される。

4.3.4 A_0^* 行列の求め方

A_0^* の定義から求め方は明らかであろう。ただし、 A_0^* 行列の要素には枝番号を入れるものとする。

4.3.5 U-DC<e> (L-DC<e>) の求め方

$e \in E, f(e) = v', s(e) = v''$ とする。

step 1. v' に reachable な節点 (v'' から reachable な節点) を R_0 から求め、これらの節点を $\{v_1, v_2, \dots, v_s\}$ とする。

step 2. A_0^* において $\{v_1, v_2, \dots, v_s\}$ に対応する行(列)に含まれ、かつ、 $\{v_1, v_2, \dots, v_s\}$ に対応する列(行)に含まれない枝集合を求める。この枝集合が U-DC<e> (L-DC<e>) である。

4.3.6 C_k の求め方

この前で求めた U-DC<e>, (L-DC<e>) を用いて C_k の定義にしたがって求める。そのとき、つぎの2つの補題が成り立つ。

補題 6 C_k の左端から m 個の列をとって得られる m 次正方行列の対角要素は非零である。

<証明>このことは C_k の定義から明らかである。
(証明終)

補題 7 C_{UL} および $C_U(C_L)$ の枝 $e_d (\in E_d)$ に対応する列にはそれぞれ必ず4個および2個以上の非零要素が含まれる。

<証明> C_U について証明する。入力節点 v_1 および出力節点 v_n はそれぞれ他のすべての節点と到達可能であるから v_1 および v_n につながる dummy edge e_d は存在しない。そこで、 e_d の両端点を v_i, v_j とすると、

$$f^{-1}(v_i) \ni e_i \ni \phi, f^{-1}(v_j) \ni e_j \ni \phi, e_i, e_j \in E, \\ i, j \neq 1, n$$

であるから、後述の定理17によって U-DC< e_i > $\ni e_d$, U-DC< e_j > $\ni e_d$ となり、 e_d に対応する C_U の列には2個以上の非零要素がある。 C_L についても同様に示される。 C_{UL} については、 C_U, C_L の2つから明らかである。(証明終)

4.3.7 絶対内部テスト端子集合 E_k^{at} の求め方

第2章における定理5, 定理6にしたがって求めればよいが、ここでは、つぎのようにして求めた。す

なわち、 C_k でただ1個の非零要素を有する列に対応する枝を求める*。その集合が E_k^{at} である。

4.3.8 C_k^{ab} 行列の求め方

C_k と E_k^{at} を用いて定義のごとく求める。

4.3.9 $\hat{E}_k^{mt}$ の求め方

$\hat{E}_k^{mt}$ を求める問題は被覆問題に帰着する。すなわち、 C_k^{ab} の行に対応する枝 $e_i (\in (E - E_k^{at}))$ にブール変数 x_i を割り当て、1つの列における非零要素の行に対応する枝に割り当てたブール変数の論理和をつくる。さらに、各列について上述のことを行なって得られる論理和のすべての論理積をとって論理関数 F を求める。そして、最後には F を prime implicant 法を用いて基礎項の和に変形して、その最小数の変数からなる基礎項に対応する枝を求めることである。しかしながら、(i) 上述のような論理演算による変形をシミュレーションすることは困難である、(ii) 故障診断においては、すべての基礎項を必要としない、という2つの理由によって、ここではつぎのように実行することを提案する。

U法およびL法においては $h = [\log_2 n] - \dim E_k^{at}$, UL法のときは $h = [\log_2(n+1) - 1] - \dim E_k^{at}$ と設定してつぎの手順を行なう**。

step 1. C_k^{ab} 行列の行に対応する枝から h 個とり出してそれを内部テスト端子候補 $E_k^{t'}$ とする。

step 2. $E_k^{t'}$ が C_k^{ab} 行列の列に対応する E' をすべて被覆するか否かを調べる。具体的には、 $E_k^{t'}$ に対応する行ベクトルの各列和をとり、その列和がすべて非零であるか否かを調べる。非零ならば $E_k^{mt'} = E_k^{t'}$ となる。

step 3. $E_k^{t'}$ の組み合わせを変えて step 2 を実行する。

step 4. step 2 で被覆されなければ step 5 へ、被覆されれば step 6 へ移る。

step 5. $h = h + 1$ として step 1 へもどる。

step 6. $E_k^{mt'}$ の全集合が $\hat{E}_k^{mt'}$ であり、最少内部テスト端子集合族 $\hat{E}_k^{mt}$ は $\hat{E}_k^{mt} = \hat{E}_k^{mt'} \cup E_k^{at}$ で求まる。

以上のデジタル・シミュレーションの演算時間に
*: これは定理5または定理6の絶対内部テスト端子の必要十分条件と等価な性質であるが、より簡単明瞭である。

** : これは内部テスト端子数の下限および絶対内部テスト端子の性質を利用して、組み合わせの数を減じていることに注意されたい。

ついてつぎのことがいえる。(i) 入力時のグラフの節点数が多い程演算時間が長い。(ii) U(L) 法の方がUL法よりも演算時間が長い。(iii) 絶対内部テスト端子が多い程演算時間が短い。

すなわち、このシミュレーションの演算時間を決定するのは C_k^{ab} の次数であることがいえよう。FACOM 270-20/30 による実施例によると $n=12$, $m=20$ のもとでのCPU時間は60秒前後、また、 $n=8$, $m=12$ のもとでのCPU時間は10秒前後となった。

4.4 デジタル・シミュレーションの例

デジタル・シミュレーションの例を図4のグラフに対するUL法について示す。

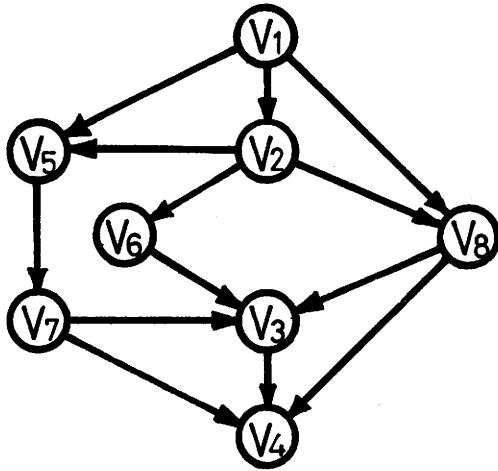


図4 例題のための a-cyclic SEC グラフ G

10 入力データとして任意の節点番号を付して得られる図4の A 行列と、テスト法ULを入れる。

$$A = \begin{matrix} & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \end{matrix} & \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \end{matrix}$$

20 A_0 行列を求める。節点番号をつぎのように変換すると上三角行列である A_0 行列が得られる。節点番号を変更した結果を図5に示す。

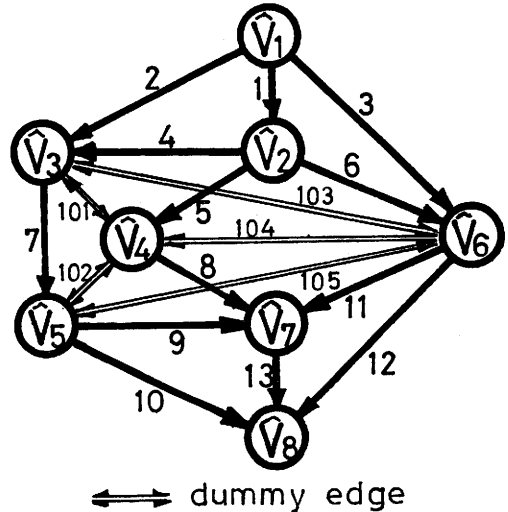


図5 A_0^* 行列に対応するグラフ

$$A_0 = \begin{matrix} & \hat{v}_1 & \hat{v}_2 & \hat{v}_3 & \hat{v}_4 & \hat{v}_5 & \hat{v}_6 & \hat{v}_7 & \hat{v}_8 \\ \begin{matrix} v_1 \rightarrow \hat{v}_1 \\ v_2 \rightarrow \hat{v}_2 \\ v_3 \rightarrow \hat{v}_7 \\ v_4 \rightarrow \hat{v}_8 \\ v_5 \rightarrow \hat{v}_3 \\ v_6 \rightarrow \hat{v}_4 \\ v_7 \rightarrow \hat{v}_5 \\ v_8 \rightarrow \hat{v}_6 \end{matrix} & \begin{pmatrix} \hat{v}_1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ \hat{v}_2 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ \hat{v}_3 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \hat{v}_4 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \hat{v}_5 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ \hat{v}_6 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ \hat{v}_7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hat{v}_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{matrix}$$

30 R_0 行列を求めるとつぎのようになる。

$$R_0 = \begin{matrix} & \hat{v}_1 & \hat{v}_2 & \hat{v}_3 & \hat{v}_4 & \hat{v}_5 & \hat{v}_6 & \hat{v}_7 & \hat{v}_8 \\ \begin{matrix} \hat{v}_1 \\ \hat{v}_2 \\ \hat{v}_3 \\ \hat{v}_4 \\ \hat{v}_5 \\ \hat{v}_6 \\ \hat{v}_7 \\ \hat{v}_8 \end{matrix} & \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{matrix}$$

40 A_0^* を求めるとつぎのようになる。ここで、dummy edge $\mathbf{E}_d$ と $\mathbf{E}$ とを区別するために dummy edge には bias として100番台の枝番号をわり当ててある。この A_0^* のグラフを図5に併記する。

$$A_0^* = \begin{matrix} & \hat{v}_1 & \hat{v}_2 & \hat{v}_3 & \hat{v}_4 & \hat{v}_5 & \hat{v}_6 & \hat{v}_7 & \hat{v}_8 \\ \begin{matrix} \hat{v}_1 \\ \hat{v}_2 \\ \hat{v}_3 \\ \hat{v}_4 \\ \hat{v}_5 \\ \hat{v}_6 \\ \hat{v}_7 \\ \hat{v}_8 \end{matrix} & \begin{pmatrix} 0 & 1 & 2 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 4 & 5 & 0 & 6 & 0 & 0 \\ 0 & 0 & 0 & 101 & 7 & 103 & 0 & 0 \\ 0 & 0 & 101 & 0 & 102 & 104 & 8 & 0 \\ 0 & 0 & 0 & 102 & 0 & 105 & 9 & 10 \\ 0 & 0 & 103 & 104 & 105 & 0 & 11 & 12 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 13 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{matrix}$$

5° 被覆行列 C_{UL} を求めるとつぎのようになる。

ここで $c_{ij}=2$ となる c_{ij} は枝 e_j が枝 e_i の上方および下方有向カットセットの両方に含まれていることを示す。

$$C_{UL} = \begin{matrix} & 1, & 2, & 3, & 4, & 5, & 6, & 7, & 8, & 9, & 10, & 11, & 12, & 13, & 101, & 102, & 103, & 104, & 105 \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \end{matrix} & \begin{pmatrix} 2 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 2 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 2 & 1 & 2 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 2 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 2 & 1 & 1 & 2 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 2 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 2 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 2 & 2 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 2 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 2 & 2 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{matrix}$$

6° 絶対内部テスト端子を C_{UL} から求めると $E_{UL}^{at} = \{7\}$ となる。

7° E_{UL}^{at} を用いて, C_{UL} から C_{UL}^{ab} を求めるとつぎのようになる。すなわち, $E_{UL}^{at} = \{7\}$ に対応する C_{UL} の行 $\{7\}$ および $E_{UL}^{at} = \{7\}$ の上方または下方有向カットセットに包含される枝に対応する C_{UL} の列 $\{3, 5, 6, 7, 8, 11, 12, 101, 102, 103, 105\}$ を除去して得られる行列が C_{UL}^{ab} である。

$$C_{UL}^{ab} = \begin{matrix} & 1 & 2 & 4 & 9 & 10 & 13 & 104 \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \end{matrix} & \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix} \end{matrix}$$

8° C_{UL}^{ab} から $\hat{E}_{UL}^{mt'}$ を求めると, $\hat{E}_{UL}^{mt'} = \{(3, 12)\}$ となり最少内部テスト端子集合族 $\hat{E}_{UL}^{mt'}$ は $\{(3, 7, 12)\}$ の1組だけとなる。

以上に述べた最少内部テスト端子決定アルゴリズムおよびそのデジタル・シミュレーションの特徴を列挙すればつぎのごとくなる。(i) 入力データおよびその前処理が簡単である, (ii) 節点の **ordering** を実行することにより, 途中の行列に関する諸処理に必要な手順および時間を節約している, (iii) **dummy edge** の挿入によって, すべてのグラフを一様な手法で処理できる, (iv) 絶対内部テスト端子および内部テスト端子数の下限を考慮に入れて, 内部テスト端子候補の組み合わせ数の減少を図っている。(v) 被覆問題を行列の一部の行ベクトルに対する列和をとる演算におきかえて行なっているためにプライム・インプリカントそのものを行なうよりも簡単である。

さらに, シミュレーションに用いる言語をグラフ的構造の表現が容易な言語[17]を用いるとか, または, データの性質を考慮に入れてビット処理を行えば, さらに記憶容量および CPU 時間を減少させることが可能であると考えられる。

5. 内部テスト端子を枝および節点としたときの相互関係^[13]

いままで, 内部テスト端子を枝に設定した場合について議論してきた。本章では内部テスト端子を節点に設定した場合[8][9][13]と枝に設定した場合との相互関係および, 節点に設定した場合の絶対内部テスト端子の質性などを明らかにする。

5.1 節点レベルのテスト法

内部テスト端子を枝および節点に設定した場合をそれぞれ枝レベル, 節点レベルと呼ぶことにする。

故障の検出は枝レベルの場合と同様である。また, 故障節点の **location** 法は枝レベルに対応してつぎの3つの場合が考えられる。

U法 入力節点 v_1 にテスト信号を入れて, 任意の節点 v_i の出力を観測し, その出力が規定出力ならば $\overline{V(v_1, v_i)}$ に故障節点が存在し, 規定出力でないならば $V(v_1, v_i)$ に存在する。

L法 任意の節点 v_i にテスト信号を入れて出力節点 v_n からの出力を観測したとき, それが規定出力ならば故障節点は $\overline{V(v_i, v_n)}$ に存在し, 規定出力でないならば, $V(v_i, v_n)$ に存在する。

UL法 任意の節点 v_i に対してつぎの3つの場合が

生じる。

(i) 入力節点 v_1 にテスト信号を入れて、節点 v_i からの出力を観測したとき、その出力が規定出力でないならば、故障節点は $V(v_1, v_i)$ に存在する。

(ii) 節点 v_i にテスト信号を入れて、出力節点 v_n からの出力を観測したとき、その出力が規定出力でないならば、故障節点は $V(v_i, v_n)$ 内に存在する。

(iii) (i) および (ii) の結果、出力がとも規定出力ならば故障節点は $\{V(v_1, v_i) \cap V(v_i, v_n)\}$ に存在する。

以上の各 location 法に用いられる節点 v_i を内部テスト端子 (internal test vertex) と呼び、それらの集合をそれぞれのテスト法に対応して $\mathbf{V}_{U^t}$, $\mathbf{V}_{L^t}$, $\mathbf{V}_{UL^t}$ と表わす。

5.2 枝レベルと節点レベルの内部テスト端子の対応関係

文献[8]において、つぎの定理が報告されている。

定理17 グラフ G において、 $f^{-1}(v) = \{e_1, e_2, \dots, e_p\}$ とすれば、つぎの関係が成立する、

$$U\text{-}DC\langle e_i \rangle \supseteq f^{-1}(v),$$

$$U\text{-}DC(v) \triangleq U\text{-}DC\langle e_1 \rangle = U\text{-}DC\langle e_2 \rangle = \dots$$

$$\dots = U\text{-}DC\langle e_p \rangle$$

また、 $s^{-1}(v) = \{e_1, e_2, \dots, e_p\}$ とすれば

$$L\text{-}DC\langle e_i \rangle \supseteq s^{-1}(v),$$

$$L\text{-}DC(v) \triangleq L\text{-}DC\langle e_1 \rangle = L\text{-}DC\langle e_2 \rangle = \dots$$

$$\dots = L\text{-}DC\langle e_p \rangle$$

ただし、 $i=1, 2, \dots, p$ (定理終)

5.2.1 U法およびL法の内部テスト端子の対応関係

定理17から表7に示す内部テスト端子の対応表を得る*。

テスト法	枝レベルの表現	節点レベルの表現
U 法	$E_U^t \xrightarrow{\quad} f(E_U^t) \equiv \sum f(e_i^t)$ $\sum_{00}[f(v_i^t)] \xleftarrow{\quad} \sum v_i^t \equiv \sum v_i^t$	
L 法	$E_L^t \xrightarrow{\quad} s(E_L^t) \equiv \sum s(e_i^t)$ $\sum_{00}[s(v_i^t)] \xleftarrow{\quad} \sum v_i^t \equiv \sum v_i^t$	

→ unique transformation

← non-unique transformation

表7 U(L)法における枝レベルと節点レベルの対応関係

この表7の対応によってつぎのことがわかる。

Remark 1 U法およびL法において、故障診断の

*: 表7および8における $00[A]$ は A 集合の要素から1つだけを任意に選定することを意味するものとする。

ための内部テスト端子数および診断のためのデータ測定回数は枝レベルにおける場合と節点レベルにおける場合とは同一である。

5.2.2 UL法における内部テスト端子の対応

定理17から表8に示す対応表を得る。

テスト法	枝レベルの表現	節点レベルの表現
枝レベルの UL 法	$E_{UL}^t \xrightarrow{\quad} f(E_{UL}^t) = V_{UO}^t$ $s(E_{UL}^t) = V_{LO}^t$	
節点レベルの UL 法	$E_{UO}^t = \sum_{00}[f(v_{UL}^t)]$ $E_{LO}^t = \sum_{00}[s(v_{UL}^t)]$	$\mathbf{V}_{UL}^t = \sum v_{UL}^t$

→ unique transformation

← non-unique transformation

表8 UL法における枝レベルと節点レベルの対応関係
 $00[A]$: 集合 A の1つの要素 (Only-One element) のみをとる

$\mathbf{V}_{kO}$: κ 法のみを用いる節点レベルの内部テスト端子

$\mathbf{E}_{kO}$: κ 法のみを用いる枝レベルの内部テスト端子

したがって、節点レベルにおける内部テスト端子の必要十分条件は、第2章の定理1, 3を表7および8によって変換すれば得られる。その結果、つぎの定理を得る。

定理18 節点レベルにおける内部テスト端子 $\mathbf{V}_{U^t}$, $\mathbf{V}_{L^t}$, および $\mathbf{V}_{UL^t}$ によって a-cyclic SEC グラフ G が1-識別可能となるための必要十分条件はそれぞれ次式が成立することである。

$$U\text{法}: \bigcup_{v \in \mathbf{V}_{U^t}} \{U\text{-}DC\langle v \rangle\} = E \cup E_d$$

$$L\text{法}: \bigcup_{v \in \mathbf{V}_{L^t}} \{L\text{-}DC\langle v \rangle\} = E \cup E_d$$

$$UL\text{法}: \bigcup_{v \in \mathbf{V}_{UL^t}} \{U\text{-}DC\langle v \rangle \cup L\text{-}DC\langle v \rangle\} = E \cup E_d$$

(証明略)

また、定理2.4に対応する unique ordering な a-cyclic SEC グラフに対する結果も同様にして得られる。

さらに、表8によってつぎの2つのことがわかる。

Remark 2 UL法における枝レベルと節点レベルの内部テスト端子をそれぞれ節点レベルおよび枝レベルに変換すると、内部テスト端子の機能がU法およびL法に分かれ、個々のレベルにおけるUL法本来の意味をもたなくなる。ゆえに、この変換には上述の意味で注意を要する。

Remark 3 unique ordering な a-cyclic SEC グラフがUL法の意味で1-識別可能となるための内部テ

スト端子の決定法についてつぎのことがいえる。節点レベルについては節点の **ordering** にしたがって内部テスト端子が決定される*。

しかし、枝レベルにおいては、一般にバイパスの枝が存在するために節点レベルよりも内部テスト端子決定の手順が複雑になる。

5.3 節点レベルにおける絶対内部テスト端子の性質および内部テスト端子数の上, 下限

5.3.1 節点レベルの絶対内部テスト端子の性質

定理5に対応して、節点レベルの絶対内部テスト端子の性質はつぎの定理19で与えられる。その前に補題を与える。

補題 8 $e \in U\text{-}DC\langle v_0 \rangle$ であることと、 $v_0 \in \{R^{-1}(f(e)) - R^{-1}(s(e))\}$ は等価である。 $(e \in L\text{-}DC\langle v_0 \rangle$ であることと $v_0 \in \{R(s(e)) - R(f(e))\}$ は等価である。)

＜証明＞ 下方有向カットセットに対しても同様に示すので以下には、上方有向カットセットに対する証明だけを行なう。 $e \in U\text{-}DC\langle v_0 \rangle \Rightarrow f^{-1}(R(v_0)) - s^{-1}(R(v_0))$ であり、

$$f^{-1}(R(v_0)) \supset s^{-1}(R(v_0))$$

であることから、 $e \in f^{-1}(R(v_0))$ かつ $e \notin s^{-1}(R(v_0))$ が成立する。ゆえに、 $v_0 \in R^{-1}(f(e))$ かつ $v_0 \notin R^{-1}(s(e))$ となって

$$v_0 \in \{R^{-1}(f(e)) - R^{-1}(s(e))\}$$

なる結論を得る。

この逆は上記の証明の逆をたどれば明らかである。

(証明終)

定理 19 つぎの3つの事柄は等価である。

(i) 節点 v_0 は $U(L)$ 法による絶対内部テスト端子である。ただし、 $v_0 \in \mathbf{V}^t(\mathbf{V}_L^t)$ とする。

(ii) $\dim [R^{-1}(f(e)) - R^{-1}(s(e))] = 1$ となる枝 e が $f^{-1}(v_0)$ に存在する。 $(\dim [R(s(e)) - R(f(e))] = 1$ となる枝 e が $s^{-1}(v_0)$ に存在する。)

(iii) $R^{-1}(s(e)) \supseteq s(f^{-1}(f(e)))$ となる枝 e が $f^{-1}(v_0)$ に存在する。 $(R(f(e)) \supseteq f(s^{-1}(s(e)))$ となる枝 e が $s^{-1}(v_0)$ に存在する。)

＜証明＞ U 法のみについて証明する。(図6参照)

(i) $\Leftrightarrow$ (ii) 補題8より明らかである。

*: 節点レベルの U 法においては出力節点を除くすべての節点、 L 法においては入力節点を除くすべての節点、 UL 法の場合は全節点数が奇数のときは **ordering** が偶数番目の節点に、全節点数が偶数のときは **ordering** が奇数または偶数番目の節点が内部テスト端子に定まる。

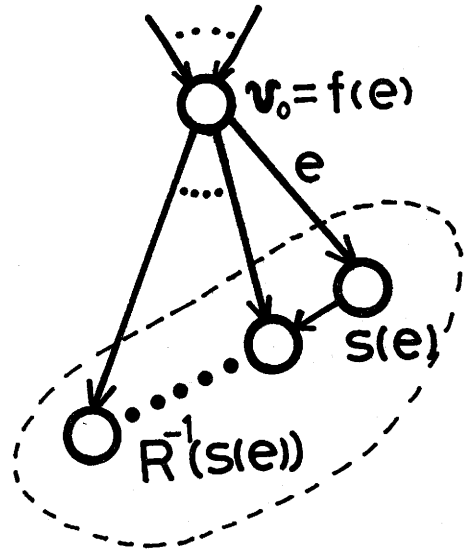


図6 節点レベルの U 法における絶対内部テスト端子の説明図

(ii) $\rightarrow$ (iii) $R^{-1}(f(e)) - R^{-1}(s(e)) = v_0$ ならば $v_0 = f(e)$ であるから

$$s(f^{-1}(v_0)) \subseteq R^{-1}(s(e))$$

である。(iii) $\rightarrow$ (ii) は上記の逆をたどれば明らかである。(証明終)

定理 20 節点レベルの UL 法において、絶対内部テスト端子は存在しない。

＜証明＞ $f(e)$ の上方有向カットセットを考えれば、

$$E\{V(v_1, f(e)) \times \overline{V(v_1, f(e))}\} \ni e$$

となり、 $s(e)$ の下方有向カットセットを考えれば、

$$E\{\overline{V(s(e), v_n)} \times V(s(e), v_n)\} \ni e$$

となり、枝 e をカットセットに含む節点は必ず2個以上存在する。したがって、節点レベルの UL 法には絶対内部テスト端子は存在しない。(証明終)

5.3.2 節点レベルにおける内部テスト端子数の上限および下限。

枝レベルと節点レベルの内部テスト端子数の上限および下限を付表1に示す。

以上にのべたように、枝レベルと節点レベルの相互関係の基本は表7および表8であるが、枝レベルによって得られた結果を節点レベルへ変換する方法は各テスト法ともに一意的であるが、その逆は一般にそうではない。本論文で枝レベルを中心にした議論を行なったのは主としてこの理由による。

6. むすび

本論文では、システム内部にテスト用端子を導入した故障診断法の基本的な性質を検討するために、システムを a-cyclic SEC グラフでモデル化して、グラフ理論的検討を行なった。

a-cyclic SEC グラフ G でモデル化した単一故障を有するシステムにおいて、可能なるすべてのテスト法 U, L, UL 法のそれぞれについて、(i) dummy edge なる概念を導入して a-cyclic SEC グラフ G を 1-識別可能にするための内部テスト端子の必要十分条件を明らかにし、(ii) また、絶対内部テスト端子であるための必要十分条件をも明らかにした。(iii) 故障表の必要十分条件および a-cyclic SEC グラフに対応するカットセット閉空間の性質を明らかにした。さらに、これらを用いて各テスト法における内部テスト端子数の下限を与えた。そして、これらの事項をもとにした最少内部テスト端子決定の有効なアルゴリズムを示した。

また、上記の下限を与えるときの内部テスト端子と節点の間の到達関係を与える故障表が明らかにされた。これらのことは、本故障診断法を最も有効ならしめるグラフ構造およびその生成法を知るための第一段階となり[14]～[16]、故障診断を考慮した論理回路の設計に役立つと思われる。最後に、枝レベル（内部テスト端子を枝に設定した場合）と節点レベル（内部テスト端子を節点に設定した場合）の故障診断の相互関係の詳細を示した。

今後の問題点としてつぎの3つがあげられる。

(i) 本故障診断法を最も有効ならしめるグラフの生成法[14]～[16]、

(ii) 本故障診断法における仮定をみだし、かつ、上述のグラフ構造をもった任意の論理回路を設計できるような節点（すなわち、ユニット）の系統的研究。

(iii) UL 法の B 型故障表の必要十分条件

本論文でのべたグラフ理論的なとらえかたは、この種の故障診断問題のすべての答を与えてくれるものではないが、基本的な性質および問題点を的確にとらえることを可能ならしめるものである。したがって、これから得られた結果をいかした前記(ii)の問題の解決がとくに望まれる。

最後に、本論文に関し、ご指導いただいた阪大工学部中西義郎教授にお礼申し上げます。また、有益な討論をいただいた阪大工学部中野氏、ならびに本学学生山本君（現在 日立製作所勤務）、北嶋君（現在 沖電気株式会社勤務）に感謝する。

参考文献

- 1) W. Mayeda and C.V. Ramamoorthy: "Distinguishability Criteria in Oriented Graphs and Their Application to Computer Diagnosis-I", IEEE, Trans. on Circuit Theory, Vol. CT-16, No. 4, pp. 448～454(1969)
- 2) C. V. Ramamoorthy and W. Mayeda: "Computer Diagnosis Using the Blocking Gate Approach", IEEE, Trans. on Computer Vol. C-20, No 11, pp. 1294～1299 (Nov., 1971)
W. Mayeda and C.V. Ramamoorthy, "Distinguishability Criteria in Oriented Graph and its Application to Computer Diagnosis-II", Proc. of IEEE International Symposium on Circuit Theory, pp. 38～39, Georgia, 1970.
- 3) W. Mayeda: "Graph and System Diagnosis", 電子通信学会, 回路とシステム理論研究会資料, CT 72-26 (1972-06)
- 4) H. Y. Chang, E. Manning and G. Metz: "Fault Diagnosis of Digital Systems", Jone Willy & Sons, Inc. (1970)
- 5) F. Harary, R. Z. Norman and D. Cartwright: "Structural Models: An Introduction to the Theory of Directed Graphs", John Wiley & Sons, Inc. (1965)
- 6) S. H. Unger: "An Algorithm for Finding the Reachability Matrix of a Directed Linear Graph", IEEE Trans. on Circuit Theory, Vol. CT-16, No. 1, pp. 130～132 (1969-02)
- 7) H. Y. Chang: "An Algorithm for Selecting an Optimum Set of Diagnostic Tests," IEEE Trans. on Electronic Computer, Vol. EC-14, No. 5, pp. 706～711 (1965-10)
- 8) 中野, 中西: "システム故障診断のための内部テスト端子決定法", 信学論(D) Vol. 54-C, No. 8 pp. 774～750 (1971-08)
- 9) 中野, 中西: "システム故障診断における1-識別可能のための必要十分条件", 信学論 (D) Vol. 55-D, No. 10, pp. 645～659(1972-10)
- 10) 松本, 飛田他: "システムの最少内部テスト端子決定に関する考察(I) (システムの故障診断に関する基本定理)", 昭和47年度電気四学会北陸支部連合大会論文集, D-15 (1972-10) または, 北嶋, 飛田: "システム故障診断に関する研究", 福井

- 大学工学部電気工学科卒業研究論文，昭和47年3月10日
- 11) 松本，飛田他；“システムの最少内部テスト端子決定に関する考察(Ⅱ)（内部テスト端子数の上限，下限）”，昭和47年度電気四学会北陸支部連合大会論文集，D-16（1972-10）
 - 12) 松本，飛田他；“システムの最小内部テスト端子決定に関する考察(Ⅲ)（最少内部テスト端子決定のデジタル・シミュレーション）”，昭和47年度電気四学会北陸支部連合大会論文集，D-17(1972-10)
 - 13) 松本，飛田他；“システムの最少内部テスト端子決定に関する考察(Ⅳ)（内部テスト端子を枝および節点としたときの相互関係）”，昭和47年度電気四学会北陸支部連合大会論文集，D-18（1972-10）
 - 14) 松本，飛田： “システム故障診断における内部テスト端子数の下限をみたすグラフの構造および生成法”，電子通信学会，回路とシステム理論研究会資料 CT73-5（1973-04）
 - 15) 松本，飛田，中西： “最少内部テスト端子をもつ a-cyclic SEC グラフの生成法”，昭和48年度電気学会全国大会論文集，1289（1973-04）
 - 16) 松本，飛田，中西： “最大情報収集能力をもつシステムの構造およびその構成法について”，昭和48年度電子通信学会全国大会論文集，1481(1973-03)
 - 17) たとえば，伊理： “グラフの構造を有する情報の処理技法について”，昭和47年度電気四学会連合大会予稿集，217，pp. 850～853

付録1 FORTRAN IV による最少内部テスト端子の決定プログラム

入力データの記号の注釈

N : a-cyclic SEC グラフGの節点数

METHOD : テスト法, U法→1 L法→2 UL法→3

BIAS1 : dummy edge の枝番号を实在の枝番号と区別するための数，例においては100としてある。

A : グラフGの adjacency matrix

```

C      ****DETERMINATION OF THE MINIMUM INTERNAL TEST EDGES****
      INTEGER A(12, 12), A1(12, 12), R(12, 12), B(20), C(20), UDC(50), DC(50), COV
      *(20, 50), NUM(40), LSUM(50), AABS(20), ABSROW(50), COVER(20, 50), ORD(12),
      *NO(40), BIAS1, BIAS2
      READ(5, 1000) METHOD, N, BIAS1
1000  FORMAT (I3/I3/I3)
      DO 1502 MO1=1, N
1502  READ(5, 1500) (A(MO1, MO2), MO2=1, N)
1500  FORMAT(25I3)
      DO 550 I=1, N
550   AABS(I)=I
      WRITE(6, 1504)
1504  FORMAT(1H1,///5X,23H INPUT ADJACENCY MATRIX)
      CALL MATRIX(A, 12, 12, N, N, AABS, N, AABS, N)
C      ****DECISION OF ORDERED ADJACENCY MATRIX****
      DO 868 MO1=1, N
      DO 868 MO2=1, N
868   R(MO1, MO2)=A(MO1, MO2)
      MB1=0
876   MB2=0
      DO 870 MB3=1, N
      MCOL=0
      DO 872 MB4=1, N
872   MCOL=MCOL+R(MB4, MB3)
      IF(MCOL. EQ. 0) GO TO 874
870   CONTINUE
874   MB1=MB1+1
      ORD(MB1)=MB3
      DO 878 MB5=1, N
878   R(MB5, MB3)=1
      DO 880 MB5=1, N

```

```

880 R(MB3, MB5)=0
      IF(MB1. NE. N) GO TO 876
C     **** CHANGING OF COLUMN AND ROW ****
      DO 882 MB5=1, N
        MB6=ORD(MB5)
        DO 882 MB7=1, N
982    R(MB7, MB5)=A(MB7, MB6)
        DO 884 MB5=1, N
          MB6=ORD(MB5)
          DO 884 MB7=1, N
984    A1(MB5, MB7)=R(MB6, MB7)
        WRITE(6, 1013)
1013  FORMAT(///5X, 22H ** VERTEX ORDERING **)
        WRITE(6, 5000) (MB5, MB5=1, N)
5000  FORMAT(5X, 25I3)
        WRITE(6, 5000) (ORD(MB5), MB5=1, N)
        WRITE(6, 1014)
1014  FORMAT(///5X, 31H ** ORDERED ADJACENCY MATRIX **)
        CALL MATRIX(A1, 12, 12, N, N, AABS, N, AABS, N)
        DO 850 MB1=1, N
          DO 850 MB2=1, N
850    R(MB1, MB2)=A1(MB1, MB2)
        MB1=N
852    MB1=MB1-1
        DO 856 MB2=MB1, N
          IF(R(MB1, MB2). EQ. 0) GO TO 856
          DO 857 MB3=MB2, N
857    R(MB1, MB3)=R(MB1, MB3)+R(MB2, MB3)
856    CONTINUE
        IF(MB1. NE. 1) GO TO 852
        DO 858 MA1=1, N
858    R(MA1, MA1)=1
        DO 860 MA1=1, N
          DO 860 MA2=1, N
            IF(R(MA1, MA2). EQ. 0) GO TO 860
            R(MA1, MA2)=1
860    CONTINUE
        WRITE(6, 1010)
1010  FORMAT(///5X, 26H ** REACHABILITY MATRIX **)
        CALL MATRIX(R, 12, 12, N, N, AABS, N, AABS, N)
        DO 702 MA1=1, N
          DO 702 MA2=1, N
702    A(MA1, MA2)=0
        MEDGE=0
        DO 17 MA1=1, N
          DO 17 MA2=1, N
            IF(A1(MA1, MA2). EQ. 0) GO TO 17
            MEDGE=MEDGE+1
            A(MA1, MA2)=MEDGE
17    CONTINUE
        DO 14 MA1=1, N
          DO 14 MA2=1, N
14    A1(MA1, MA2)=A(MA1, MA2)
        BIAS2=BIAS1
        NN=N-1
        DO 15 MA1=3, NN

```

```

      MA2=MA1
      DO 15 MA3=2, MA2
      IF(R(MA3, MA1). NE. 0) GO TO 15
      BIAS1=BIAS1+1
      A1(MA3, MA1)=BIAS1
      A1(MA1, MA3)=BIAS1
15  CONTINUE
      N1=MEDGE+BIAS1-BIAS2
      WRITE(6, 1007)
1007  FORM(////5X, 32H ** MODIFIED ADJACENCY MATRIX **)
      CALL MATRIX(A1, 12, 12, N, N, AABS, N, AABS, N)
      DO 16 MA1=1, N
      DO 16 MA2=1, N1
16  COV(MA1, MA2)=0
      IF(METHOD. EQ. 2) GO TO 125
C     **** THE EDGES COVERED BY UDC(EDGE)****
      I=1
10  J=1
20  IF(A(I, J). EQ. 0) GO TO 30
      MB1=A(I, J)
      K1=1
      K2=0
60  IF(R(K1, I). EQ. 0) GO TO 70
      K2=K2+1
      B(K2)=K1
70  K1=K1+1
      IF(K1. LE. N) GO TO 60
      K3=1
      MA1=0
80  K4=B(K3)
      K5=1
90  IF(A1(K4, K5). EQ. 0) GO TO 100
      K6=1
110 IF(K5. EQ. B(K6)) GO TO 100
      K6=K6+1
      IF(K6. LE. K2) GO TO 110
      MA1=MA1+1
      UDC(MA1)=A1(K4, K5)
      MB2=A1(K4, K5)
      IF(MB2. LE. MEDGE) GO TO 105
      MB2=MB2-BIAS2+MEDGE
105 COV(MB1, MB2)=COV(MB1, MB2)+1
100 K5=K5+1
      IF(K5. LE. N) GO TO 90
      K3=K3+1
      IF(K3. LE. K2) GO TO 80
      IF(METHOD. EQ. 1) GO TO 30
C     **** THE EDGES COVERED BY LDC(EDGE) ****
125 M1=1
      M2=0
120 IF(R(J, M1). EQ. 0) GO TO 130
      M2=M2+1
      C(M2)=M1
130 M1=M1+1
      IF(M1. LE. N) GO TO 120
      M3=1

```

```

      MA2=0
140  M4=C(M3)
      M5=1
150  IF(A1(M5, M4). EQ. 0) GO TO 160
      M6=1
170  IF(M5. EQ. C(M6)) GO TO 160
      M6=M6+1
      IF(M6. LE. M2) GO TO 170
      MA2=MA2+1
      DC(MA2)=A1(M5, M4)
      MB2=DC(MA2)
      IF(MB2. LE. MEDGE) GO TO 155
      MB2=MB2-BIAS2+MEDGE
155  COV(MB1, MB2)=COV(MB1, MB2)+1
160  M5=M5+1
      IF(M5. LE. N) GO TO 150
      M3=M3+1
      IF(M3. LE. M2) GO TO 140
30   J=J+1
      IF(J. LE. N) GO TO 20
      I=I+1
      IF(I. LE. N) GO TO 10
      WRITE(6, 4004)
4004  FORMAT(1H1, ///5X, 22H ** COVERING MATRIX **)
      DO 32 MA1=1, N1
32    ABSROW(MA1)=MA1
      CALL MATRIX(COV, 20, 50, MEDGE, N1, ABSROW, N1, ABSROW, N1)
      DO 34 MA1=1, MECGE
      DO 34 MA2=1, N1
      IF(COV(MA1, MA2). EQ. 0) GO TO 34
      COV(MA1, MA2)=1
34    CONTINUE
C     **** PROGRAM OF REDUCED COVERING MATRIX ****
      MABS=0
      DO 902 MA1=1, MEDGE
      MABST=0
      DO 901 MA2=1, MEDGE
901    MABST=MABST+COV(MA2, MA1)
      IF(MABST. NE. 1) GO TO 904
      MABS=MABS+1
      AABS(MA1)=1
      GO TO 902
904    AABS(MA1)=0
902    CONTINUE
      IF(MABS. NE. 0) GO TO 950
      JEDGE=MEDGE
      N2=N1
      DO 910 MA1=1, JEDGE
910    AABS(MA1)=MA1
      WRITE(6, 2002)
2002  FORMAT(///32H ** THERE IS NO ABSOLUTE EDGE **)
      GO TO 955
950    DO 906 MA1=1, N1
906    ABSROW(MA1)=0
      DO 908 MA1=1, MEDGE
      IF(AABS(MA1). EQ. 0) GO TO 908

```

```

DO 908 MA2=1, N1
ABSROW(MA2)=ABSROW(MA2)+COV(MA1, MA2)
908 CONTINUE
JINDEX=1
DO 907 MA1=1, N1
JINDEX=JINDEX * ABSROW(MA1)
IF(JINDEX. EQ. 0) GO TO 909
907 JINDEX=1
GO TO 913
909 JEDGE=0
DO 912 MB1=1, MEDGE
IF(AABS(MB1). NE. 0) GO TO 912
JEDGE=JEDGE+1
N2=0
DO 912 MB2=1, N1
IF(ABSROW(MB2). NE. 0) GO TO 912
N2=N2+1
COVER(JEDGE, N2)=COV(MB1, MB2)
912 CONTINUE
913 WRITE(6, 7000) (MB2, MB2=1, MEDGE)
7000 FORMAT(1H1, ///21H ** ABSOLUTE EDGES **,/(/25I3))
WRITE(6, 7002) (AABS(MA1), MA1=1, MEDGE)
7002 FORMAT(/(25I3))
IF(JINDEX. EQ. 0) GO TO 915
WRITE(6, 7001)
7001 FORMAT(/39H ** COVERING END WITH ABSOLUTE EDGES **)
GO TO 900
915 MA1=0
DO 914 MA2=1, MEDGE
IF(AABS(MA2). NE. 0) GO TO 914
MA1=MA1+1
AABS(MA1)=MA2
914 CONTINUE
MA1=0
DO 916 MA2=1, N1
IF(ABSROW(MA2). NE. 0) GO TO 916
MA1=MA1+1
ABSROW(MA1)=MA2
916 CONTINUE
WRITE(6, 7004)
7004 FORMAT(///32H ** DECREASED COVERING MATRIX ** )
CALL MATRIX(COVER, 20, 50, JEDGE, N2, AABS, JEDGE, ABSROW, N2)
C **** PROGRAM OF MINIMUM EDGES ****
955 GO TO (956, 957, 958), METHOD
956 WRITE(6, 6008) (AABS(K1), K1=1, JEDGE)
6008 FORMAT(1H1, ///45H ** MINIMUM EDGES OF U* INTERNAL TEST EDGES ** //(
*5X, 25I3))
GO TO 959
957 WRITE(6, 6009) (AABS(K1), K1=1, JEDGE)
6009 FORMAT(1H1, ///45H ** MINIMUM EDGES OF L* INTERNAL TEST EDGES ** //(
*5X, 25I3))
GO TO 959
958 WRITE(6, 6010) (AABS(K1), K1=1, JEDGE)
6010 FORMAT(1H1, ///46H ** MINIMUM EDGES OF UL* INTERNAL TEST EDGES **/(
*/5X, 25I3))
C **** (NKOSU)=(NUMBER OR INTERNAL TEST TERMINALS) ****

```

```

C      **** NO(1), NO(2),.....,NO(J),.....
C      WHERE *NO(J)* IS THE J-TH INTERNAL TEST EDGE ****
959  INDEX=0
      IF(METHOD. EQ. 3) GO TO 660
      SHO=ALOG (SHO)/ALOG (2)
      SHE=AINT (SHO)
      IF(SHO-SHE) 652, 652, 654
652  SHE=SHE-1. 0
654  NKOSU=SHE
      GO TO 666
660  SHO=ALOG(N+1)/ALOG(2)
      SHE=AINT(SHO)
      IF(SHO-SHE) 662, 662, 664
662  SHE=SHE-1.0
664  NKOSU=SHE-1.0
666  NKOSU=NKOSU-MABS
      IF(NKOSU. LT. 0) NKOSU=0
600  IF(INDEX. NE. 0) GO TO 900
      NKOSU=NKOSU+1
      NO(1)=0
      JTOP=1
602  DO 604 NA1=JTOP, NKOSU
      NO(NA1)=NO(NA1)+1
      IF(NO(NA1). GT. (JEDGE-NKOSU+NA1)) GO TO 606
      NA6=NO(NA1)
      NUM(NA6)=1
      IF(NKOSU. EQ. NA1) GO TO 700
      NA2=NA1+1
604  NO(NA2)=NO(NA1)
      GO TO 700
606  IF(NA1. EQ. 1) GO TO 600
      JTOP=NA1-1
      GO TO 602
700  NZERO=0
      DO 608 NZERO=1, JEDGE
      DO 610 NA3=1, NKOSU
      IF(NZERO. EQ. NO(NA3)) GO TO 608
610  CONTINUE
      NUM(NZERO)=0
608  CONTINUE
C      **** CACULATION OF LOGICAL SUN ****
801  DO 802 IA=1, N2
802  LSUN(IA)=0
      IF(MABS. NE. 0) GO TO 810
      DO 804 NN1=1, MEDGE
      IF(NUM(NN1). EQ. 0) GO TO 804
      DO 804 NN2=1, N1
      LSUM(NN2)=LSUM(NN2)+COV(NN1, NN2)
804  CONTINUE
      GO TO 815
810  DO 814 NN1=1, JEDGE
      IF(NUM(NN1). EQ. 0) GO TO 814
      DO 814 NN2=1, N2
      LSUM(NN2)=LSUM(NN2)+COVER(NN1, NN2)
814  CONTINUE
815  LPROD=1

```

```

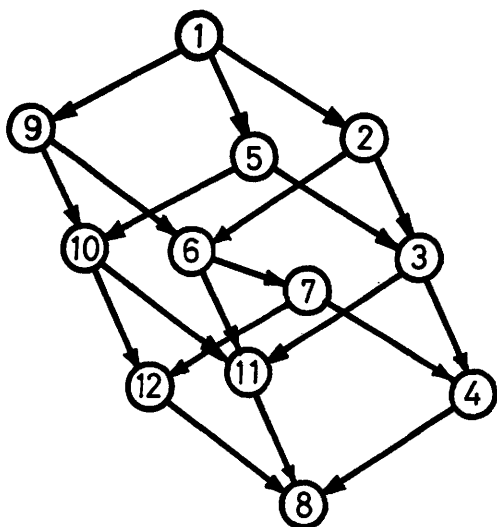
DO 806 NN1=1, N2
LPROD=LPRO*DLSUM(NN1)
IF(LPROD.EQ. 0) GO TO 808
806 LPROD=1
INDEX=INDEX+1
WRITE(6, 6000) INDEX, (NUM(NN4), NN4=1, JEDGE)
6000 FORMAT(/I5, 25I3/(5X, 25I3))
808 JTOP=NKOSU
GO TO 602
900 STOP
END

C      **** TRANSFORMATION OF WRITE STATEMENT ****
SUBROUTINE MATRIX(MAT1, N, M, JA1, JA2, MAT2, JC1, MAT3, JC2)
DIMENSION MAT1(N, M) MAT2(JC1), MAT3(JC2)
JA3=0
JA4=JA2
550 IF(JA4.LE. 25) GO TO 552
JB1=25*JA3+1
JB3=JA3+1
JB2=25*JA3
JA4=JA4-25
WRITE(6, 5600) (MAT3(JA5), JA5=JB1, JB2)
DO 554 JA5=1, JA1
554 WRITE(6, 5500) MAT2(JA5), (MAT1(JA5, JA6), JA6=JB1, JB2)
GO TO 550
552 IF(JA3.EQ. 0) GO TO 558
JB2=JB2+1
WRITE(6, 5600) (MAT3(JA5), JA5=JB2, JA2)
DO 556 JA5=1, JA1
556 WRITE(6, 5500) MAT2(JA5), (MAT1(JA5, JA6), JA6=JB2, JA2)
RETURN
558 WRITE(6, 5600) (MAT3(JA5), JA5=1, JA2)
DO 560 JA5=1, JA1
560 WRITE(6, 5500) MAT2(JA5), (MAT1(JA5, JA6), JA6=1, JA2)
RETURN
5500 FORMAT(I5, 25I3)
5600 FORMAT(/5X, 25I3)
END

```

処理データの例

付図1に示す a-cyclic SEC グラフを UL法によって1-識別可能にするための最少内部テスト端子を求めた結果は以下のごとくである。



付図1 例題のための a-cyclic SEC グラフ

INPUT ADJACENCY MATRIX

#	1	2	3	4	5	6	7	8	9	10	11	12
1	0	1	0	0	1	0	0	0	1	0	0	0
2	0	0	1	0	0	1	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0	1	0
4	0	0	0	0	0	0	0	1	0	0	0	0
5	0	0	1	0	0	0	0	0	0	1	0	0
6	0	0	0	0	0	0	1	0	0	0	1	0
7	0	0	0	1	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	1	0	0	0	1	0	0
10	0	0	0	0	0	0	0	0	0	0	1	1
11	0	0	0	0	0	0	0	1	0	0	0	0
12	0	0	0	0	0	0	0	1	0	0	0	0

VERTEX ORDERING

#	1	2	3	4	5	6	7	8	9	10	11	12
#	1	2	5	3	9	6	7	4	10	11	12	8

: 入力時の節点番号
 ## : ordering された節点番号

ORDERED ADJACENCY MATRIX

##	1	2	3	4	5	6	7	8	9	10	11	12
1	0	1	1	0	1	0	0	0	0	0	0	0
2	0	0	0	1	0	1	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	1	0	0	0
4	0	0	0	0	0	0	0	1	0	1	0	0
5	0	0	0	0	0	1	0	0	1	0	0	0
6	0	0	0	0	0	0	1	0	0	1	0	0
7	0	0	0	0	0	0	0	1	0	0	1	0
8	0	0	0	0	0	0	0	0	0	0	0	1
9	0	0	0	0	0	0	0	0	0	1	1	0
10	0	0	0	0	0	0	0	0	0	0	0	1
11	0	0	0	0	0	0	0	0	0	0	0	1
12	0	0	0	0	0	0	0	0	0	0	0	0

REACHABILITY MATRIX

##	1	2	3	4	5	6	7	8	9	10	11	12
1	1	1	1	1	1	1	1	1	1	1	1	1
2	0	1	0	1	0	1	1	1	0	1	1	1
3	0	0	1	1	0	0	0	1	1	1	1	1
4	0	0	0	1	0	0	0	1	0	1	0	1
5	0	0	0	0	1	1	1	1	1	1	1	1
6	0	0	0	0	0	1	1	1	0	1	1	1
7	0	0	0	0	0	0	1	1	0	0	1	1
8	0	0	0	0	0	0	0	1	0	0	0	1
9	0	0	0	0	0	0	0	0	1	1	1	1
10	0	0	0	0	0	0	0	0	0	1	0	1
11	0	0	0	0	0	0	0	0	0	0	1	1
12	0	0	0	0	0	0	0	0	0	0	0	1

MODIFIED ADJACENCY MATRIX

##	1	2	3	4	5	6	7	8	9	10	11	12
1	0	1	2	0	3	0	0	0	0	0	0	0
2	0	0	101	4	102	5	0	0	109	0	0	0
3	0	101	0	6	103	105	107	0	7	0	0	0
4	0	0	0	0	104	106	108	8	110	9	116	0
5	0	102	103	104	0	10	0	0	11	0	0	0
6	0	0	105	106	0	0	12	0	111	13	0	0
7	0	0	107	108	0	0	0	14	112	114	15	0
8	0	0	0	0	0	0	0	0	113	115	117	16
9	0	109	0	110	0	111	112	113	0	17	18	0
10	0	0	0	0	0	0	114	115	0	0	118	19
11	0	0	0	116	0	0	0	117	0	118	0	20
12	0	0	0	0	0	0	0	0	0	0	0	0

101~118は dummy edge の番号である。

COVERING MATRIX

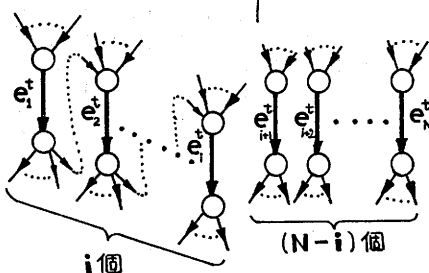
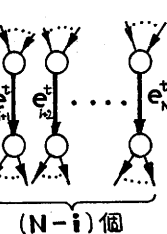
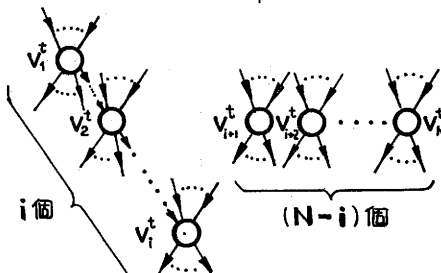
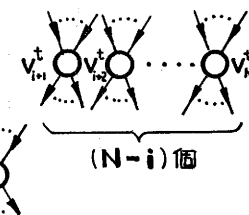
EUE_s の枝番号 E の枝番号		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	101	102	103	104	105
1	2	1	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	1	1	0	1	1
2	1	2	1	1	0	0	0	0	0	0	1	0	1	1	1	0	0	0	0	0	0	1	0	1	1	1
3	1	1	2	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
4	0	1	1	2	1	1	0	0	0	0	0	0	1	1	0	0	1	0	0	1	1	1	0	1	0	1
5	0	1	1	1	2	0	0	1	1	1	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0	1
6	1	0	1	1	0	2	1	0	0	0	0	0	1	1	0	0	1	0	0	1	1	0	1	1	1	1
7	1	0	1	0	0	1	2	0	1	0	1	0	1	0	1	1	0	0	0	0	1	0	1	0	1	1
8	0	0	1	0	1	0	1	2	1	0	0	0	0	1	0	0	0	0	1	1	0	1	1	1	1	1
9	0	0	1	0	1	0	1	1	2	0	0	0	1	0	0	1	1	0	0	1	1	0	1	1	1	1
10	1	1	0	0	1	0	0	1	1	2	1	0	0	0	0	0	1	1	0	0	0	1	1	1	1	1
11	1	1	0	0	0	0	1	0	1	1	2	0	1	0	1	1	0	0	0	0	0	1	1	1	1	0
12	0	1	0	1	0	0	0	1	0	0	1	2	1	0	0	0	0	1	1	0	1	0	1	1	1	1
13	0	1	0	1	0	0	0	1	0	1	1	2	0	0	1	1	0	0	1	1	0	1	0	1	1	1
14	0	1	0	1	0	0	0	1	0	0	1	0	1	2	1	0	0	0	1	1	1	0	1	1	1	1
15	0	1	0	1	0	0	0	0	0	0	1	0	1	1	2	1	0	1	1	0	1	0	1	1	1	1
16	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	2	0	0	1	1	0	0	0	0	0	0
17	1	0	0	0	0	1	0	0	1	1	0	0	1	0	0	1	0	0	1	2	1	0	1	1	0	1
18	1	0	0	0	0	1	0	0	0	1	0	0	0	0	1	1	1	2	1	0	1	1	0	1	1	1
19	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	1	2	1	0	0	0	0	0	0
20	0	0	0	1	0	1	0	0	0	0	0	0	1	1	0	1	1	0	1	2	0	0	0	1	0	1

106	107	108	109	110	111	112	113	114	115	116	117	118
26	27	28	29	30	31	32	33	34	35	36	37	38
1	0	1	0	1	1	1	1	1	0	0	0	0
2	1	1	1	1	0	1	1	0	1	0	0	0
3	1	1	1	1	1	0	0	0	0	1	0	0
4	1	0	1	1	1	0	0	1	1	0	1	1
5	1	1	1	1	0	1	1	1	0	0	1	0
6	1	1	1	0	1	0	0	1	1	0	1	1
7	0	1	0	1	1	1	1	1	1	1	1	0
8	1	1	1	1	1	0	0	1	0	1	1	1
9	1	1	1	1	1	0	0	0	1	1	1	0
10	1	1	1	0	0	1	1	1	0	0	1	0
11	0	0	0	1	1	1	1	1	1	1	1	0
12	1	1	1	1	0	1	1	1	1	1	1	0
13	1	0	0	1	0	1	0	0	1	1	0	0
14	1	1	1	1	0	1	1	1	1	1	0	1
15	1	1	1	1	0	1	1	0	1	0	1	1
16	0	0	0	1	1	1	1	1	1	1	1	0
17	0	1	0	1	1	1	1	1	1	1	0	0
18	0	1	0	1	1	1	1	1	0	0	1	1
19	0	1	1	0	0	0	1	1	1	1	1	0
20	1	0	1	0	1	0	0	1	1	0	1	1

THERE IS NO ABSOLUTE EDGE

MINIMUM EDGES OF UL*INTERNAL TEST EDGES

E の枝番号 最少内部テスト端子 集合の組番号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0
3	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0

テスト法	枝レベル		節点レベル	
	UL法	U法, L法	UL法	U法, L法
上限	$N \leq n-1$	$N \leq n-1$	$N \leq n-2$	$N \leq n-1$
一般下限	$N \geq \lceil \log_2(n+1) \rceil - 1$	$N \geq \lceil \log_2 n \rceil$	$n \leq 2^{N+1} + N - 1$ ②③	$N \geq \lceil \log_2 n \rceil$
内部テスト端子間に特別の関係と有する場合の下限	$n \leq \{2(1+i) \cdot 2^{N-i} - 1\} + (i-1) \cdot i/2 \dots \text{⑩}$  i個 (N-i)個 内部テスト端子 e_i^t の相互関係	$n \leq (i+1) \cdot 2^{N-i}$  (N-i)個 内部テスト端子 e_i^t の相互関係	$n \leq \{2(1+i) \cdot 2^{N-i} - 1\} + (i-1) \cdot i/2 + N$ ④  i個 (N-i)個 内部テスト端子 v_i^t の相互関係	$n \leq (i+1) \cdot 2^{N-i}$  (N-i)個 内部テスト端子 v_i^t の相互関係

付表1 枝レベルおよび節点レベルの内部テスト端子数の上限および下限 式⑩および式⑪において $i=1$ とすると、それぞれ枝レベルの一般下限式 式⑩および節点レベルの一般下限式 式②③になる。
 (注) 付表1の式④は $n \leq \{2(i+1)2^{N-i} - 1\} + (i-1)i/2 + N$ が正しいことを付記する。

付録2

枝レベルおよび節点レベルにおける内部テスト端子数の上限および下限を各テスト法について表にしたものを付表1に示す。付表1には内部テスト端子の間に特別の到達可能関係を与えたときの各テスト法に対する内部テスト端子数の下限をも示してある。いま、式⑩において $i=1$ とすれば、定理15の関係式を得ることに注意されたい。

また、式⑪において $i=1$ とすれば、節点レベルのUL法における下限を与える式を得る。

付図2には、枝レベルのUL法の内部テスト端子数 N と節点数 n の関係が示されており、点線は式⑩の関係を示すものである。

付表1の式⑩の説明

内部テスト端子に付表1に示されるような関係をもたせると、まず、枝レベルのUL法のととき、

$$V(v_1, f(e_1^t)) \neq V(v_1, f(e_2^t)) \neq \dots \neq V(v_1, f(e_i^t)) \quad \dots \text{②1}$$

$$V(v_1, f(e_j^t)) \neq V(v_1, f(e_k^t))$$

$$j \neq k, j = i+1, i+2, \dots, N \\ k = 1, 2, \dots, N \quad \dots \text{②2}$$

$$V(s(e_1^t), v_n) \neq V(s(e_2^t), v_n) \neq \dots \neq V(s(e_i^t), v_n) \quad \dots \text{②3}$$

$$V(s(e_j^t), v_n) \neq V(s(e_k^t), v_n) \\ j \neq k, j = i+1, i+2, \dots, N \\ k = 1, 2, \dots, N \quad \dots \text{②4}$$

なる関係がある。そこで、式②1, ②2, ②3, ②4を考慮して最大の相異なる行数をもった故障表を書くと付表2のようになる。

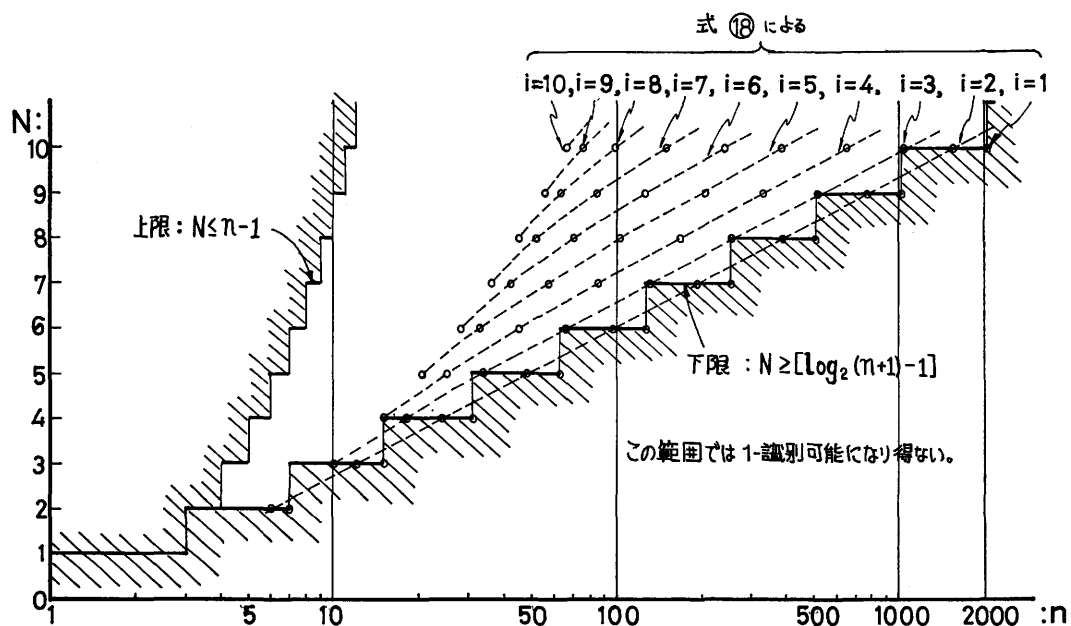
したがって、行の数は $2(\{2(i+1)2^{N-i} - 1\} + (i-1) \cdot i/2)$ となる。ゆえに、

$$n \leq \{2(i+1)2^{N-i} - 1\} + (i-1) \cdot i/2 \quad \dots \text{②5}$$

が得られる。

また、付表1の枝レベルのU(L)法の場合は付表2の故障表のU(L)法のところのみに注目すれば、相異なる最大行数は $(i+1)2^{N-i}$ となる。したがって、U(L)法に対して次式が得られる、

$$n \leq (i+1)2^{N-i} \quad \dots \text{②6}$$



付図2 UL法における内部テスト端子数 N と節点数 n の関係