

通信システムの段階的な試験のための試験系列 自動生成手法とその実現†

佐藤 文明^{††} 宗 森 純^{†††}
勝山 光太郎^{††} 水野 忠 則^{††}

大規模で複雑なサービスを提供する通信システムの開発には、多くの費用と時間がかかり、開発の効率化が望まれている。本論文では、通信システムの試験工程の期間短縮と費用削減のために、試験系列を段階的に生成する手法と、それを實現した試験系列自動生成ツール“TENT”について述べる。通信システムの試験工程は、試験の詳細さに応じていくつかの段階があり、さらに1つの段階のなかにいくつかの試験項目が存在する。TENTは、試験工程内の各段階に適した詳細度と、各試験項目に適した系列を生成できるように以下の点について考慮されている。(1) MT法、TT法そしてSW法という複数の生成手法を用いて生成する。(2)上記の各手法に対して、通信システムの仕様のフェーズ別、機能別の試験を可能とするために、系列の切り出し機能を付加する。また、TENTは、国際標準に沿ったツールとするために、CCITTが勧告する形式的仕様記述言語SDLのサブセットで記述された仕様を入力とし、試験系列をISOが規定する試験スイート表記法TTCNを使って出力する。このような方針でTENTを實現し、実際に通信プロトコルに適用して評価を行った。その結果、実際の通信プロトコルに適用可能であり、通信システムの試験に使いやすく、ドキュメント化しやすい試験系列が生成できることを確認した。

1. はじめに

通信ソフトウェア開発においては、その試験に要する期間や費用の削減が重要な課題となっている。試験には、試験仕様の開発、特に試験系列を要求仕様から抽出する作業が必要である。現在、この作業は専門家によって手作業で行われている。そのため、試験仕様の誤りや試験漏れ、試験仕様の開発費用や期間が問題となっている。

通信システムの試験には、適合性試験、性能試験、相互接続試験や堅牢性試験などがあるが、まず通信システムが要求仕様を満足しているかどうかを試験する適合性試験が必要となる¹⁾。適合性試験には、段階的に基本接続試験、機能範囲試験、動作試験、適合性決定試験がある²⁾。

専門家は、このような試験行程の各段階や項目に適した系列を、自分の知識と経験を使って要求仕様から抽出している。しかし、試験工程に要する期間やコストの観点、および試験系列の網羅性や品質の観点から

この作業を自動化する研究が行われてきた。試験系列を要求仕様から生成するアルゴリズムについて、いくつかの手法が提案されている³⁾⁻⁵⁾。また、この手法を實現した生成システムについてもいくつかの報告がある^{6),7)}。しかし、従来の生成システムは、試験の実体に即したのではなく、さらに実用化研究が必要なものであった。

例えば、文献7)の生成システムは、入力として仕様記述言語Estelle⁸⁾を用いている。さらに、生成手法として、文献3), 4)および5)の手法が採用されている。そのため、試験系列は網羅性のある系列が自動的に得られる。しかし、このシステムは、それぞれの生成手法が単純に實現されただけであり、試験工程内の各段階や、試験目的を意識したものではない。そのため、生成手法の特性を活かした試験系列の適用や、試験の効率的な遂行に関する検討は行われていなかった。

本論文では、試験工程内の各段階に適用される試験系列の特性の検討、および効率的な試験を行うための試験系列生成手法の提案を行っている。以下、第2章では、通信システム試験系列に要求される条件と、要求仕様から試験系列を生成する既存の生成手法の問題点について述べる。次に第3章では、我々が開発した通信システム試験系列生成ツール“TENT”(Test sequence generation tool)について、その設計方針と構成について述べる。第4章では、実際にこのツ

† An Implementation of Test Sequence Generation Method for Stepwise Testing of Communication Systems by FUMIAKI SATO (Information Systems and Electronics Development Laboratory, Mitsubishi Electric Corp.), JUN MUNEMORI (Department of Information and Computer Science, Faculty of Engineering, Kagoshima University), KOTARO KATSUYAMA and TADANORI MIZUNO (Information Systems and Electronics Development Laboratory, Mitsubishi Electric Corp.).

†† 三菱電機(株)情報電子研究所

††† 鹿児島大学工学部情報工学科

ルを通信プロトコルに適用した結果について述べ、第5章では、その結果について考察する。第6章は本論文のまとめである。

2. 試験の分類と試験系列に要求される条件^{1),2)}

通信システムの仕様は、状態遷移表をはじめとした有限状態機械 (FSM) モデルで記述されることが多い。そこで、試験系列生成アルゴリズムも、FSM モデルに基づく手法が多く開発されている。FSM モデル M は、5つ組

$$M = \langle S, I, O, f, g \rangle \quad (1)$$

で定義され、状態集合 S 、入力集合 I 、出力集合 O と、状態と入力から状態遷移を起こす状態遷移関数 f と状態と入力から出力を決定する出力関数 g とから構成される。試験系列は、この出力関数と状態遷移関数が正しく実現されていることを検証する必要がある。

また、通信システムの試験では、観測される試験対象システムの中味はブラックボックスであり、システムの入出力からシステムが要求仕様に適合しているかどうかを判定する必要がある。

2.1 試験の段階による分類

ここでは、通信システムの適合性試験を例として、段階的に行われる試験系列について考察する。通信システムの適合性試験において、実際に試験対象を動作させて行う試験は、通常、次の2段階になる¹⁾。

(1) 基本接続試験

基本接続試験は、試験の最も初期の段階で行われる試験で、通信システムが基本的な接続機能を持っており、次に行われるさらに詳細な動作試験が可能であることを確認する。この段階では、詳細な試験は必要でないため、試験系列は要求仕様の基本的な部分を網羅するだけで十分である。一方、適用回数は試験の初期の段階であることから最も多くなるので、短い系列が要求される。

(2) 動作試験

動作試験では、基本接続試験によって基本的な接続は可能であることが確認された通信システムに対して、要求仕様に規定されたすべての動作が正しく実現されているかどうかを確認する。動作試験は、その詳細さの度合から、各状態の各イベントに対する状態遷移の確認が少なくとも一度は行われる“網羅試験”と、状態遷移の前後の状態の詳細な確認を含む“徹底試験”とに分けることができる。

表 1 試験段階による試験系列の特徴

Table 1 Characteristics of test sequence on test stage.

試験の種類	適用順序	複雑さ	使用回数	系列の条件
基本接続試験	1	小	多	簡潔な系列 ある程度の網羅性
動作試験	網羅試験	2	中	十分な網羅性
	徹底試験	3	大	十分な網羅性 誤り検出能力の 高い系列

網羅試験と徹底試験に使われる試験系列の違いは、誤りの検出能力とその系列の長さである。

上記の各試験に要求される試験系列の特徴を表1にまとめて示す。

2.2 試験の範囲による分類

動作試験は、試験の目的や試験の対象とする機能範囲を試験項目として設定して行われる。試験の項目別の分類としては、その試験対象領域に着目したフェーズ別試験、機能に着目した機能別試験がある。通信システムの要求仕様では、通信システムの処理を大きくいくつかのフェーズに分割して、フェーズごとにまとまった処理を定義している。例えば、相手の通信システムとの論理的な通信路を確立する“コネクション確立フェーズ”、その通信路を使ってデータの送受信を行う“データ転送フェーズ”、相手の通信システムとの通信路の解放を行う“コネクション解放フェーズ”等である。これらのフェーズに限定された試験がフェーズ別試験である。さらに、1つのフェーズ内でも、正常処理、異常処理（異常データが送られてきたので再送を要求する等）、そして例外処理（電源異常やコネクションが突然切断された等）といった様々な機能が存在する。また、複数のフェーズに跨った機能の分類も考えられる。これらの機能に限定した試験が機能別試験である。これらの項目別の試験においては、なるべく試験項目内に限定された試験系列が必要である。

表 2 試験範囲による試験系列の特徴

Table 2 Characteristics of test sequence on test area.

試験の種類	分割の範囲	系列の条件
フェーズごとの試験	フェーズ (状態の集合) 単位	なるべくフェーズ内を通過する系列
機能ごとの試験	機能 (状態遷移の集合) 単位	なるべく機能の範囲内を通過する系列

このような試験の範囲によって分類された試験の必要な試験系列の特徴を表2にまとめる。

2.3 既存の生成手法と適用上の問題点

既存の試験系列生成手法として代表的なものは、TT法³⁾、PW法⁴⁾、SW法⁹⁾、そしてDS法⁵⁾である。TT法は、基本的にFSMのすべての状態遷移を網羅する系列を求める手法である。しかし、単純に状態遷移を網羅するだけの試験系列では、状態遷移関数の実現誤りを検出しない系列となる場合がある²⁾。

一方、PW法、SW法およびDS法は、状態遷移の前後の状態を、入出力系列から推定するための系列として、判定系列¹⁰⁾、あるいは特性集合 W ¹¹⁾と呼ばれる系列の集合を試験系列の一部に含んでいる。判定系列は、1つの系列であり、ある状態に対してこの系列を与えると、その出力系列から状態が判別できるという特性を持っている。しかし、FSMモデルの性質によっては、存在しない場合がある。特性集合 W は、試験系列の集合であり、その系列をある状態に対して適用すると、その出力系列集合からその状態が判別できるという性質をもつ。また、FSMモデルが最小(冗長な状態がない)ならば、必ず存在するという性質を持っている。PW法、SW法、そしてDS法は、TT法と比べると状態遷移関数の実現誤りにも対応できる試験系列が生成できる。しかし、試験系列の長さは、TT法と比較して非常に長くなりやすい性質を持っている。

このように試験系列生成手法には様々な特性があり、各試験に適した検出能力を持つ試験系列を、1つの試験生成手法によって生成することは困難である【問題点1】。また、試験系列が対象とするフェーズや機能範囲に応じて生成されるためには、生成アルゴリズムに試験系列の切り出しアルゴリズムの拡張が必要となる【問題点2】。

また、上記の生成手法は、すべての状態遷移を網羅するとか判定系列を含む等の条件でなるべく短い系列とするために、1本の連続した入力イベント系列となっている。このような系列は、誤りが系列の後半で検出されると、再試験でその状態まで系列を繰り返し適用する必要がある。効率的な試験を行うには適さない【問題点3】。また、試験項目別の試験を行うことも困難であり、試験の結果の解析や、ドキュメント化にも適していない【問題点4】。

また、上記の3手法の中で、最も簡潔な系列

が生成されるのはTT法であるが、基本接続試験ではTT法ほどの網羅性は必要ない。基本接続試験用のもっと簡潔な試験系列が望まれている【問題点5】。

3. 試験系列生成ツール TENT

ここでは、第2章で論じた試験系列の条件と生成手法に関する5つの問題点を踏まえて、通信システム試験系列生成ツールTENTの設計方針、試験系列生成手法、そしてモジュール構成について述べる。

3.1 設計方針

試験工程のモデルを図1に示す。試験工程は、試験の詳細度に応じて複数の試験段階に分割される。それぞれの試験段階は、次の段階のための予備試験となっている。最終的には、最後の段階の試験を効率的に通過することが目標である。

それぞれの試験段階では、いくつかの試験範囲、あるいは試験目的に応じた1つ以上の試験系列が用意される。

TENTは、これらの試験段階と試験項目に応じた試験系列を生成する必要がある。TENTの設計方針としては、次の点を考慮した。

(1) 各種の試験に対応できる系列

通信システムの試験は、基本接続試験や動作試験、動作試験におけるフェーズ別試験や項目別試験などの、多くの段階別、目的別試験が存在する。それぞれ、試験の目的や詳細さ、適用回数が異なっている。そのため、専門家が要求仕様から抽出する試験系列は、試験の種類に応じた試験系列となる。我々は、系列生成アルゴリズムの種類と特性を考慮して、各種の試験に対応できる生成ツールとすることが重要と考え

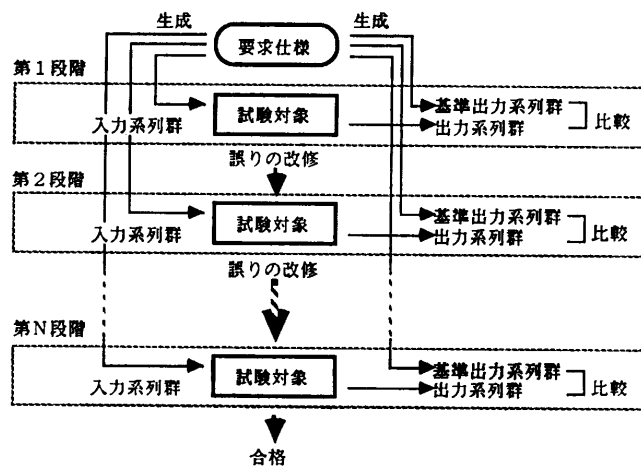


図1 段階的試験のモデル

Fig. 1 Model of stepwise testing.

た。

(2) 標準化された記述方法の採用

試験系列生成ツールの入力や出力は、通信システムの要求仕様や試験仕様の一部である。そのため、ツール固有の記述方法やデータの表現方法を持つことは避けて、なるべく広く使われている標準の記述を用いることが重要である。

(3) 品質の良い試験系列の生成

通信システムの試験の品質を良くするために、通信システムの FSM モデルに基づく試験系列生成アルゴリズムを用いる。特に、試験系列に状態の存在の確認を含んだ、誤り検出機能の高い試験系列生成アルゴリズムを開発し採用する。また、基本接続試験のような簡単な試験に対する試験系列においても、各状態へ到達できて動作試験が可能であるという確認はできる系列を生成する手法を開発し採用する。

3.2 生成手法

第2章で述べた目的に応じた試験系列を得るために、我々は次のような生成手法を開発した。

3.2.1 複数の手法による生成機能

試験系列は、生成に用いるアルゴリズムにより、生成される系列の誤り検出能力や、系列長が異なってくる。そのため、我々は、試験目的に応じた試験系列生成ができるように、次のような複数の試験系列生成手法を採用した。

(1) MT (Modified TT) 法—基本接続試験用

MT 法は、ここで新たに提案する生成手法である。TT 法は、すべての状態に規定された遷移をすべて通るような系列を求めるものであった (図 2 (a))。しかし、基本接続試験においては、すべての遷移を確認する必要はなく、より簡潔で、しかも 3.1 節で述べたようなある程度の網羅性を持つ系列が必要である。そこで、MT 法では、すべての状態遷移を対象とするの

ではなく、すべての状態に到達できることを確認する系列を生成する。MT 法、すべての状態を一通り通過するような、なるべく短い系列 (図 2 (b)) を求める方法である。MT 法の試験系列生成の手順は、付録 1 に述べる。

(2) TT (Transition Tour) 法³⁾—網羅試験用

TT 法は、現在広く用いられている生成方法であり、標準的な試験系列生成手法であると考えられる。状態遷移表のすべての欄を、なるべく短い手順で網羅するような試験系列を生成する。ただし、試験の範囲に限定した試験系列を得るために、状態遷移のほかに、フェーズ情報や機能分割の情報を系列生成時に参照するように拡張されている。

(3) SW (Single transition checking method using W set) 法⁹⁾—徹底試験用

SW 法は、特性集合 W を通信システムの状態の判定のために用いた試験系列生成手法である。SW 法は、DS 法と比較して、試験系列長の上限值は長い。PW 法と比べると、試験系列長の上限値は PW 法と同程度であるが、個々の遷移の試験を基本としている SW 法の特性を使った試験系列の短縮法¹²⁾を適用すると、PW 法より短い場合が多い。SW 法によって生成された系列は、状態を確認しながら試験できるので、出力関数と状態遷移関数の実現の誤りをともに検出できる。SW 法は、MT 法、TT 法と比べて、系列長は長くなる。

SW 法の試験系列は、各状態に対して、特性集合の要素系列を 2 回ずつ与える状態存在確認用の L_s 系列と、各状態に各入力イベントを与え状態遷移をさせた後特性集合の要素系列を与える、状態遷移確認用の L_t 系列に分けて生成する⁹⁾。

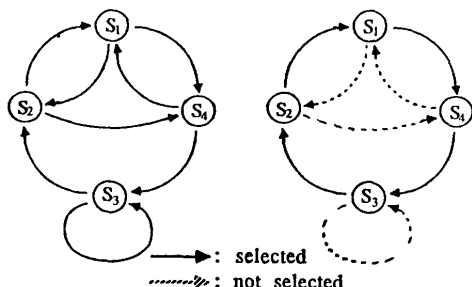
表 3 に、我々が採用した試験系列生成手法の特徴をまとめて記述した。

3.2.2 系列の分割生成機能

(1) フェーズ分割機能

通信システムの状態はフェーズごとにいくつかのサブセットに分割される。図 3 の例に示すように、それぞれの状態のサブセットについて試験系列を生成することによって、試験対象以外のフェーズを通過することの少ない試験系列を求めることができる。

例えば、TT 法や SW 法では、状態遷移の単位で試験系列の生成を行うため、状態遷移が他のフェーズに出てしまう場合、そこで一旦試験系列を打ち切って、改めてそのフェーズの別の状態から開始するよう



(a) TT 法 (b) MT 法

図 2 MT 法によって選択される状態遷移

Fig. 2 State transition selected by MT method.

表 3 TENT で用いている生成手法の特徴
Table 3 Characteristics of generation methods of TENT.

手 法	MT 法	TT 法	SW 法
系列長の上限値 (オーダ)	$2(m-1)^2$ $(O(m^2))$	$mn+(mn-1)(n-1)$ $(O(m^2n))$	$2mwv+mw^2(m-1)+m(nw+nv)$ $+mnw(m-1)$ $(O(m^2n))$
試験系列の誤り検出能力	基本動作の誤り網羅性はないが、すべての状態を通る系列であるために、状態遷移が起こることと、各状態に少なくとも1つ以上の出力関数を実現されていることを確認する	出力関数のすべての誤りと、ほとんどの遷移関数 遷移関数の誤りは検出しない場合もある	出力関数と、遷移関数のすべての誤り
系列生成のためのFSMの条件	強連結	強連結 完全定義	強連結 完全定義 最小

m: 状態数, n: 入力数, w: W 集合の要素, v: W 集合の要素系列長の総和

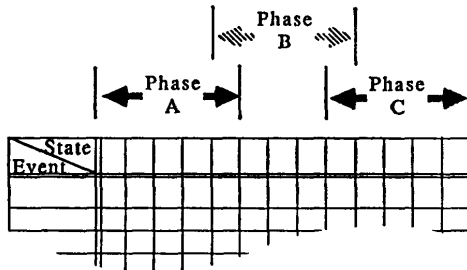


図 3 遷移表のフェーズ分割の例
Fig. 3 Example of phase division of transition table.

State Event	Phase X				
		n	n	n	e
	n	n	e	n	e
	e	r	n	e	e
	e	e	r	n	n
	e	e	r	r	r
	e	e	e	e	r
	e	e	e	e	r

n: 正常処理機能 e: 異常処理機能 r: リカバリ機能

図 4 遷移表の機能分割の例
Fig. 4 Example of functional division of transition table.

な系列を求めることにする。

(2) 機能分割機能

1つのフェーズ内においても、正常処理、例外処理、異常処理やその他の目的別の処理が含まれてい

る。これらの処理を対象とした試験系列の生成を行う方法としては、それぞれの目的に応じたフラグを状態遷移に付けておくことにする(図4)。TT法やSW法では、系列生成時にそのフラグを基準として対象とする状態遷移を識別することによって、目的別の試験系列を生成できるようになる。

3.3 生成ツールの構成

TENT は、4つのツール群とユーザインタフェース部分で構成されている。TENTを構成するツールには、CCITT(国際電信電話諮問委員会)が勧告する仕様記述言語SDL¹³⁾のテキスト表記から、FSMの状態遷移表を生成するSDL・状態遷移表トランスレータ(以下単にトランスレータ)、状態遷移表を編集したり、機能分割情報を入力する状態遷移表エディタ、フェーズ情報を入力するフェーズ情報エディタ、そして状態遷移表から実際に試験系列を生成する試験系列ジェネレータ(以下単にジェネレータ)がある。TENTの内部構成を図5に示す。

SDLのほかに、仕様記述言語としてはLOTOS¹⁴⁾とEstelle⁸⁾がISO(国際標準化機構)で標準化されている。しかし、SDLは従来から通信技術者が採用している状態遷移表と同じ拡張有限状態機械(EFSM)に基づいている点と、他の言語に比べて広く使われている点からこのツールで採用している。

試験仕様の開発者(ユーザ)は、SDL用のグラフィックエディタ¹⁵⁾や、テキストエディタ等によって作成されたSDL仕様から、トランスレータを使って状態遷移表に変換する。状態遷移表のデータに対し

て、ユーザは試験対象とする機能ごとに機能分割情報を各遷移に対するフラグとして入力する。状態遷移表エディタは、機能分割情報を、入力単位、出力単位、および状態単位で入力し編集することができる。さらに、試験の範囲を限定するために、試験のフェーズ情報を入力する。状態遷移表と、機能分割情報およびフェーズ情報から、ジェネレータによって試験系列を生成する。ジェネレータは、試験の段階を選択することによって、適切な生成方式を選択し試験系列を生成する。

トランスレータは、SDL と状態遷移表との表現方法の変換、およびデータ構造の変換を行う。SDL と状態遷移表は、共に基本概念として拡張 FSM (EFSM) モデルに基づいているが、表現力が異なるため、ループ表現等のいくつかの制約を設けて変換を行っている。また、SDL には、暗黙の動作（定義されている以外の入力があったときの動作）の定義をするための構文を追加している。トランスレータは、着目する状態にすべての入力が定義されていない場合、定義されていない入力についての動作を暗黙の動作定義によって補間する。また、条件分岐に着目して、拡張 FSM の状態遷移表から FSM の状態遷移表への変換も行っている。

ジェネレータは、状態遷移表から FSM モデルに基づく生成手法によって試験系列を生成する。ジェネレータのモジュール構成を図 6 に示す。ジェネレータは、生成方式セレクタを通して、MT 法、TT 法および SW 法の各生成モジュールを起動する。各生成モジュールでは、サブモジュールとして、リセット系列生成（到達性検証）モジュール、および特性集合（W 集合）生成モジュールを含んでいる。

リセット系列生成モジュールでは、試験対象システムを初期状態に遷移させたり、初期状態から試験対象の状態に遷移させる系列（リセット系列）を生成する。このモジュールでは、状態遷移表をグラフとしてとらえて、Dijkstra 法¹⁶⁾による最短パス検出法を用いている。さらに、リセット系列生成時に、与えられた状態遷移表の到達性の検証を行う。特性集合生成モ

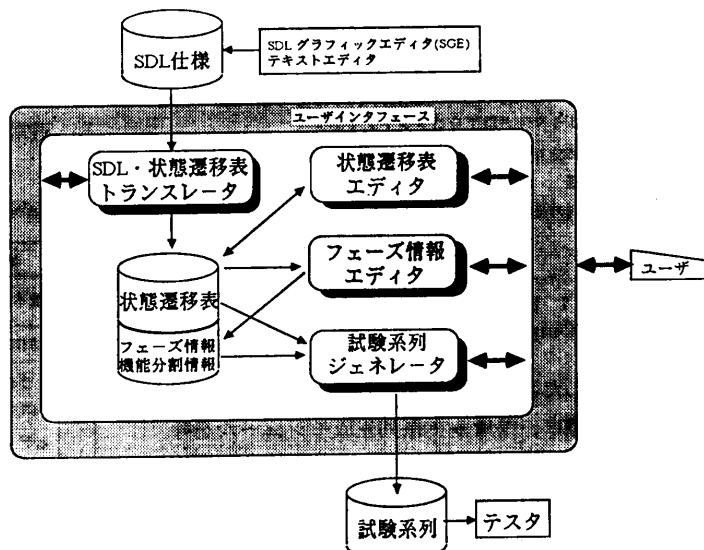


図 5 TENT の構成
Fig. 5 Internal structure of TENT.

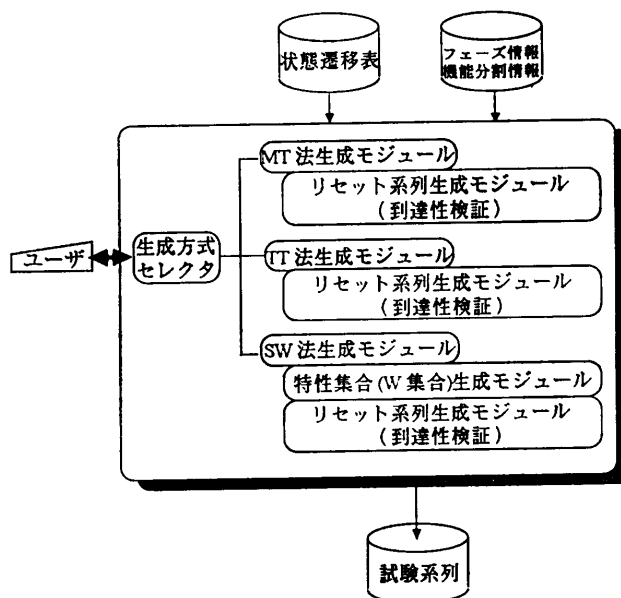


図 6 ジェネレータのモジュール構成
Fig. 6 Modules of sequence generator.

ジュールでは、状態集合を入力に対する応答で分類する等価分割の手順で分割し、分割の過程でその根拠となった入力を逆に追跡することによって特性集合を生成する¹¹⁾。同時に、状態遷移表の仕様の冗長性をも検証している。

MT 法生成モジュールは、リセット系列生成モジュールを最短路検出のために使用する。TT 法生成モジュール、および SW 法生成モジュールでは、試

<テスト> <方向> <イベント名> [<デリミタ><コメント>]

テスト :- L(下位テスト) | U(上位テスト)
 方向 :- ?(テストへの入力) | !!(テストからの出力)
 デリミタ :- # (以下をコメントとする)
 コメント :- <状態><系列の種類>
 状態 :- State-状態名
 系列の種類 :- reset (リセット系列) |
 w (W集合の要素系列) |
 _ (無し) (通常のイベント)

図 7 試験シーケンスの出力形式

Fig. 7 Output format of test sequence.

試験範囲に限定した系列を生成するためリセット系列が必要になった際に、リセット系列生成モジュールを呼び出す。SW 法生成モジュールでは、さらに特性集合生成モジュールを呼び出して、試験系列に特性集合を組み込んでいる。

ジェネレータから生成される出力形式は、図 7 のようになっている。記述項目はテスト (観測点)、イベントの方向、イベント名とコメントである。この形式は、ISO が規定している TTCN¹⁷⁾ の動作部形式に準拠している。コメントには、被試験対象がとるべき状態名と、系列の性質 (特性集合の系列やリセット系列など) を記述している。これによって、試験を行っている際にエラーが発生したときの解析情報が得られる。

4. 実際のプロトコルへの適用

段階的な試験に対して、試験系列生成手法を実現した TENT の適用性について評価するために、TENT を実際のプロトコルに適用した。対象としたプロトコルは、ISO が規定するトランスポートプロトコル・クラス 2¹⁸⁾ と、CCR¹⁹⁾ (コミットメント/同時実行性/回復制御) プロトコルである。トランスポートプロトコルは、状態数が 7 個であり、小規模の状態遷移表である。CCR は、状態数が 30 個であり、中規模の遷移表である。

トランスポートプロトコルでは、以下のような目的別に試験系列をそ

れぞれ生成した。なお、ここでフェーズ分割は、プロトコル仕様に記述されているように、コネクション確立・解放フェーズとデータ転送フェーズに分割している。正常と異常の機能分割の単位は、状態遷移表に動

表 4 試験系列の生成結果

Table 4 Results of test sequence generation.

系列	ケース数	イベント数
(1)基本接続試験(MT)	5	13
(2)動作試験A(TT)	22	272
a.コネクション/正常	7	66
b.コネクション/異常	6	122
c.データ転送/正常	7	50
d.データ転送/異常	2	34
(3)動作試験B(SW)	6	3920
a.コネクション/状態確認	1	434
b.コネクション/正常	1	1046
c.コネクション/異常	1	1669
d.データ転送/状態確認	1	117
e.データ転送/正常	1	367
f.データ転送/異常	1	287

```

/*
/* CCR SUPUS/SUBASS SPECIFICATION
/*
PROCESS CCR_A;

STATE S_NONE;

INPUT BEG_req;
DECISION CV;          /* Commitment-State Variable */
(P0):
  TASK ACT[1];
  OUTPUT BEG;
  NEXTSTATE W_BEGACK;
(P1):
  OUTPUT BEG;
  NEXTSTATE W_BEGACK;
ENDDECISION;

INPUT OTHER;
OUTPUT nothing;
NEXTSTATE S_NONE;

STATE S_ACT;

INPUT ACT;
DECISION PR_WIN;
(P10): /* This PM is Winner */
  OUTPUT nothing;
  NEXTSTATE S_ACT;
(P11): /* This PM is not Winner */
  OUTPUT nothing;
  NEXTSTATE S_NONE;
ENDDECISION;

INPUT PRP_req;
OUTPUT PRP;
NEXTSTATE S_PRP;

INPUT REF;
    
```

図 8 CCR プロトコルの SDL/PR 記述

Fig. 8 SDL/PR description for CCR protocol.

STATE	INPUT	ACTION ,	OUTPUT ,	NEXT STATE
S_NONE	BEG_req	CV	P0	ACT[1]
		BEG	W_BEGACK	
		CV	P1	BEG
		W_BEGACK		
	OTHER	nothing	S_NONE	
S_ACT	ACT	PR_WIN	P10	nothing
		S_ACT		
		PR_WIN	P11	nothing
		S_NONE		
	PRP_req	PRP	S_PRP	
	REF	R_ACK_ind	S_REF	
	RDY	RDY_ind	S_OFF	
	ROLL_req	CV	P6	ROLLI
		S_ROLL		
		CV	^P6	ACT[5]
	ROLLI	S_ROLL		
ROLH_req	CV	P6H	ROLHI	
	S_ROLH			
	CV	^P6H	ACT[5H]	
	ROLHI	S_ROLH		
RACT_req	RACTI	S_WRACK		

図 9 CCR プロトコルの変換された遷移表
Fig. 9 CCR transition table translated by TENT.

作が規定されている部分を正常処理，規定されずに暗黙の動作となっている部分を異常処理として分割している。

- (1) 基本接続試験
- (2) 動作試験 A (網羅試験)
 - a. コネクション/正常
 - b. コネクション/異常
 - c. データ転送/正常
 - d. データ転送/異常
- (3) 動作試験 B (徹底試験)
 - a. コネクション/状態確認
 - b. コネクション/正常
 - c. コネクション/異常
 - d. データ転送/状態確認
 - e. データ転送/正常
 - f. データ転送/異常

これらの試験目的に対する試験系列の生成結果を表 4 に示す。比較のため，3.2.2 項で述べている分割生成機能を追加しない TT 法，および SW 法による系列の生成も行ったが，その結果の系列長は以下のようになった。

TT 法： 254 イベント (2)

SW 法： 3149 イベント (3)

また，CCR プロトコルは，実際にテストに入力して試験を行う例に適用した。図 8 は，CCR プロトコルの SDL 仕様の一部である。図 9 は，図 8 から変換された遷移表の例である。図 10 は，TENT によって生成された動作試験の試験系列の一部である。CCR プロトコルで生成された試験系列の長さは，

基本接続試験 : 111 イベント

動作試験 A (網羅) : 2401 イベント

動作試験 B (徹底) : 123216 イベント

であった。

5. 評価および考察

ここで，2.3 節の問題点と対比して評価する。問題点 1 については，複数の試験系列生成手法を採用し，試験の段階に応じて異なる誤り検出能力の試験系列を使って解決し，問題点 2，3 および 4 については試験系列生成手法に対して，試験対象に限定した短い系列を生成する機能を追加することで解決した。試験範囲を限定する機能は，フェーズ単位の分割と機能単位の

1	U!	A_ASSOCIATE_req	# STATE = IDLE (reset)
2	L?	AARQ	
3	L!	AARE	# STATE = Wait_AARE (reset)
4	U?	A_ASSOCIATE_conf	
5	U!	BEG_req	# STATE = S_NONE (reset)
6	L?	BEG	
7	L!	BEG_ACK	# STATE = S_BEGACK (reset)
8	U?	nothing	
9	U!	PRP_req	# STATE = S_ACT
10	L?	PRP	
11	L!	REF	# STATE = S_PRP
12	L?	R_ACKO	
13	U?	REF_ind	
14	U!	ROLL_req	# STATE = S_REF
15	L?	ROLLI	
16	L!	ROLLC	# STATE = S_ROLL
17	U?	ROLL_conf	
18	U!	BEG_req	# STATE = S_NONE
19	L?	BEG	
20	L!	BEG_ACK	# STATE = S_BEGACK
21	U?	nothing	
22	L!	REF	# STATE = S_ACT
23	L?	R_ACKO	
24	U?	REF_ind	
25	U!	ROLH_req	# STATE = S_REF
26	L?	ROLHI	
27	U!	CRASH	# STATE = S_ROLH
28	L?	FAIL	

図 10 CCR プロトコルの試験系列生成結果

Fig. 10 Results of test sequence generation for CCR.

分割の2つを用意することで、詳細な指定を可能とした。問題点5については、MT法という基本接続試験用の生成手法を定義して解決した。

TENTでは試験系列の本数の増加に伴い、試験対象とする状態まで状態遷移をさせる予備的な系列が増加するため、全体の系列長が増加する。今回の実験では、生成手法を単純に実現した場合の生成結果と比べて、それほど多くの増加はなかった。表4から、TENTの動作試験Aの系列の総和は、TT法の分割生成機能がない場合(式(2))に対して約7%の増加であり、同様に動作試験Bでは、SW法の分割生成機能がない場合(式(3))に対する24%の増加で済んでいる。分割数と系列長の増加率との関係については今後の実験で明らかにする予定である。

また、TENTでは、試験の効率化のために、試験系列を分割することを提案している。それは、試験系列の総和が短くても、1本の長い系列を生成した場合に非効率な試験となるからである。

例として、 E 個の実現誤りが試験対象システムに含まれているとする。この誤りを、(a)長さ P イベントの1本の系列で試験する場合と、(b)長さ P/x イベントの x 本の系列(全体で P イベント)で試験する場合を考える。試験方法は、誤りを検出したらそこで打ち切り、誤りを訂正した後に系列の最初から試験を再開するものと仮定する。また、それぞれの誤りは独立であると仮定する。(a)では、誤りが均等の間隔で分

散していた場合、誤りをすべて検出するまでの全体のイベント数 Ta は

$$Ta = P/E + 2P/E + \dots + P \\ = (P/E) \cdot (1 + 2 + \dots + E) \\ = \frac{P(1+E)}{2} \quad (4)$$

となる。これに対して、(b)では、1本で E/x 個の誤りが検出されるとすれば、全体のイベント数 Tb は

$$Tb = x \left(\frac{P/x}{E/x} + \frac{2P/x}{E/x} + \dots + \frac{(E/x)P/x}{E/x} \right) \\ = (xP/E) \cdot (1 + 2 + \dots + E/x) \\ = \frac{P(1+E/x)}{2} \quad (5)$$

で与えられ、これで試験が終わる。式(4)と式(5)の結果から、1本の系列で試験を行うより、複数の系列で試験を行う方が、試験系列の総和が長くても実際の試験では効率的であることが分かる。

この実験で生成された系列は、各段階に応じて分割して生成されており、それぞれ機能・フェーズ別に生成されている。試験の種類は11、総ケース数は33である。そして、TT法を用いて生成した系列の系列長は、平均して約10から20程度のイベント数となり、試験に使いやすい系列となっていることが分かる。また、試験仕様の記述に用いる場合や、試験結果の解析を行う上でも、目的別に短い長さで生成された系列は有効であった。

トランスレータにおいて、SDLから遷移表へ変換する過程で、条件分岐の部分に関しては、条件を満たすイベントを入力するという仮定で分割している。しかし、CCRでは外部から制御できないコミットメント状態変数という条件要素が存在している。この変数を参照している条件分岐では、同じ入力イベントでも異なる動作をする可能性があるため、結果を予想できず試験系列が決定できない。そのために、CCRの試験対象からは削除する必要が生じた。この場合、試験対象かどうかを検査しながら生成する本論文の分割生成機能を使うことによって容易に対応することができた。しかし、本手法には非決定的なシステムの試験には、完全に対応できないという根本的な適用限界があり、この問題を解決することが今後の課題である。

TENTにおいては、FSMの遷移表から試験系列

を生成する段階で、要求仕様の到達性が判定される。また、SW 法において、W 集合を生成する過程で要求仕様が冗長かどうか判定される。これによって、製品試験以前に誤りが検出でき、試験の効率化に有効であった。

6. おわりに

通信システムの試験工程を支援するために、試験系列を SDL 仕様から自動生成するツール TENT について、その構成、特徴、および評価結果について論じた。TENT は、入力として仕様記述言語 SDL を用い、出力として、試験シナリオ記述言語 TTCN を用いている。生成される試験系列は、試験の目的に応じた系列長と誤り検出能力を持つ。実際のプロトコル試験に用いた結果、十分に試験系列を生成することが可能であり、入力の SDL 仕様についても、到達性と冗長性の検証が行えた。

今後は、SDL で記述された仕様の検証に関する研究を行っていくと同時に、試験仕様記述を支援するシステムとの有機的な結合について検討する。

謝辞 本論文を執筆するにあたり、貴重な御助言を頂いた九州大学 牛島和夫教授、ならびに同研究室の皆様へ深く感謝する。また、本論文に多くのコメントを頂いたモンリオール大学 G. v. Bochmann 教授、およびコンコルディア大学 B. Sarikaya 教授へ深く感謝する。最後に、本研究を行うにあたり、熱心に討論頂いた三菱電機(株)情報電子研究所の関係各位に感謝する。

参 考 文 献

- 1) ISO: Conformance Testing Methodology and Framework Part-2: Abstract Test Suite Specification, ISO DIS, 9646-2 (1989).
- 2) Rayner, D.: *OSI Conformance Testing, Computer Networks and ISDN Systems 14*, pp. 79-98, North-Holland (1987).
- 3) Naito, S. and Tsunoyama, M.: Fault Detection for Sequential Machines by Transition Tours, *Proc. of IEEE Computing Conference*, pp. 238-243 (1981).
- 4) Chow, T. S.: Testing Software Design Modeled by Finite State Machine, *IEEE Trans. Softw. Eng.*, Vol. SE-4, No. 3, pp. 178-187 (1987).
- 5) Gonenc, G.: A Method for the Design of Fault Detection Experiments, *IEEE Trans. Comput.*, Vol. C-19, pp. 551-558 (1970).
- 6) 島 隆富, 櫻井常雄, 唐沢俊一: データ通信プロトコル試験装置, 国際通信の研究, No. 130, pp. 533-541 (1986).
- 7) Favreau, J. P. and Linn, R. J.: *Automatic Generation of Test Scenario Skeletons from Protocol Specifications Written in Estelle, Protocol Specification, Testing and Verification VI*, pp. 191-202, North-Holland (1986).
- 8) ISO: Estelle: A Formal Description Technique Based on an Extended State Transition Model, ISO 9074 (1989).
- 9) 佐藤文明, 宗森 純, 井手口哲夫, 水野忠則: 有限オートマトンに基づくシステムの試験系列自動生成法の提案—単一遷移検査系列法—, 信学会論文誌, Vol. J72-B-I, No. 3, pp. 183-192 (1989).
- 10) 当麻喜弘: 順序回路論 コンピュータ基礎講座 5, 昭晃堂 (1980).
- 11) Gill, A.: *Introduction to the Theory of Finite State Machines*, McGraw-Hill (1962).
- 12) 佐藤文明, 宗森 純, 勝山光太郎, 水野忠則: 通信システム試験系列生成手法 SW 法の系列短縮アルゴリズム, 1989年電子情報通信学会春期全国大会, p. 3-154 (1989).
- 13) CCITT: *Functional Specification and Description Language, Recommendation Z. 100* (1989).
- 14) ISO: LOTOS: A Formal Description Technique Based on the Temporal Ordering of Observational Behaviour, ISO 8807 (1989).
- 15) 宗森 純, 水野忠則: SDL グラフィックエディタの設計と製作, 情報処理学会論文誌, Vol. 29, No. 7, pp. 676-685 (1988).
- 16) 赤 攝也, 伊理正夫, 後藤英一, 広瀬 健: 情報処理のための数学, 共立出版 (1982).
- 17) ISO: Conformance Testing Methodology and Framework Part-3: Tree and Tabular Combined Notation, ISO DP 9646-3 (1989).
- 18) ISO: Connection Oriented Transport Protocol Specification, ISO 8073 (1986).
- 19) ISO: Commitment, Concurrency and Recovery, ISO DIS 9804-9805 (1987).

付 録

1. MT 法の試験系列生成手法

MT 法は、すべての状態を一通り通過するような、なるべく短い試験系列を求める方法である。MT 法の試験系列生成手法は、以下の手順で行う。

- (a) 各状態の通過フラグをクリアする。
- (b) 各状態間での最短パスを求める (Dijkstra 法¹⁶⁾を採用)。
- (c) 通過フラグが立っていない、初期状態からの最短パスが最長の状態を求める。
- (d) 初期状態から最短パスで(c)の状態へ遷移し

て戻る系列を生成する。

(e) 途中の経路の状態の通過フラグを立てる。

(f) すべての状態を通過していれば終了。

(g) 手順(c)へ戻る。

初期状態から最短パスで往復するのは、状態数を m とすると、高々 $2(m-1)$ イベントである。また、初期状態から他の状態までの往復の回数は高々 $m-1$ 回である。よって、MT 法で生成される試験系列の上限値は、

$$Lmt = 2(m-1)^2 \quad (\text{付-1})$$

となる。

(平成元年8月31日受付)

(平成2年7月10日採録)



佐藤 文明 (正会員)

昭和37年生。昭和59年岩手大学工学部電気工学科卒業。昭和61年東北大学大学院修士課程修了。同年三菱電機(株)入社。現在、同情報電子研究所にて、通信ソフトウェア開発環境、形式的仕様記述技法に関する研究に従事。プログラミング言語、並列分散処理システムに興味を持つ。電子情報通信学会会員。



宗森 純 (正会員)

昭和30年生。昭和54年名古屋工業大学電気工学科卒業。昭和56年同大学院工学研究科電気工学専攻修士課程修了。昭和59年東北大学大学院工学研究科電気及通信工学専攻博士課程修了。工学博士。同年三菱電機(株)入社。昭和59年～平成元年同社情報電子研究所にて、通信ソフトウェア開発環境の研究開発に従事。平成元年4月より鹿児島大学工学部情報工学科助教授。ソフトウェア工学、ヒューマンインタフェース、形式記述技法、ニューラルネットワークに興味を持つ。電子情報通信学会、オフィスオートメーション学会各会員。



勝山光太郎 (正会員)

昭和29年生。昭和51年3月大阪大学基礎工学部制御工学科卒業。同年三菱電機(株)入社。現在同社情報電子研究所システム技術開発部。入社以来通信ソフトウェアの開発、分散処理システムの研究開発に従事。電子情報通信学会会員。



水野 忠則 (正会員)

昭和20年生。昭和43年名古屋工業大学経営工学科卒業。同年三菱電機(株)入社。現在同社情報電子研究所システム技術開発部。工学博士。情報通信システムおよび分散処理システムに関する研究・開発に従事。著書としては『マイコンローカルネットワーク』(産報出版)、『分散処理システム・デザイン』(共訳、工学社)、『電子メールとメッセージ通信』(監訳、工学社)などがある。電子情報通信学会、オフィスオートメーション学会、日本経営工学会、IEEE 各会員。