

記憶を利用した順序回路の故障検査系列*

上林 弥彦** 矢島 脩三***

Abstract

In this paper, distinctions among several known procedures for fault detecting experiments are discussed and general procedures including them are given. As a special case, a procedure is given under the restriction of the upper bound of memory. This procedure is effectively used for a sequential machine realized by a shift register circuit. Several discussions of its application to other cases are also given.

1. まえがき

順序回路の故障検査では、大別して、つぎの2つの方法が知られている。

- (1) 故障検査をするための回路を付け加える。
- (2) 入出力関係から故障発見のできるような検査入力系列を作成する。

一般に、(1)の方法では回路が複雑になれが、回路の動作中に故障発見のできる特色がある。(2)の方法は、回路の複雑さの代償として長い検査時間が要求される。最近では、(1)と(2)の中間的な立場として、回路を少し複雑にすることにより、入出力関係によって故障発見でき、しかも検査入力系列の作成が容易になるような方法も扱われている。集積回路では、回路の複雑さに対して入出力線数が少ないので、時間的冗長さによって入出力関係のみで故障検査できる(2)の方法が有用である。しかし、すべての順序回路が(2)の方法で故障検査できるとは限らない。このようなことができるため、通常、つぎの3つの条件を仮定する。

- (a) 与えられた順序回路は強連結である。
- (b) もとの順序回路を含むような故障回路はない。(これは、入力記号数の増加、出力記号数の増加、もとの順序回路の状態より到達できない状態の付加などによって起こる)。

(c) 故障回路の数は有限である。

当然のことながら、等価な状態の付加は故障とはみ

なさない。ある状態から到達できない状態があれば、その状態を初期状態とする故障検査系列では、到達できない状態についての情報が得られない。このためにはすべての状態へ、到達可能な状態が1つあればよいが、任意の状態からこの状態へリセットできるためには、(a)の条件が必要となる。(b)の条件に反する故障回路が存在する場合、これともの順序回路とを入出力系列のみで分けるのは、故障のしかたがはっきりしていない限り無理である。検査系列長は有限でなければならないので、有限長の検査入力系列で可算無限個以上の順序回路から1つの順序回路を分けることができない場合の起こる可能性があり、(c)の条件を仮定すれば、このような場合を除くことができる。

このような立場から、従来の故障検査系列の作成法を分類すれば、つぎのようになる。この場合、上記の条件以外におおのこの方法で、さらにいくつかの条件が追加されることを注意しておく。

- (1) 回路の実現方法を考え、その可能なすべての故障を考え、それに対応する故障回路(有限個)と、もとの回路とを分ける系列を考える。
- (2) 回路の状態数の上限 n を与え、状態数 n 以下のすべての順序回路のうち、与えられた順序回路を含むような順序回路(条件(b)による)を除いた回路全体(有限個)と、与えられた回路とを分ける系列を考える。
- (3) 回路の記憶の上限 μ を与え、記憶 μ 以下のすべての有限記憶順序回路のうち、与えられた順序回路を含むような順序回路(条件(b)による)を除いた回路全体(有限個)と与えられた回路とを分ける系列を考える。

(1)の方法は、Moore¹⁾, Seshu, Freeman²⁾, Poage, McClusky³⁾ らによって研究され有効なアルゴリズム

* Fault Detecting Sequences Utilizing Memory of Sequential Machines, by Yahiko KAMBAYASHI, (Dept. of Computer Science, University of Illinois), and Shuzo YAJIMA (Dept. of Information Science, Kyoto University)

** イリノイ大学計算機学科, 本研究には京都大学大学院在学中に
従事した

*** 京都大学工学部情報工学科

† 出力記号数の増加がないという制限はつけなくてよい (G!!!)

が知られているが、可能な故障回路の数が大きければ、この方法は非実用的なものとなっている。

Hennie⁴⁾ は、(2)の立場による検査系列の求め方について述べている。この場合、故障回路の数が非常に多いので、(1)の場合のように最短の系列を求めることは不可能に近いが、非常に興味深い手法を提唱している。(3)の立場は、一般的に、本論ではじめて扱われるものであるが、従来知られていた決定性順序回路の故障検査系列(長さ k の系列パターンが必ず一度ずつ現われるような系列)³⁾やインパルス応答(系列100...)による線形順序回路の同定⁶⁾などは、この立場の特殊なものであるとみなすことができる。

このように整理すれば、これらの方法は(b)、(c)の条件を満足するような適当な故障条件を与えて、その故障条件を満足するすべての順序回路と、与えられた順序回路とを分ける検査系列を作成する」というように一般化される。したがって、故障条件として、従来知られているもの以外でも、実際に意味のある合理的なものを考えれば、それに応じて異なった検査系列の作成方法を得ることができる。

2節では、基本的定義と故障検査系列に関する一般的な扱いを述べ、固定形および適応形の故障検査系列の作成法について述べる。

3節では、一つの故障条件として、有限記憶順序回路の記憶の上限を与えた場合についての故障検査系列について考慮している。この検査系列は順序回路が、シフトレジスタを用いて構成されているときに利用すれば、有用である。それは、状態数の上限の制限よりも、つぎの点で実際的であるためである。

(1) シフトレジスタ回路では、等価な状態割当てや、冗長状態があるので、故障によって状態数のふえることが多い。

(2) シフトレジスタの段数のふえる故障は、一般に考えなくてよい。

順序回路の記憶を考慮することによる Hennie 流の検査系列の求め方も示した。

2. 故障検査系列に関する基本的考察

2.1 基本的事項

以下で扱う順序回路は Mealy 形とする。順序回路 M は、 $M=(S, X, Y, \delta, \lambda)$ で示される。ここで、 $S=\{s_1, s_2, \dots, s_n\}$ は状態の有限集合、 $X=\{x_1, x_2, \dots, x_r\}$ は入力記号の有限集合、 $Y=\{y_1, y_2, \dots, y_q\}$ は出力記号の有限集合であり、 δ および λ はそれぞれ

次状態関数および出力関数と呼ばれ、つぎの(1)、(2)の形で与えられる。

$$s(t+1)=\delta(s(t), x(t)) \quad (1)$$

$$y(t)=\lambda(s(t), x(t)) \quad (2)$$

ここで、 $s(t)$ 、 $x(t)$ 、 $y(t)$ は、それぞれ時刻 t における状態、入力、出力である。

δ および λ は、つぎの(3)、(4)、(5)によって入出力系列に拡張される。

$$s(t+i)=\delta(s(t), x(t)x(t+1)\dots x(t+i-1)) \quad (3)$$

$$s(t)=\delta(s(t), \epsilon) \quad (4)$$

$$y(t+i)=\lambda(s(t), x(t)x(t+1)\dots x(t+i)) \quad (5)$$

ここで、 ϵ は長さ0の系列である。

[定義 1]^{3), 12)} つぎの(6)または(7)を満足する最小の負でない整数 μ_i 、 μ_0 が存在すれば、この回路は有限記憶順序回路であるといい、入力記憶 μ_i 、出力記憶 μ_0 、記憶 $\mu=\max(\mu_i, \mu_0)$ を持つという、

$$y(t)=f(x(t), x(t-1), \dots, x(t-\mu_i), y(t-1), \dots, y(t-\mu_0)) \quad (6)$$

$$y(t)=f(x(t), x(t-1), \dots, x(t-\mu_i)) \quad (\mu_0=0 \text{ に対応}) \quad (7)$$

$\mu_i=0$ のときを出力記憶形、 $\mu_0=0$ のときを入力記憶形と呼び、有限記憶でないものを無限記憶順序回路という。

状態数 n の回路では、

$$\mu \leq \frac{1}{2}n(n-1) \quad (8)$$

が成立することが知られている。

[定義 2] 状態集合遷移関数 θ_i 、 θ_{i0} をつぎのように定義する。ここで、 $X^*(Y^*)$ は、長さ0をも含む入(出)力系列のすべてよりなる集合である。

$$S_1 \subset S, \omega \in X^*, \theta_i(S_1, \omega) = \{s' \mid s' = \delta(s, \omega), s \in S_1\} \quad (9)$$

$$S_1 \subset S, \omega \in X^*, \nu \in Y^*, \theta_{i0}(S_1, \omega/\nu) = \{s' \mid s' = \delta(s, \omega), \nu = \lambda(s, \omega), s \in S_1\} \quad (10)$$

とくに、 $\theta_i(S, \omega) = \{s\}$ のように1状態になる場合、入力系列 ω は順序回路 M における状態 s に対する同期系列と呼ばれる。

$s \approx s'$ ($s, s' \in S$) ならば

$$\lambda(s, \omega) \approx \lambda(s', \omega)$$

が成立するとき、 ω を初期状態決定系列という。

系列 ω を加えたときの出力系列により、一意的に最終状態が決定できる場合、 ω を最終状態決定系列と呼

ぶ. 初期状態と入力系列が与えられると, 最終状態が決まるので, 初期状態決定系列は最終状態決定系列である. また定義により, 同期系列も最終状態決定系列となる.

2.2 故障検査系列の一般的な扱い

故障検査系列の作成方法は1節で述べたように故障条件によって変わってしまうものである. ここでは, 故障条件を一般的な形で与えて, 故障検査系列の作成法を考察する.

順序回路 M を, 系列集合の受理回路または変換回路として扱う場合, 初期状態を含めて1つの順序回路の定義がされる. いいかえれば, 同じ M でも初期状態が異なれば, 異なる順序回路とみなされる. M にある初期状態 s を与えたときの順序回路を $M(s)$ と表現することにする.

故障条件 Ω は, 初期状態付きの順序回路全体の集合 μ_0 から 2^{μ_0} (μ_0 の部分集合の集合) への写像である.

$$\Omega(M(s)) = \{M_1(s_1), M_2(s_2), \dots, M_m(s_m)\}$$

であれば, $M(s)$ が故障すれば, $M_i(s_i)$ ($i=1, \dots, m$) のどれかになることを示している. Ω はさらにつぎの条件を満足するとする.

(1) $\Omega(M(s)) = \{M_1(s_1), M_2(s_2), \dots, M_m(s_m)\}$ であれば, M の s 以外の任意の状態 s' を考えると

$$\Omega(M(s')) = \{M_1(s'_1), M_2(s'_2), \dots, M_m(s'_m)\}$$

となる. すなわち, 初期状態のみ異なる回路どうしの故障回路も初期状態のみ異なる (もちろん, おなじでもよい).

(2) すべての $M(s)$ について $\Omega(M(s))$ は M と同形の回路および M を含む順序回路を含まない.

(3) すべての $M(s)$ について $\Omega(M(s))$ の要素数は有限である.

(1) の制限は, 故障条件がハードウェアの故障から導かれるとすれば妥当であると思われる. (2), (3) は1節の条件 (b), (c) に対応するものである.

Hennie は, $\Omega(M(s))$ は M と同じかそれ以下の状態数のすべての順序回路のうち M および M を含む回路を除いた集合とした場合の検査系列の求め方を示した. 次節では M と同じかそれ以下の記憶長を持つすべての順序回路のうち M および M を含む回路を除いた集合とする場合について議論する.

つぎに, 検査系列の一般的な構成法を与えるために若干の記号を定義する.

R を X^* の部分集合とする. $\omega \in X^*$ に対し

$$R[\omega] = \{\omega' \mid \omega\omega' \in R\}$$

と定義する. また R の要素のうち最小長のものの1つを $\min(R)$ とする.

検査系列を加える場合に, つぎに加える入力があるまでの入力系列を加えた応答によって変わる場合を適応形といい, そうではなく, あらかじめ決められた一種類の入力系列のみを考える場合を固定形という.

$\Omega(M(s)) \cup M(s)$ の中から $M(s)$ だけを, その系列を加えたときの応答によって分けることのできるような入力系列すべての集合を

$$U_{\Omega}(M(s))$$

と表わす.

$$\omega \in U_{\Omega}(M(s)) \text{ なら } \omega' \in X^* \text{ として,}$$

$$\omega\omega' \in U_{\Omega}(M(s))$$

となる.

$U_{\Omega}(M(s))$ の要素と $U_{\Omega}(M(s))[\omega]$ の要素を求める方法があれば, つぎの方法で検査系列の構成ができる.

(固定形検査系列の構成法)

1. 順序回路 M が状態 s に対する同期系列 ω を持つ場合

$$\omega' \in U_{\Omega}(M(s))$$

として, 固定検査系列は

$$\omega\omega'$$

となる ($\omega' = \min(U_{\Omega}(M(s)))$ とするとよい).

2. 同期系列を持たない場合

つぎの l ができるだけ小さくなるような ω を選ぶ.

$$\Theta_l(S, \omega) = \{s_{a_1}, s_{a_2}, \dots, s_{a_l}\}$$

ω' をつぎのように選べば, $\omega\omega'$ が検査系列である.

$$\omega' \in \bigcap_{i=1}^l U_{\Omega}(M(s_{a_i}))$$

ω' の求め方として最短ではないが, つぎの方法を利用することもできる.

$$\omega_{a_1} \in U_{\Omega}(M(s_{a_1}))$$

$$\omega_{a_2} \in U_{\Omega}(M(s_{a_2}))[\omega_{a_1}]$$

$$\vdots$$

$$\omega_{a_l} \in U_{\Omega}(M(s_{a_l}))[\omega_{a_1} \dots \omega_{a_{l-1}}]$$

とすれば, $\omega' = \omega_{a_1} \dots \omega_{a_l}$ となり, 検査系列は

$$\omega\omega_{a_1} \dots \omega_{a_l}$$

となる. この入力系列 $\omega_{a_1} \dots \omega_{a_l}$ によって許しうる出力の種類は ω を加えたあとの状態に応じて, l 種類あることになる.

この方法は, まず ω を加えて可能な状態の数をへらしてやる. ω を加えたあとの状態が s_{a_i} であると仮定したときの検査は $\omega_{a_1} \omega_{a_2} \dots \omega_{a_l}$ で行なわれるため, この系列で故障検査が可能である. この場合, ω の選

び方や, $s_{a_1} \dots s_{a_t}$ の名前のつけ方によって全体の系列長が影響されることを注意しておく。

〔適応形検査系列の構成法〕

1. 初期状態決定系列 ω を持っている場合

ω を加えた結果, 順序回路の初期状態が s であるとわかれば

$$\omega' \in U_Q(M(s))[\omega]$$

を加えてやればよい。初期状態決定系列は最終状態決定系列でもあるので, つぎの方法によってもよい。

2. 初期状態決定系列を持たない場合

任意の順序回路は, 最終状態決定系列 ω を持つから, ω を加えた結果の状態を s として

$$\omega_s' \in U_Q(M(s))$$

を求め, ω のあとに ω_s' を加えてやればよい。

$U_Q(M(s))$ の要素は S のすべての s についてではなく, $\theta_i(S, \omega)$ に属す s についてのみ求めておけばよい。しかし, これを求める操作は複雑なので, ある状態 s_0 についてのみ $U_Q(M(s_0))$ の要素を求めておき, ω を加えた結果, 状態 s' に移ったとすると, s' より s_0 に遷移させる入力を ω'' として, $U_Q(M(s'))$ の要素を $\omega''\omega'$ としてやればよい。

したがって, $\Omega, M(s)$ が与えられたとき, $U_Q(M(s))$ や $U_Q(M(s))[\omega]$ の要素のうち, できるだけ短い系列を求めることが重要である。

3. 有限記憶順序回路の故障検査系列とその応用

本節では, 有限記憶順序回路 M が与えられた場合, $\Omega(M(s))$ は, $M(s)$ とただかた同じ長さの記憶になるような制限を満足する順序回路の集合とした場合の, $U_Q(M(s))$ の求め方について述べ, 最後に Hennie 流の状態数の上限を制限した場合の, 系列を計算機で求めるための応用を示す。

3.1 基本的定義

長さ k の入出力系列対を $Q_k = \omega_k/\nu_k$ で表わす。ここで, 入力系列 ω_k , 出力系列 ν_k の長さは k である。次式が成立すれば, 入出力系列対 ω/ν は状態 s に適用できるという。

$$\nu = \lambda(s, \omega)$$

記憶の木は, 順序回路の記憶を求めるためのもので, おのおのの道は入出力系列に対応する。根よりの道の長さを各節点のレベルという。各節点は, 状態集合 $S_k(Q_k)$ に対応する。ここで, k はその節点のレベルであり, Q_k はこの点へ至る道を示す長さ k の入出力

系列である。

〔定義 3〕 順序回路 M に対応する長さ h の記憶の木 $T_h(M)$ は, つぎのようにして構成されるレベル h 以下の, すべての節点を含む木である。

(1) 根に対応する状態集合

$$S_0(\varepsilon/\varepsilon) = S = \{s_1, s_2, \dots, s_n\} \tag{10}$$

(2) もし, 空でない $S_k(\omega/\nu)$ と $S_{k+1}(\omega x/\nu y)$ が存在して, (11) 式が満足されるなら ($0 \leq k \leq h-1$), $S_k(\omega/\nu)$ に対応する節点から $S_{k+1}(\omega x/\nu y)$ に対応する節点へ向かう, x/y と名づけられた有向枝が存在する。

$$\theta_{i0}(S_k(\omega/\nu), x/y) = S_{k+1}(\omega x/\nu y) \tag{11}$$

あきらかに, 記憶 μ を持つ最小化された有限記憶順序回路では, レベル $k (\geq \mu)$ 以上の節点に対応する状態集合はすべて1つの状態よりなる。

〔定義 4〕 順序回路 M に対する, 長さ h の記憶表 $L_h(M)$ および, 長さ h の記憶グラフ $G_h(M)$ は, ともに, つぎのようにして $T_h(M)$ より求められる。

$L_h(M)$: $T_h(M)$ のレベル h の点に対応する状態集合を $S_k(Q_k^1), S_k(Q_k^2), \dots, S_k(Q_k^m)$ とし, $S_k(Q_k^i) = \{s_{i1}, \dots, s_{iji}\} (i=1, \dots, m)$ とする。

各行は, すべての s_{ij} および x_i について, つぎの要素より成る。

$$Q_k^i, s_{ij}, x_i/y_i, \delta(s_{ij}, x_i)$$

ここで, $y_i = \lambda(s_{ij}, x_i)$ である。

$G_h(M)$: $L_h(M)$ の各行に 1, 2, ..., t と番号をつけ, これらに対応する t 個の点と, 下に定義する枝より成る。 $x, x_i \in X, y, y_i \in Y, \omega \in X^*, \nu \in Y^*, L_h(M)$ の i 行目 $\omega x_i/\nu y_i, s, x_i/y_i, s', L_h(M)$ の j 行目 $\omega x_j/\nu y_j, s', x_m/y_m, s''$ ならば, そのときに限り節点 i から節点 j へむかう x_i/y_i と名づけられた有向枝がある。

グラフに G に対して, つぎのようにして得たグラフ G' を, グラフ G の interchange グラフという。すなわち, G の枝 i について対応する G' の節点 i があり, G においてある節点への入枝 j_1 と, その節点からの出枝 j_2 があるとき, そのときに限って, G' の節点 j_1 から節点 j_2 へ向かう有向枝があるという条件を満足している場合である。長さ 0 の記憶グラフ $G_0(M)$ は, 順序回路 M の状態遷移図の interchange グラフとなっている。

〔定義 5〕 $\text{su}f_i(\omega)$ は, 系列 ω の長さが i 以上のときは, うしろの長さ i の系列をとることを意味し, $i-1$ 以下では, そのままであることを示す関数であ

る。すなわち、 $\omega = x_1 \cdots x_2 x_1$ に対し、

$$l \geq i \text{ なら } \text{su}f_i(\omega) = x_i x_{i-1} \cdots x_1$$

$$l < i \text{ なら } \text{su}f_i(\omega) = \omega$$

である。

〔定義 6〕 入力記憶 μ_i 、出力記憶 μ_o 、記憶 $\mu = \max(\mu_i, \mu_o)$ を持つ有限記憶順序回路の完全記憶表 $L^{\epsilon_{\mu_i, \mu_o}}(M)$ および完全記憶グラフ $G^{\epsilon_{\mu_i, \mu_o}}(M)$ は、次のように定義される。

$L^{\epsilon_{\mu_i, \mu_o}}(M) : L_h(M) (h \geq \mu)$ の各行

$$Q_h^i, s_{ij}, x_i/y_i, \delta(s_{ij}, x_i)$$

に対して、 $L^{\epsilon_{\mu_i, \mu_o}}(M)$ の行

$$Q_h^{i'}, x_i/y_i, \delta(s_{ij}, x_i)$$

を得る。ここで

$$Q_h^i = \omega/\nu$$

として

$$Q_h^{i'} = \text{su}f_{\mu_i}(\omega)/\text{su}f_{\mu_o}(\nu)$$

である。 $L_h(M)$ のいくつかの行が $L^{\epsilon_{\mu_i, \mu_o}}(M)$ の 1 つの行に対応する場合がある。

$G^{\epsilon_{\mu_i, \mu_o}}(M) : L^{\epsilon_{\mu_i, \mu_o}}(M)$ の各行に対応する節点 (節点名は行番号となる) よりなり、つぎの条件が満足されると、節点 i から節点 j へ x_i/y_i と名づけられた枝がある。

$L^{\epsilon_{\mu_i, \mu_o}}(M)$ の i 行目、 j 行目がそれぞれ、

i 行目 $x_i/y_i Q, x_i/y_i,$

j 行目 $Q x_i/y_i, x_i/y_i$

である。

つきに、2つの操作を定義する。

〔操作 1〕 グラフ G は、 n 個の節点 $1, 2, \dots, n$ より成るとし、各枝は有向枝で整数の重みがついているとする。このような G と節点 i が与えられた場合、 i を始点として、すべての節点を通り、かつ、通路の各枝の重みの和のできるだけ小さい道を求める操作を操作 1 とする。

操作 1 は、与えられたグラフにつきのような変形をほどこすことにより、traveling salesman problem の手法を利用できる。ここで $\omega(i, j)$ を節点 i と節点 j を結ぶ枝の重みとする。

① グラフの 3 つの節点 i, j, k について

$$\omega(i, j) + \omega(j, k) < \omega(i, k)$$

なら i と k を結ぶ枝の重みを $\omega(i, j) + \omega(j, k)$ でおきかえる。もとのグラフは、すべての節点を通る道をもつことから強連結 (任意の節点から任意の節点へ至る道がある) であるから、節点 i と k の間に枝がない場合これを重み無限大の枝があるとみなす

と、上の操作によって得られるグラフは完全グラフ (任意の節点間を直接むすぶようなどちらの方向の枝も存在する) となる。

② 上記の変形で得られたグラフに traveling salesman problem の手法を適用する。解はすべての節点を一度だけ通ってもとへもどる閉じた道を与える。

③ ① の操作を逆にたどることにより、この道に対応するすべての節点を少なくとも一度通る閉じた道を求める。 $\omega(i, k)$ が $\omega(i, j) + \omega(j, k)$ で置き換えられていれば、② の解が節点 i から k に向う枝を通してあれば、これを $i \rightarrow j \rightarrow k$ と通る 2 つの枝で置き換える。

④ 得られた道は閉じた道となるために最後の部分はすでに通った節点のみを通してあることがある。そこで道の終わりの部分からすでに通っている節点を除いてやる。

Traveling salesman problem では解はすべての節点を一度だけ通る最短の閉じた道となっているので、与えられたグラフによっては同じ節点を 2 度以上通った方が短くなる場合があり、操作 1 では同じ節点を 1 度しか通らないという性質は不要なので①の変形をすることになる。閉じた道という条件も不要であるので④の変形で長さをへらしうる。

〔操作 2〕 操作 1 と同じく、グラフ G 、節点 i 、および G の節点の部分集合 S が与えられたとき、節点 i を始点として、 S に属するすべての節点を通り、かつ、通路の各枝の重みの和のできるだけ小さい道を求める。

操作 2 はつきのようにして操作 1 に還元できる。

(1) G の S に含まれない節点 i に注目する。もし、節点 j より節点 i にはいる重み a の枝があり、節点 i より出て節点 k にはいる重み b の枝があれば、節点 j より出て節点 k へ向う重み $a+b$ の枝を付け加える。節点 i を通るすべての長さ 2 の道に、この操作をして、節点 i を消す。

(2) 上の操作を S に含まれない節点が無くなるまでくり返す。得られたグラフを G' とする。

G に操作 2 を適用する代わりに、 G' に操作 1 を適用すればよい。すでに知られている traveling salesman problem に変形しないで直接とくアルゴリズムも知られている¹⁰⁾。

操作 1 は $Uq(M(s))$ の要素を求めるために、また、操作 2 は $Uq(M(s))[\omega]$ の要素を求めるために用いら

れる。

3.2 有限記憶順序回路の故障検査系列

入出力系列 Q_i に含まれるすべての長さ i の部分入出力系列の集合を $Q_i^{(1)}$ で示す。

〔定義 7〕 つぎの条件が満足されれば、入出力系列 Q_i は記憶 μ を持つという。

すべての $Q_{\mu}x/y \quad Q_{\mu}'x'/y' \in Q_i^{(\mu+1)}$

に対して $Q_{\mu} = Q_{\mu}', \quad x = x' \quad \text{ならば}$

$$y = y'$$

〔補題 8〕 Q_i が記憶 μ を持てば、記憶 $\mu+1$ を持つ。

〔補題 9〕 Q_i は記憶 $i-1$ を持つ。

〔定義 10〕 Q_i が、負でない整数 μ_i, μ_0 に対して記憶 $\mu = \max(\mu_i, \mu_0)$ を持ち、かつ、つぎの条件を満足すれば、この系列は完全 (μ_i, μ_0) 系列であるという。

Q_i の任意の部分系列 $x_1 \dots x_{\mu_i} / y_1 \dots y_{\mu_0}$ に対して、すべての $x \in X$ について、部分系列

$$x_2 \dots x_{\mu_i} x / y_2 \dots y_{\mu_0} y$$

が必ずただ一種のみ Q_i に含まれる。

〔定理 11〕 完全 (μ_i, μ_0) 系列は、入力記憶が μ_i 以下で、出力記憶が μ_0 以下のすべての有限記憶順序回路のうち、ただ 1 つの強連結成分を表わす。

したがって、入力記憶 μ_i 、出力記憶 μ_0 を持つ強連結有限記憶順序回路 M に対して、 $Q(M)$ は入力記憶 μ_i 以下、出力記憶 μ_0 以下のすべての有限記憶順序回路のうち、 M を含まないもの全体の集合となる場合は、 M を表現する完全 (μ_i, μ_0) 系列を求めることにより、故障検査系列を求めることができる。

強連結な有限記憶順序回路では、その完全記憶グラフも強連結になるので、つぎの方法で完全 (μ_i, μ_0) 系列が得られる。

〔定理 12〕 強連結有限記憶順序回路 M の完全記憶

グラフ $G^{\mu_i, \mu_0}(M)$ のすべての節点を通る 1 本の有向道に対応して、道の始端の節点に対応する記憶の入出力系列のあとに、道上の枝に対応する入出力対を順次つけることによって構成した入出力系列は、 M を表現する完全 (μ_i, μ_0) 系列となる。

すべての節点を通る有向道は、操作 1 を利用して求めることができる (各枝の重みはすべて 1 とする)。

$U_Q(M(s))$ の要素は、 s に対応する完全記憶グラフ上の節点を始点とし、すべての節点を通る有向道に対応する入力系列として求められる。 $U_Q(M(s))[\omega]$ の要素は、 s に対応する完全記憶グラフ上の節点を始点とし、入力系列 ω によって通過した道上の節点集合 (始点も含む) を S_1 とすると、 ω を加えおわったところの節点を始点として、 S_1 に含まれない完全記憶グラフのすべての節点を通る有向道に対応する入力系列として求められる。これには操作 2 を利用する。

入力記憶順序回路の完全記憶グラフはグッドの遷移図になるので、そのすべての節点を通る最短の道は、そのグラフのハミルトニアンになる。Gill は、 P 値入力、 μ_i -入力記憶の順序回路の長さ $p^{\mu_i+1} + \mu_i$ の入力系列で検査できることを示している。

節点数 n の強連結グラフのすべての節点を通る道の長さは、たかだか $\frac{n^2}{4}$ であるから

〔定理 13〕 p 値入力、 q 値出力で入力記憶 μ_i 、出力記憶 μ_0 を持つ強連結有限記憶順序回路 M を表現する完全 (μ_i, μ_0) 系列の長さはたかだか

$$\frac{1}{4} p^{2(\mu_i+1)} q^{2\mu_0} + \max(\mu_i, \mu_0)$$

である。

〔例 1〕 Fig. 1 の順序回路 M が与えられたとする。この回路の記憶の木は Fig. 2 に示されており、記憶 μ は 2 である。Table 1 は長さ 1 の記憶表 $L_1(M)$

| | 0 | 1 |
|---|------|------|
| 1 | 2, 1 | 3, 1 |
| 2 | 1, 0 | 3, 1 |
| 3 | 5, 1 | 4, 1 |
| 4 | 6, 0 | 3, 0 |
| 5 | 1, 0 | 4, 0 |
| 6 | 2, 1 | 4, 0 |

Fig. 1 Machine M

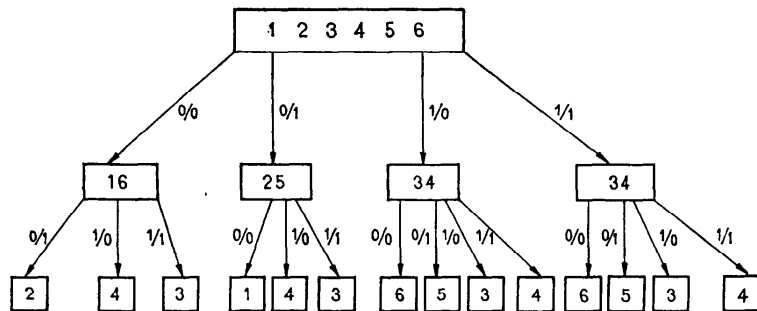


Fig. 2 Memory Tree for M .

Table 1 $L_1(M)$

| | Q_i^i | S_{ij} | x_i/y_i | S_{ij}' |
|----|---------|----------|-----------|-----------|
| 1 | 0/0 | 1 | 0/1 | 2 |
| 2 | 0/0 | 1 | 1/1 | 3 |
| 3 | 0/0 | 6 | 0/1 | 2 |
| 4 | 0/0 | 6 | 1/0 | 4 |
| 5 | 0/1 | 2 | 0/0 | 1 |
| 6 | 0/1 | 2 | 1/1 | 3 |
| 7 | 0/1 | 5 | 0/0 | 1 |
| 8 | 0/1 | 5 | 1/0 | 4 |
| 9 | 1/0 | 3 | 0/1 | 5 |
| 10 | 1/0 | 3 | 1/1 | 4 |
| 11 | 1/0 | 4 | 0/0 | 6 |
| 12 | 1/0 | 4 | 1/0 | 3 |
| 13 | 1/1 | 3 | 0/1 | 5 |
| 14 | 1/1 | 3 | 1/1 | 4 |
| 15 | 1/1 | 4 | 0/0 | 6 |
| 16 | 1/1 | 4 | 1/0 | 3 |

Table 2 $L_{2,1}(M)$

| | $Q_i^{i'}$ | | x_i/y_i |
|----|------------|-----|-----------|
| 1 | 0/d | 0/0 | 0/1 |
| 2 | 0/d | 0/0 | 1/1 |
| 3 | 0/d | 0/1 | 0/0 |
| 4 | 0/d | 0/1 | 1/1 |
| 5 | 0/d | 1/0 | 0/0 |
| 6 | 0/d | 1/0 | 1/0 |
| 7 | 0/d | 1/1 | 0/1 |
| 8 | 0/d | 1/1 | 1/1 |
| 9 | 1/d | 0/0 | 0/1 |
| 10 | 1/d | 0/0 | 1/0 |
| 11 | 1/d | 0/1 | 0/0 |
| 12 | 1/d | 0/1 | 1/0 |
| 13 | 1/d | 1/0 | 0/1 |
| 14 | 1/d | 1/0 | 1/1 |
| 15 | 1/d | 1/1 | 0/0 |
| 16 | 1/d | 1/1 | 1/0 |

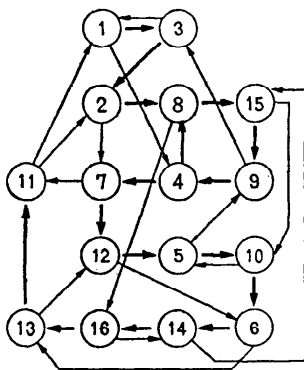


Fig. 3 Complete Memory Graph for M .

であり、Table 2 は完全記憶表 $L_{2,1}(M)$ である。これに対応する完全記憶グラフ $G_{2,1}(M)$ は Fig. 3 に示されている。

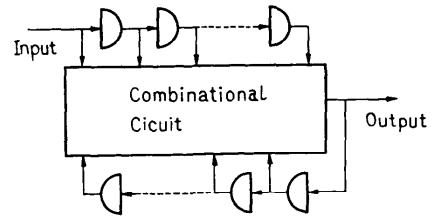


Fig. 4 Canonical Realization of Finite Memory Sequential Machine.

図3の太線は、すべての節点を通る道を示しており、これから得られる M を表現する、完全 (2, 1) 系列はつぎのようになる。

入力 000011001010111100

出力 101011011100001010

3.3 一般順序回路への応用

上記の有限記憶順序回路の故障検査系列は、Fig. 4 のような、有限記憶順序回路のシフトレジスタによる標準実現を考えた場合、シフトレジスタの段数のふえる故障が起こらなと仮定した場合の検査系列となっている。

[定理 14]⁷⁾ (Friedman) 任意の p 値入力順序回路は、適当に等価な状態を増すと、入力用の p 値シフトレジスタ 1 個と、2 値フィードバックシフトレジスタ 1 個を用いて構成できる。

[系 15] 任意の順序回路は、等価な状態をつけ加えれば、2 値の出力線を 1 本加えるだけで、有限記憶にできる。

この場合のシフトレジスタによる構成法を Fig. 5 に

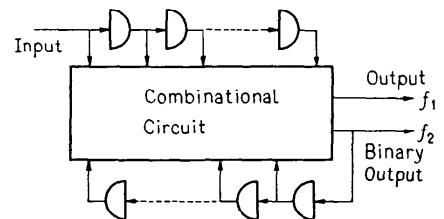


Fig. 5 Realization of Arbitrary Sequential Machine by Finite Memory Machine by Adding One Binary Output Line.

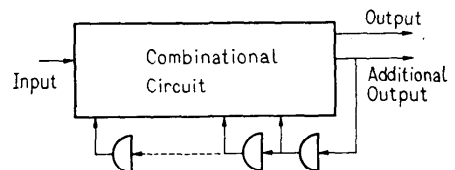


Fig. 6 Shift Register circuit.

示す。出力 f_1 と出力 f_2 をまとめて考えることにより、このような方法で実現した順序回路の検査系列を 3.2 節の方法で求めることができる。一般に、シフトレジスタによって構成された回路では、レジスタの入力端子より出力線を出すことにより有限記憶（出力記憶）とでき、3.2 節の方法が適用できる (Fig. 6)。

3.4 状態数の上限を制限した検査系列への応用

順序回路 M が与えられたとき、 $Q(M)$ を状態数 n 以下の M を含まないようなすべての順序回路の集合とする。このような場合の $UQ(M(s))$ の求め方を Hennie が示した。しかし、この手法ではあまり短い系列が得られない。村上氏¹¹⁾らの計算機による総当たり法では、状態数がすこし大きくなると実用的ではなくなる。記憶が長いほど、状態数が大きくなる傾向があるので、記憶グラフによる 1 つの方法を以下に示す。

[状態数の上限を制限した場合の検査系列]

- (1) $i=0$
- (2) 入力系列 ω_i を発生して (3) へ
- (3) $G_i(M)$ のすべての節点を ω_i による道が通れば (4) へ。そうでなければ i を $i+1$ として (2) へ。
- (4) n 状態以下で ω を加えたとき M と同じ出力を出すもの全体の数を決める。これが 1 個以下なら (6) へ。2 個以上 k_0 個以下なら (5) へ。その他の場合は (2) へ。
- (5) (4) で得られた M 以外の回路と M とを、それぞれ ω を加えたあとの状態を初期状態として分ける系列 v を求める。(7) へ。
- (6) 検査系列は ω である。
- (7) 検査系列は、 ωv である。

$k_0=1$ として (5) を省いてもよい。 $G_0(M)$ の代わりに適当な h について $G_h(M)$ を用いれば、(4) に要する時間を減らすことができる。しかし最短のものが得られるとは限らなくなる (村上氏らの方法は $G_0(M)$ を用いたことになる)。

(1), (2), (3) の代わりに (3) を満足する入力系列を操作 1 で求める。 k_0 を十分大きくするという方法もある。

【例 2】 Fig. 7 に示される順序回路 M' は初期状態決定系列を持たないものである。この記憶グラフ $G_1(M')$ は Fig. 8 に示されている。このグラフ上のすべての節点を通るという条件で系列を発生し、 $k_0=1$ として、つぎの検査系列が得られた。

1 0 0 1 0 0 0 0 1 1 1 1

| | | |
|-------|----------|----------|
| | 0 | 1 |
| S_1 | $S_2, 0$ | $S_3, 0$ |
| S_2 | $S_2, 0$ | $S_1, 1$ |
| S_3 | $S_3, 1$ | $S_1, 1$ |

Fig. 7 Machine M'

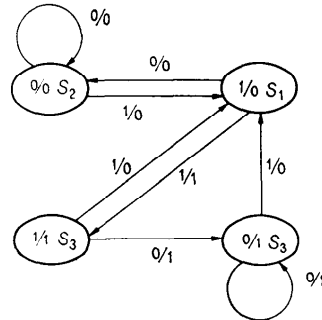


Fig. 8 $G_1(M')$

1 1 1 0 0 0 0 0 0 1 0 1

Hennie はこの回路に長さ 23 の検査系列を与えている。

とくに、状態数 n の有限記憶順序回路の場合、前節で与えた記憶の上限を制限した系列を考えると、これを満足し、かつ状態数が n 以下の順序回路は非常に少ないので (記憶長が与えられた順序回路より長い、無限であるような回路) これを初期系列とし、これに、上述の (5) の方法で得た系列をつけて検査系列としてもよい。

謝 辞

本研究に対して援助いただいた京都大学工学部前田憲一教授ならびに清野武教授に深謝する。

参考文献

- 1) E. F. Moore, "Gedanken-Experiments on Sequential Machines," *Automata Studies*, Princeton University Press, 1956.
- 2) S. Seshu and D. N. Freeman, "The Diagnosis of Asynchronous Sequential Switching Systems," *IRE Trans.*, vol. EC-11, No. 4, 1962.
- 3) A. Gill, "Introduction to the Theory of Finite-State Machines," McGraw-Hill, 1962.
- 4) F. C. Hennie, "Fault Detecting Experiments for Sequential Circuits," *IEEE, SCTLD*, 1964.
- 5) J. F. Poage and E. J. McClusky, "Derivation of Optimum Test Sequence for Sequential Machines," *IEEE, SCTLD*, 1964.

- 6) A. Gill, "Linear Sequential Circuits," McGraw Hill, 1966.
- 7) A. D. Friedman, "Feedbacks in Synchronous Sequential Switching Circuits," IEEE Trans., vol. EC-15, June 1966.
- 8) 上林, 矢島, "有限記憶順序回路の故障検査とその応用", 信学会オートマトン研資, 昭和 43 年 6 月.
- 9) 河田, 樹下, 尾崎, "シフトレジスタ形順序回路の故障検査について", 信学論 (C), 52-C, 7, p. 378 (昭和 44 年 7 月).
- 10) 茂木, "指定された nodes を経由する shortest path の問題", 電気 4 学会連大, 53, 昭和 45 年.
- 11) 村上, 樹下, 尾崎, "故障検査系列の計算機による探索", 信学誌, 50, 10, 1967.
- 12) Y. Kambayashi, S. Yajima and I. Ohbayashi, "On Finite-Memory Sequential Machines," IEEE Trans., vol. C-19, No. 3, 1970.
(昭和46年5月24日受付)