

双対オートマトンに基づいた適合性試験の 誤りシーケンスの実現手順

荒木 哲郎[†] 高橋 修^{††} 吉武 静雄^{††}

本論文では、遠隔試験法およびディレールポイント方式に基づいた適合性試験システムにおいて、試験シーケンスを実現する方法を提案する。すなわち汎用的な適合性試験システムとして、種々のプロトコル製品との通信を容易にするためのプロトコル実行部と、主にプロトコルエラーに関するシーケンスの生成・試験をつかさどる試験論理部からなる適合性試験システムにおいて、その上で実現されるべき試験論理を、双対なオートマトンの概念を用いて生成する方法を述べる。主な結果は次のとおりである。(i) 試験対象製品を正常な状態遷移動作および、エラーの遷移動作について網羅的に試験することができるためには、適合性試験システムが試験対象製品の実装プロトコルと双対な関係にあるオートマトンを実現することが必要であることを示した。(ii) 双対オートマトンに基づいた試験シーケンスを網羅的にかつ体系的に得るための手順および、それらを適合性試験システムのディレールポイント上で実行すべき試験論理を実現する手順を与えた。(iii) (ii)の手順によって生成される試験論理(L-アークと呼ぶ)は、プロトコルの状態遷移の約30~40%に当たる試験を実施するものであり、これをプロトコル実行部(P-アークと呼ぶ)が動作する環境下で種々組み合わせて用いることにより、60~70%程度のP-アークの遷移部分を共通化できるため、両者を一体化してそれぞれ個別に試験シーケンスを実現する方法に比べると、効率的に試験が実施できること、および実際に遠隔試験方法、ディレールポイント方式による試験システムを用いた実験結果を示した。

The Realization Procedure of Conformance Test Error Sequences Based on the Concept of a Dual Automaton

TETUO ARAKI,[†] OSAMU TAKAHASHI^{††} and SHIZUO YOSITAKE^{††}

This paper is concerned with the problems involved in realizing all the test sequence to be applied in a practical test system for exhaustively testing conformance of protocol implementations to a standard protocol. It proposes a new technique to systematically obtain all test logic required to realize the test sequences for all protocol errors. This technique is based on the concept of a "dual" Finite State Machine. Results of study include the following: (i) Exhaustive testing of a protocol implementation is possible if a test system realizes a dual Finite State Machine. (ii) Procedures for exhaustively and systematically obtaining test sequences based on the dual Finite State Machine and actually realizing them in a test system, are shown. (iii) These procedures are confirmed to be of practical use through our experiments with the test system (PROVES) which enabled an evaluation of the system's capability.

1. はじめに

プロトコルの標準化がISO等¹⁾で進むにつれて、プロトコルを実装した個々の製品が、標準プロトコルに正しく準拠していることを試験するシステム(適合性試験システムと呼ぶ)の開発が必要となっており、ISO/IEC JTC 1/SC 21で標準化作業が進められている²⁾。また従来このような適合性試験システムの研究がEuronet³⁾、NBS⁴⁾、NPL^{5),10)}等で行われており、

試験サービスも現在NCC(National Computing Center; 英)、CTS(Conformance Testing Services; 欧)、COS(Corporation for Open Systems; 米)およびINTAP(情報処理相互運用技術協会; 日)等で行われている¹⁵⁾。

適合性試験システムを研究・開発する上での重要な問題は、試験対象製品を網羅的に試験するためのシーケンスの生成アルゴリズムを研究し、それらを実際の適合性試験システムに実装する方法を研究することである。従来、試験シーケンスの生成法については文献5), 9), 11)~15)の研究があるがこれらの正常系および異常系の試験シーケンスを試験システム内でいかに実現するかについては、抽象的なレベルで扱うか、ま

[†] 福井大学工学部電子工学科

Department of Electrical and Electronics Engineering, Faculty of Engineering, Fukui University

^{††} NTT ネットワーク高度化推進本部

Network Strategy Department, NTT

たはプロトコル処理機能と試験シーケンス生成部機能と一体化して実現するものが多い。

本論文では、試験対象製品側に極力手を入れず、複数レイヤの試験を一体化して行う試験をねらいとしており、遠隔試験法に基づく汎用的な適合性試験システムの実現を目指している。実際の適合性試験システムのソフトウェア開発を容易にし、またシステム開発後に試験実行機能の拡張に柔軟に対応できる（プロトコル実行機能の独立性を保ち、試験実行機能の追加、変更を容易にできる）ように、また試験時の送信 PDU (Protocol Data Unit) の作成が容易であるディレールポイント方式にもとづいて、試験シーケンスを具体的に実現する方法を研究することを目的にしている。

特に、汎用的な適合性試験システムとして、種々のプロトコル製品との通信を容易にするためのプロトコル実行部と、主にプロトコルエラーに関する試験シーケンスの生成・試験をつかさどる試験論理部から成るシステムを考え、その上で実現されるべき試験論理を、双対なオートマトンの概念を用いて生成し実現する方法を述べる。

2. ディレールポイント方式による適合性試験モデル

OSI プロトコルを実装した製品が、標準プロトコルに準拠しているか否かを試験する適合性試験システムのモデルについて述べる。

適合性試験システム（以後単に試験システムと呼ぶ）と試験対象製品の一般的なモデルを、OSI の参照モデル¹⁾に基づいて図1に示す。ここで試験システムは、各試験対象製品と通信回線を介して通信を行いながら、また各製品をブラックボックスと見なしてその入出力を観測することによって試験を行う。試験システムは二つの要素から構成される。第一の要素は各試験対象製品と実際にプロトコル規定に従って正しく通信を行うために必要なプロトコル実行部である。ここで試験システムと試験対象製品の各々のプロトコル実行部の関係は、一方が送信側のとき、他方は受信側になることである。

もう一つの要素は試験論理部と呼ばれ、主にプロトコルエラーに関する試験シーケンス（すなわち試験システムのプロトコル実行部では生成できないもの）を生成・試験するものである。

試験システムにおいて、プロトコル実行部と試験論理部を結合するためのモデルを図2に示す。ここでは制御がレイヤ間インタフェースのサービスアクセスポイントで分岐され（このようなメカニズムをディレールポイントと呼ぶ）、試験論理部へ送られる。

試験対象製品の単一レイヤの試験において、試験システム側から試験の環境設定や同期制御等を行うために、試験対象製品内に試験用モジュール（例えば、上位/下位テストなど）を組み込む方法が提案されているが^{4),6),9)}、多様な異機種製品を試験する場合には、製品ごとにプログラムを準備するわずらわしさがあり、実用的な観点からは短期間に試験を行う上で大きな阻害要因となる。本論文では、このような試験用モジュールを、製品内に一切組み込まない方法で試験することとする。ただし、試験対象製品側には、製品を操作するために必要なアプリケーションプログラムやオペレーションは存在するものとする。

また各試験対象製品を試験する場合の実行単位としては、初期状態から始まり再び初期状態に戻る一つのサイクル（試験サイクルと呼ぶ）を考え、各状態遷移アークが少なくとも一回は通過する方法で求まる試験サイクルの集合に対して試験を実施することとする。

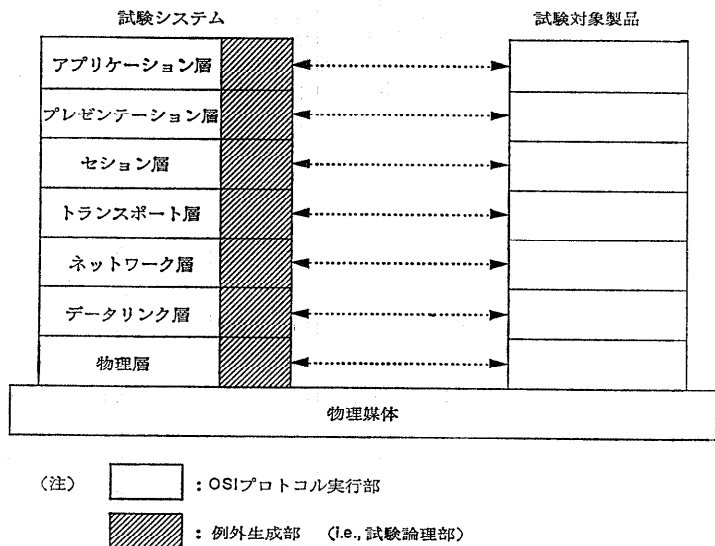


図1 OSI 参照モデルに基づく試験システムのモデル
Fig. 1 A test system model based on reference model of OSI.

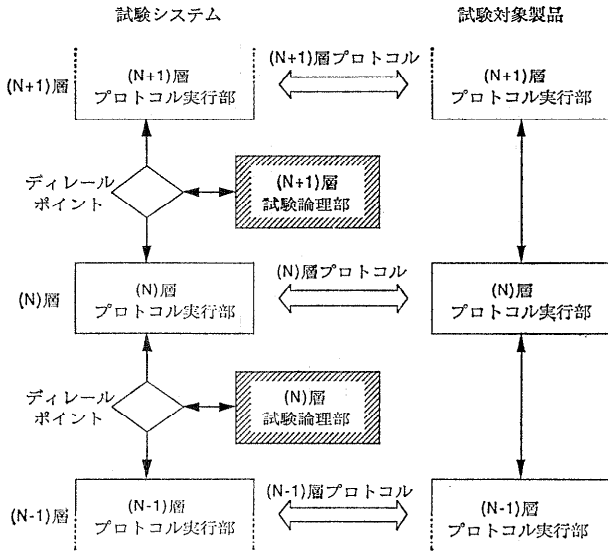


図 2 試験論理部の実現モデル
Fig. 2 Test logic part realization model.

3. 双対オートマトンによる適合性試験システムの特性化

ここでは、2章で述べたディレールポイント方式による試験システムが、試験対象製品を網羅的に試験するために実現しなければならない状態遷移構造を求める。

3.1 プロトコルモデル

図1において各レイヤは一つのエンティティから構成される。同位のレイヤのエンティティは、同位のプロトコルを用いて互いに通信する。このような各プロトコルエンティティは、次の5組から成るオートマトンによって定義される。

$$M = (Q, I, O, \delta, W)$$

ここで、

- Q: 有限な状態の集合
- I: 有限な入力シンボルの集合
- O: 有限な出力シンボルの集合
- $\delta: Q \times I \rightarrow Q$, 状態遷移関数
- W: $Q \times I \rightarrow O$, 出力関数
- $q_0: M$ の初期状態 ($q_0 \in Q$)

である。

また、Mの表現として次のような遷移グラフも用いる。各頂点をMの一つの状態に、また各アーク(遷移アークと呼ぶ)を一つの遷移に対応づけ、その遷移を引きおこす入力と遷移結果の出力によって各遷移アーク

クをラベルづけする。

本論文で考えるオートマトンは強連結で、次の二つのタイプの入力/出力シンボルを有するものと仮定する。

- (1) 外部シンボル: 通信回線を介して、同位レイヤのエンティティ間で送受信される PDU である。
- (2) 内部シンボル: (1)以外のシステム内部で生じるイベントで、抽象プリミティブ (ASP: Abstract Primitive, 例えばコンファメーション/インディケーション等) である。

このうち、内部シンボルは本試験に関して重要なインパクトを与えないから、一様に特別なシンボル“*”で代表して表すことにする。すなわち、入力集合 I、および出力集合 O は各々、

$$I = \{i_1, i_2, \dots, i_{m-1}, *\}$$

$$O = \{o_1, o_2, \dots, o_{t-1}, *\}$$

と表され、 i_j ($1 \leq j \leq m-1$)、 o_k ($1 \leq k \leq t-1$) は PDU である。明らかに、上記(1)、(2)の組み合わせより、すべての遷移アークは、四つのカテゴリ—すなわち、*/*, */o, i/*, i/o に分類される。

3.2 双対オートマトン

3.1節で述べたプロトコルエンティティを表すオートマトン M に対して、それと双対なオートマトン \bar{M} を次のように定義する。

$M = (Q, I, O, \delta, W)$ に対して、 $\bar{M} = (\bar{Q}, \bar{I}, \bar{O}, \bar{\delta}, \bar{W})$ が双対なオートマトンと呼ばれるのは、写像 $\alpha = (\alpha_Q, \alpha_I, \alpha_O)$ の下で、 \bar{M} が以下の条件を満たすときである。すなわち、

$$\alpha_Q: Q \rightarrow \bar{Q} \text{ は 1 対 1 写像}$$

$$\alpha_I: I \rightarrow \bar{I} \text{ は 恒等写像}$$

$$\alpha_O: O \rightarrow \bar{O} \text{ は 恒等写像であり、任意な } q \in Q, \text{ 任意な } i \in I \text{ および } W(q, i) = o \text{ となるある } o \in O \text{ に対して、}$$

- (I) i または o が内部シンボルの時:

M と \bar{M} の間に次式が成り立つ。

$$\alpha_Q(\delta(q, i)) = \bar{\delta}(\alpha_Q(q), \alpha_O(o)),$$

および

$$\alpha_I(i) = \bar{W}(\alpha_Q(q), \alpha_O(o))$$

(上式の関係を、図3に示す)。

- (II) i と o がともに外部シンボルの時:

特定の状態 $\bar{q}' \in \bar{Q}$ (\bar{q}' は Q の中には存在しない

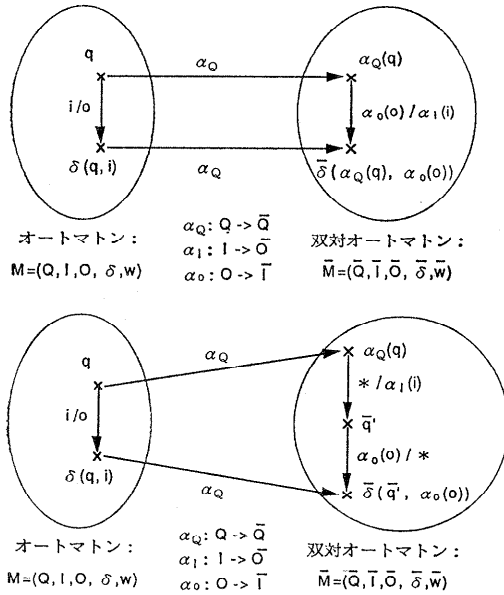


図3 オートマトン M, \bar{M} 間の双対条件
Fig. 3 The condition between M and \bar{M} .

状態) に対して M と \bar{M} の間に次式が成り立つ。
 $\bar{\delta}(\alpha_Q(q), *) = \bar{q}'$, かつ $\bar{\delta}(\bar{q}', \alpha_O(o)) = \alpha_Q(\delta(q, i))$ および $\alpha_I(i) = \bar{w}(\alpha_Q(q), *)$ かつ $\bar{w}(\bar{q}', \alpha_O(o)) = *$.

上述した双対オートマトン \bar{M} は、ちょうど M のオートマトン M に対して入力と出力の関係が逆になっている (一方が送信側のとき、他方は受信側となる関係)。このような双対オートマトンの概念は、Zafropulo によって示された duologue⁸⁾ の概念と類似なものである。また双対オートマトンは、プロトコル仕様で規定されている通信相手側の状態遷移とは異なっている。すなわち、一般にプロトコル仕様では、エラーの受信に関する遷移動作は規定されているが、エラーの送信に関する遷移動作は規定されない。試験対象製品を網羅的に試験するためには、このようなエラーの送信も試験システム側で実現しなければならないが、双対オートマトン M には必要なエラー送信動作がすべて含まれている。ただし、 M の遷移のうちでカテゴリーのタイ

プが $O/*$ (試験システムが試験対象製品から外部出力シンボル O を受信する場合の遷移) や $*/*$ (試験システムからの外部シンボルの送受信がなく、試験対象製品内の内部シンボルによる場合の遷移) に対して必要な試験対象製品側の環境設定等は、製品側で行われるオペレーションやアプリケーションプログラムによってすべて可能であるとする。

ここで試験対象製品のプロトコルエンティティで実現されるオートマトンを M とするとき、試験システムのプロトコル実行部で実現されるオートマトン (M_P と表すが、基本的には M と同一なもの) と双対オートマトン \bar{M} の関係は、一般に図4のように表される。同図において、

- (I) $M_P \cap \bar{M}$: プロトコル実行部で実現可能な双対オートマトンの遷移部分
- (II) $\bar{M} - (M_P \cap \bar{M})$: 双対オートマトンの中で試験論理部で実現しなければならない遷移部分。
- (III) $M_P - (M_P \cap \bar{M})$: 試験対象製品がオートマトン M で許されていない遷移を実行したとき、適合性試験システムのプロトコル実行部が検出される遷移部分である。

以上のことから次の命題が成り立つ。

【命題】 試験対象製品のプロトコルエンティティを表すオートマトンを M とするとき、試験システムがエラー受信時の動作を含めた M のすべての状態遷移を試験できるためには、試験システムが M と双対なオートマトン \bar{M} を実現することが必要である。

(略証) 試験システムにおいて、 M の状態遷移の中の

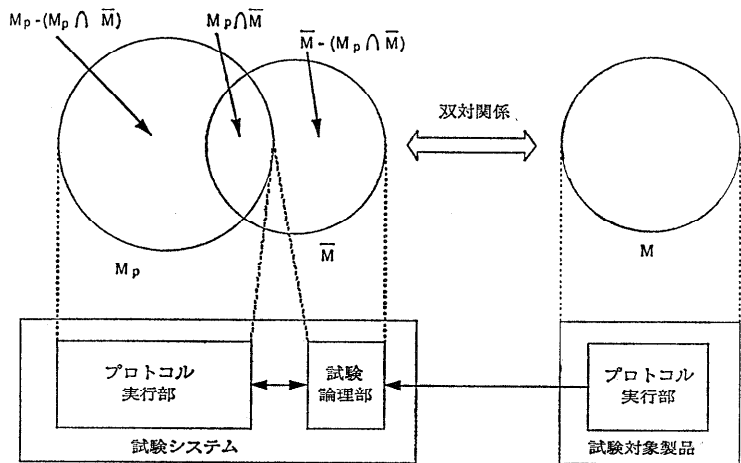


図4 試験システムの構造とオートマトン M, \bar{M}, M_P 間の関係
Fig. 4 The structure of test system and the relation among M, \bar{M} and M_P .

正常な遷移動作に対応する部分は、 $M_P \cap \bar{M}$ によって実現され、また試験対象製品側がプロトコルエラーを受信およびエラーを送信する動作に対応する部分はそれぞれ、 $\bar{M} - (M_P \cap \bar{M})$ および $M_P - (M_P \cap \bar{M})$ によって実現されることから、明らかである。

(証明終)

以後、 $M_P \cap \bar{M}$ および $\bar{M} - (M_P \cap \bar{M})$ の各々に含まれる各遷移アークを、それぞれ P-アークおよび L-アークと呼ぶことにする。

3.3 双対オートマトンの導出手順

ここでは双対オートマトンならびに L-アークの集合 (試験論理部で実現すべき遷移アークの集合) を導出する手順を示す。

ステップ 1. オートマトン M の各状態 q_i を \bar{q}_i で置きかえる。

ステップ 2. 各遷移アーク i_k/o_l を次のように置き換える。

- (I) $i_k = *$ かつ $o_l = *$ のとき、 $*/* \Rightarrow */*$
- (II) $i_k = *$ かつ o_l が外部シンボルのとき、 $*/o_l \Rightarrow$

$o_l/*$

- (III) i_k が外部シンボルかつ $o_l = *$ のとき、 $i_k/* \Rightarrow */i_k$

- (IV) i_k および o_l がともに外部シンボルのとき、 i_k/o_l を、 $*/i_k$ と $o_l/*$ に分解し、それらの遷移アーク間に新しい状態を定義する。ステップ 1 および 2 より双対オートマトン \bar{M} が得られる。

ステップ 3. \bar{M} の初期状態 q_0 を、 M_P の初期状態に対応づける ($q_0 = \bar{q}_0$)。

ステップ 4. q_0 から出発または終了する遷移パス (連続した遷移アークの列) が、 M_P と \bar{M} 間で一致するならば、これらのパスによって通過される状態 \bar{q}_i を、 M_P の対応する状態でおきかえる。またこれらのパスを、 \bar{M} から取り除く。

ステップ 3 および 4 より、L-アークの集合が求まる。《例》ISO の OSI トランスポートプロトコル (クラス 0)^{9),10)} に対する双対オートマトンを、上記の手順に従って求めると図 5 のようになる。同図 (a) のオートマトン M および M_P はそれぞれ試験システムおよび試験対象製品のプロトコル実行部に相当し、二重線アークは $M_P - (M_P \cap \bar{M})$ に対応する。また同図 (b)

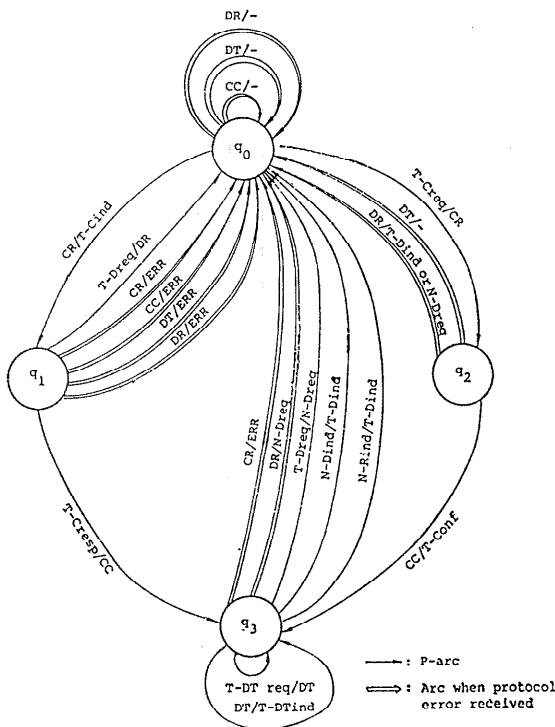


図 5 (a) オートマトン M および M_P (試験システムおよび試験対象製品のプロトコル実行部)

Fig. 5 (a) FSM (protocol implementation parts in product and the test system).

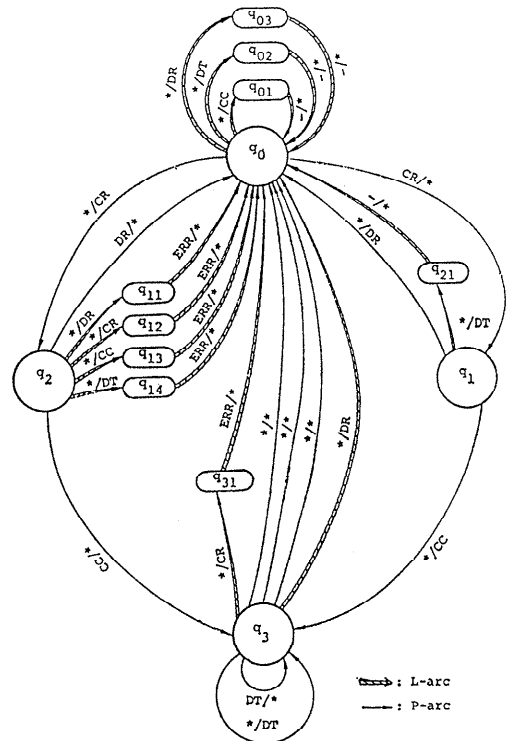


図 5 (b) 双対オートマトン \bar{M}
Fig. 5 (b) Dual FSM.

Note:

- Legend: 1) Input Events
 *T_Creq = T_Connect_Request
 *T_Dreq = T_Disconnect_Request
 *T_DTreq = T_Data_Request
 CR = Connect_Request PDU
 CC = Connect_Confirm PDU
 DT = Data_Request
 DR = Disconnect_Request PDU
 *N_Dind = Network_Disconnect_Indication
 *N_Rind = Network_Reset_Indication

2) Output Events

- *T_Cind = T_Connect_Indication
 *T_Dind = T_Disconnect_Indication
 *T_Cconf = T_Connect_Confirm
 *T_DTind = T_Data_Indication
 ERR = Error PDU
 *N_Dreq = Network_Disconnect_Request
 where "*": internal symbol
 "-": an output event means that no output is generated

3) State

- q₀: Idle
 q₁: Wait_for_TSAP_Accept_Resp
 q₂: Wait_for_TPDU_Connect_Conf
 q₃: Data Transfer

(c)

図5 OSI トランスポートプロトコル (クラス0) のオートマトンモデルと双対オートマトン
 Fig. 5 FSM model and dual FSM for OSI Transport protocol (class 0).

の双対オートマトン \bar{M} でハッチを施したアークが L-アークに当たり、実線のアークが P-アークは各々、 $\bar{M} - (M_P \cap \bar{M})$ および $M_P \cap \bar{M}$ に対応している。

4. 試験論理の導出方法

ここでは3章で求めた L-アークの集合を実際に試験論理部で実現する方法を述べる。

4.1 試験サイクルと試験理論

試験対象製品を試験する上で必要となる試験サイクルを以下に定義する。Mにおいて、初期状態 q₀ から出発し、q₀ で終わるもので途中で q₀ を通過しないパス (一つの unilogue⁸⁾ に等価) を試験サイクルと呼ぶ。特に M および \bar{M} において、すべ

ての遷移アークが少なくとも一回通過する方法で得られる試験サイクルの集合を、それぞれ P_M および P \bar{M} と書く。ここで、P_M は試験対象製品を網羅的に試験するとき、試験システムによって実現されるべき試験サイクルの集合を示す。このような P_M は、文献14)等の手法を用いて求めることができる。

Pは次の二つのカテゴリに分類される。

カテゴリ 1. 一つの試験サイクルが、P-アークのみから構成されるもので P-サイクルと呼ぶ (図6(a)).

カテゴリ 2. 一つの試験サイクル内に少なくとも一つの L-アークを含むもので L-サイクルと呼ぶ (図6(b)). また一つの L-サイクル内で、連結した L-アークの集合を L-パスと呼ぶ。

3.2節の議論より、プロトコル実行部は常に一つの P-サイクルを実行し、またすべての P-サイクルはプロトコル実行部で試験される。また L-サイクルは、試験論理部とプロトコル実行部間で実行される。すなわち、一つの L-パスは、P-サイクルのある遷移 (P-アーク) がプロトコル実行部で生じたとき、試験論理部で実行され、また L-パスが試験論理部で終了したとき、再びプロトコル実行部で P-サイクルを実行する。

次に一つの L-パスを実行する試験論理を一つのオートマトンと見なして次のように定義する。

$$T = (S, O_1 \times O_2, I_1 \times I_2, g, h)$$

ここで、S は状態の集合、O₁ × O₂ は入力シンボルの集合 (O₁ は試験システムのプロトコル実行部からの出力集合、O₂ は試験対象製品からの出力集合)、I₁ × I₂ は出力シンボルの集合 (I₁ は適合性試験システ

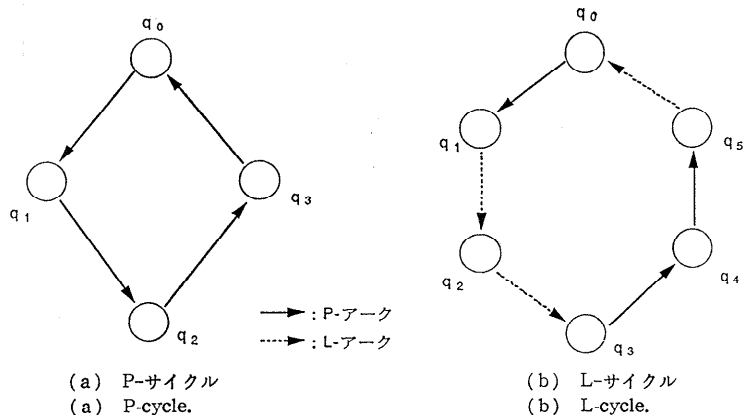


図6 試験サイクルの二つのカテゴリ
 Fig. 6 Two categories of test cycles.

ムのプロトコル実行部への入力集合, I_2 は試験対象製品への入力集合), g および h は各々遷移関数および出力関数を表す. また試験の開始状態, 終了状態, および中止状態をそれぞれ S_0 , S_f および S_z (いずれも S の元) と表す.

4.2 試験理論の基本的な生成手順

PM の一つの L -サイクルおよび一つの P -サイクルをそれぞれ C_L および C_P で表す. C_L 内の各 L -パスの開始頂点および終了頂点をそれぞれ q_s , q_e とし, 試験論理 T の任意な入力および出力シンボルをそれぞれ (o_{1j}, o_{2k}) および (i_{1l}, i_{2m}) とする. ここで, $o_{1j} \in O_1$, $o_{2k} \in O_2$, $i_{1l} \in I_1$, および $i_{2m} \in I_2$. 試験論理 T を生成するための基本的な手順を以下に示す. ただし, o_{1j} , および o_{2k} は T で同時に生じないものとし, また適合性試験システムのプロトコル実行部は正しく動作するものと仮定する.

ステップ 1. C_L 内の任意な L -パスが次の条件を満たしているかをチェックする.

(I) q_s で終了する P -アークが存在しないとき, (例えば $q_s = q_e$), C_L 内の q_s で開始する P -アークのタイプは $*/i$ であること.

(II) q_s で終了する P -アークが存在するとき, C_L 内で q_s で終了する P -アークは, C_P 内の q_s においても存在すること.

(本ステップは, 一つの L -パスを実現するために, C_P をいかに選ぶかを示す. もし条件が満たされなければ, 別の C_P が選ばれる.)

ステップ 2. P -アーク上の任意な入力シンボル (o_{1j}, o_{2k}) に対して,

(I) q_s で終了する P -アークが存在し, かつ o_{1j} または o_{2k} が q_s で終了する P -アークの入力または出力シンボルのとき,

①初期状態 S_0 を S_1 にする.

②入力/出力を, $(o_{1j}, -)/(-, o_{1j})$ または $(-, o_{2k})/(-, i_{2m})$ とする. ここで, “-” は入力および出力が存在しないことを示し, また i_{2m} は L -パス内の最初の L -アーク出力シンボルを表す.

(II) q_s で終了する P -アークが存在せず, かつ o_{1j} , または o_{2k} が q_s から出発する P -アークの入力または出力シンボルであるとき,

① S_0 を入力が $(o_{1j}, -)$ のとき S_1 または $(-, o_{2k})$ のとき S_2 にする.

②入力/出力を, $(o_{1j}, -)/(-, i_{2m})$ または $(-, o_{2k})/(i_1^+, i_2^+)$ とする. ここで i_1^+ および i_2^+ は試験を中止

するための出力シンボルを表す.

(III) o_{1j} または o_{2k} が q_s で開始または終了する P -アークの入力または出力シンボルでなく, かつ C_P 内の P -アークに対する出力シンボルであるとき,

① S_0 を S_0 のままとする.

②入力/出力を $(o_{1j}, -)/(-, o_{1j})$ または $(-, o_{2k})/(o_{2k}, -)$ とする.

(IV) o_{1j} または o_{2k} が C_P 内の P -アークの入力または出力シンボルでないとき,

① S_0 を S_2 とする.

②入力/出力を, $(-, o_{2k})/(i_1^+, i_2^+)$ とする.

(本ステップは, L -パス内の L -アークの開始をいかに行うかを述べている. 特に(I)または(II)の場合には, 適合性試験システムのプロトコル実行部は, 状態 q_s , または q_s' , q_s'' で停止している. ここで q_s' は q_s の直後の状態を, また q_s'' は q_s の直後の状態を表す.)

ステップ 3.

(I) L -パス上の入力シンボル (o_{1j}, o_{2k}) の系列に対して,

①状態 S_u を S_{u+1} にする ($u=1, 2, \dots, h-1$). ただし, $S_h = S_f$ (最後の L -アークにあたる).

②入力/出力を, $(-, -)/(-, i_{2m})$ とする.

(II) 任意の S_u および L -パス上に現れない任意の入力シンボル (o_{1j}, o_{2k}) に対して,

① S_u を S_2 とする.

②入力/出力を, $(-, o_{2j})/(i_1^+, i_2^+)$ とする.

(本ステップは, 各 L -パスを実際に試験論理として実現する方法を示す.)

ステップ 4. ステップ 3-(I)に対して,

①状態 S_{f+v} を S_{f+v+1} とする ($v=0, 1, \dots$)

②入力/出力を $(-, -)/(i_{1l}, -)$ または $(o_{1j}, -)/(i_{1l}, -)$ とする. ここで o_{1j} および i_{1l} は q_s または q_s' , q_s'' から出発し, q_e で終わる P -アーク上の入力および出力シンボルである.

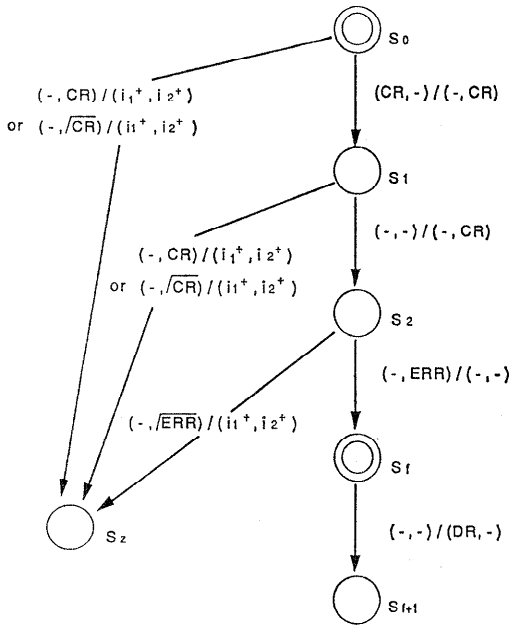
(本ステップは, 試験システムのプロトコルの実行部の状態を, q_s , q_s' , q_s'' から q_e へ移す方法を示す.)

《例》上記手順に基づいて, 図5で示したプロトコルに対する試験論理 T の生成例を図7に示す. ここで P -サイクル (C_P) および L -サイクル (C_L) としては次のものを考える.

$$C_L: (q_0) */CR.**/CR.**/CR*ERR/(q_0)$$

$$C_P: (q_0) */CR*CC/*DT/**/(q_0)$$

また上記に示した試験論理 T は, T の試験が開始する前にあらかじめキューに蓄えておき, T が開始され



- : 入/出力シンボルが存在しないことを示す
 S₀: 試験の開始状態
 S₅: 試験の終了状態
 S₂: 試験の中止状態
 / \bar{a} : シンボル a 以下のシンボル

図 7 図 5 に対する試験論理の例
 Fig. 7 A test logic example for Fig. 5.

るとき、試験システムの制御部によって T はキューから順に取り出され、試験論理部に設定されて試験を開始する。

5. 試験論理の評価と実験結果

ここでは、4章で示された試験論理の生成手順によって作られる試験論理が、実際の試験において果たす役割を状態遷移アークの実現の観点から評価するとともに、本手順が実際の試験システムに適用されて行われた適合性試験の実験結果を述べる。

5.1 試験論理の評価

図 4 の試験システムを構成する次の三要素の中で、試験論理部が果たす役割を、各々が担当するオートマトンの状態遷移の観点から評価する。

- $M_P \cap \bar{M}$: 正常な遷移動作をつかさどる部分.
- $\bar{M} - (M_P \cap \bar{M})$: 試験論理部によって実現されるプロトコルエラーの送信動作をつかさどる部分.
- $M_P - (M_P \cap \bar{M})$: プロトコルエラーの受信動作をつかさどる部分.

図 5 の OSI トランスポートプロトコル (クラス 0) に対して、試験システムの各要素が実現する状態遷移アーク数を比較すると、

$$(M_P \cap \bar{M}) : \{\bar{M} - (M_P \cap \bar{M})\} : \{M_P - (M_P \cap \bar{M})\} = 11 : 10 : 10$$

となり、試験論理部は全体の約 30% に当たる遷移部分 (他のプロトコルの場合文献 18) では、約 42%) を占めている。このことは、4.2 節で述べた L-サイクルの試験に当たって、試験システムのプロトコル実行部が動作する環境 (P-アーク) 下で、各種試験論理 (L-アーク) を組み合わせて試験する方法 (4.2 節で示した試験論理の生成手順) を用いることにより、残りの 70% を占める P-アークの遷移部分の共通化が図られるため、両アークを一体化して各々個別に試験サイクルを実現する場合に比べて、効率的に試験が実施できることを示している。

5.2 実験結果

我々が開発した適合性試験システム (PROVES)^{17), 18)} において、本論文で提案した手順を適用して試験論理を作成した結果および、実際にコンピュータや端末など五つの試験対象製品を用いて行った実験結果より、次の知見を得た。

- (1) 生成手順によって実現される各試験論理 (L-アーク) は、プロトコル実行部 (P-アーク) の環境化で動作することから、試験論理で実現すべき試験範囲が局所化されることにより、プログラムの作成が容易となり、実験に用いた試験論理は、Pascal 言語で平均約 49 ステップ程度の規模で実現できること、また PROVES を用いた実験では各試験対象製品の試験において、112 個の試験サイクルが用いられたが、生成手順による試験論理 (L-アーク) を P-アークを組み合わせる方法で効率的に試験サイクルが実現でき、各試験サイクルは平均で約 2 分内 (試験論理の環境設定や製品側でのオペレーション時間等に要する時間を含む) 行えることを確認した。
- (2) これらの試験サイクルを用いた試験の結果、各試験対象製品のプロトコル実装において、検出されたプロトコル誤りは、平均 7 件でありその誤りのうち、54% がプロトコル仕様の解釈誤りに基づくものであった。これらの誤りは製品開発側では検出が難しいものであり、適合性試験の有効性を示すものである。
- (3) 遠隔試験法による適合性試験では、試験システム

側と試験対象製品側の間で、一般に状態の不一致が生じる(例えばカテゴリーのタイプが $O/*$ の遷移で、試験対象製品の固有な実装状態に依存して定まるビジー状態時等の外部出力シンボル O を、試験システムが受信する場合の遷移)¹⁸⁾ことがあるが、今回の実験においては、製品側でのアプリケーションプログラムやターミナルオペレーションによって製品内部のイベントを疑似的に生成(ビジー条件等)させる方法で解決を図り、結果としてプロトコルの準拠性を試験する上で必要な基本的な部分(すなわち、 $*/*$ タイプの遷移アーク部分を除いた、約90%の遷移アーク)が試験可能であることを確認した。

6. おわりに

本論文では、試験システムにおいて必要な試験論理を体系的に生成するための手順を、双対オートマトンの概念に基づいて示した。試験論理は全体の試験の中で約30~40%を占めること、またディレールポイント方式における試験論理の生成法により、60~70%の遷移アークを共通的に用いて試験サイクル(L-サイクル)を実現する方法で効率的な試験が行えることを、実際の試験システムを用いた実験により確認した。

今後に残された問題としては、現在の1対1の通信をベースとした適合性試験から、トランザクション処理、ディレトリ、ネットワーク管理などにおいて現れる n 対 m の通信形態にも適用可能なマルチパーティ試験に拡張していくことがあげられる。

謝辞 最後に本研究において御指導頂いた慶応大学 苗村憲司教授ならびに、NTT 情報通信網研究所知識処理研究部河岡司部長に感謝する。

参 考 文 献

- 1) Reference Model of Open System Interconnection, ISO/DIS 7498 (1982).
- 2) ISO/IEC JTC 1/Sc 21: Information Technology—Open Systems Interconnection—Conformance Testing Methodology and Framework, ISO DIS 9646 (1991).
- 3) Weaving, K.: Euronet Reference and Test Center, *Comp. Comm.*, Vol. 3, No. 5, pp. 221-223 (1980).
- 4) Nightingale, J.S.: Protocol Testing using a Reference Implementation, *Proceedings of the IFIP WG 6.1 2nd International Workshop on Protocol Specification, Testing, and Veri-*

fication, pp. 513-520 (1982).

- 5) Sarikayi, B. and Bochmann, G.V.: Some Experience with Test Sequence Generation for Protocol, *Proceedings of the IFIP WG 6.1 2nd International Workshop on Protocols Specification, Testing, and Verification*, pp. 555-567 (1982).
- 6) Rayner, D.: A System for Testing Protocol Implementations, *Comp. Net.*, Vol. 6, No. 6, pp. 383-395 (1982).
- 7) ISO/CCITT: Draft Transport Protocol Specification (Dec. 1981).
- 8) Zafropulo, P., et al.: Protocol Validation by Duologue-matrix Analysis, *IEEE Trans.*, Vol. COM-26, No. 8, pp. 1187-1194 (1978).
- 9) 齊藤, ほか: オートマトンモデルによる HDLC プロトコル製品検証の一方法, 信学論 (D), Vol. J 63-D, No. 8, pp. 642-649 (1980).
- 10) Rayner, D.: OSI Conformance Testing, *Computer Networks and ISDN System*, Vol. 14, pp. 79-98 (1987).
- 11) Sarikaya, B., Bochmann, G.V. and Cerny, E.: A Test Design Methodology for Protocol Testing, *IEEE Trans. on Software Engineering*, Vol. SE-13, No. 5, pp. 518-531 (1987).
- 12) Sabnani, K. and Dahbura, A.: A Protocol Test Generation Procedure, *Computer Networks and ISDN System*, Vol. 15, pp. 285-297 (1988).
- 13) Sidhu, D.P. and Leung, T.: Formal Methods for Protocol Testing: A Detailed Study, *IEEE Trans. on Software Engineering*, Vol. 15, No. 4, pp. 413-426 (1989).
- 14) Sarikaya, B.: Conformance Testing: Architectures and Test Sequences, *Computer Networks and ISDN System*, Vol. 17, pp. 111-126 (1989).
- 15) Sidhu, D.P.: Protocol Testing: The First Ten Years, The Next Ten Years, *Protocol Specification, Testing and Verification, X*, pp. 47-68 (1990).
- 16) 田中, 清水: OSI 実現とその課題 (V) 適合性試験, 情報処理, Vol. 31, No. 1, pp. 89-97 (1990).
- 17) Kawaoka, T., et al.: A Method for Verifying Layered Protocol Products and Its Application Products, *Proc. ICCS*, pp. 379-384 (1980).
- 18) Araki, T., Takada, K. and Yositate, S.: A Test Logic Generation Method for Layered Protocol Implementations, *COMPSAC*, pp. 356-365 (1983).

(平成4年11月11日受付)

(平成5年4月8日採録)

**荒木 哲郎** (正会員)

昭和 23 年生。昭和 46 年福井大学工学部電気工学科卒業。昭和 51 年東北大学大学院博士課程修了。同年日本電信電話公社入社。横須賀電気通信研究所，NTT 情報通信処理研究所を経て，現在福井大学工学部電子工学科勤務。プロトコルの設計・試験，自然言語処理の研究に従事。東北大学工学博士。電子情報通信学会，人工知能学会，IEEE 各会員。

**高橋 修** (正会員)

1975 年北海道大学工学研究科修士課程修了。同年日本電信電話公社(横須賀電気通信研究所)入社。以来通信プロトコルの開発，適合性試験，標準化，導入推進業務に従事。現在 NTT ネットワーク高度化推進本部ネットワーク部主幹技師。

**吉武 静雄** (正会員)

昭和 42 年広島大学工学部電気工学科卒業。同年日本電信電話公社入社。電気通信研究所，横須賀電気通信研究所において大型電子計算機のオペレーティングシステム，ネットワークアーキテクチャ，プロトコルのコンFORMANCE 試験などの研究に従事。その後ネットワーク開発センター，ネットワーク高度化推進本部において通信プロトコルの標準化を担当。平成 5 年 NTT ソフトウェア株式会社入社。現在，分散応用ソフトウェアの開発に従事。電子情報通信学会会員。