通信プロトコル適合性試験における レジスタ操作に対する試験系列の生成手法

樋口昌宏[†] 小原勝[†]
中石敬治[††] 藤井護[†]

本論文では，拡張有限状態機械（EFSM）のサブクラスであるカウンタ付き有限状態機械（FSM-C）としてモデル化された通信プロトコルに対して，そのレジスタ操作の正しさを試験する系列が満たすべき条件，およびそのような系列が存在するときそれを生成する手順を提案する．本手法を用いて生成した試験系列を用いると，遷移条件の条件判定およびレジスタ代入操作の単一誤りを検出できる．また，この手法の有効性を確認するため，提案手法に基づく試験系列生成システムを作成し，OSI セションプロトコルに対して実際に試験系列の生成を行った．その結果，すべての試験対象に対して試験系列が存在し，自動生成できることが確認できた．

A Method for Generating Test Cases for Register Operations in Conformance Testing of Communication Protocols

Masahiro Higuchi,[†] Masaru Kohara,[†] Takaharu Nakaishi[††] and Mamoru Fujii[†]

In this paper, we propose a condition of test cases for communication protocols modeled as FSM with counters (FSM-C), a subclass of Extended Finite-State Machines (EFSM). By using test cases satisfing the proposing condition, single fault of on determinig inequality or register assignments can be detected. We also propose a procedure to find test cases which satisfy the condition. To see the usefulness of this method, we developed a test cases generating system based on the proposing method, and applied the system to the OSI session protocol. As a result we can verify that all of test cases are generated automatically.

1. まえがき

通信システムの開発において，作成されたシステムがプロトコル仕様どおりに動作するかどうかの試験（適合性試験）は重要である．プロトコル適合性試験は試験対象となるシステム（IUT: Implementation Under Test）に試験系列を入力し，その際の出力が仕様どおりであるかどうかを観察することによって行われる．適合性試験に用いる試験系列の生成は従来人手により行われていた．試験効率を上げるためには試験系列を自動で生成する手法を確立する必要がある．

有限状態機械（FSM）モデルで定義された通信プロトコルに対する試験系列の自動生成手法は広く研究されている．[1] しかし，多くの通信プロトコルは一連番号やタイマを扱うためにFSMの有限制御部に整数値などを保持するレジスタを持たせた拡張有限状態機械（EFSM）モデルで定義される．EFSMに対する試験系列の生成手法として文献2)では，状態の存在および状態遷移の正しさについて試験する系列の生成手法を提案している．また，各遷移で実行するレジスタ操作を試験する系列の生成手法は，Wangらが文献3)で提案している．しかし，Wangらの手法ではレジスタ操作のうち遷移条件の正しさを試験することができない．さらに，レジスタ代入に関しても，代入された値が直接または他のレジスタを介して出力に現れない場合には，その正しさを試験することはできない．我々の研究グループではEFSMの部分クラスであるカウンタ付き有限状態機械（FSM-C）に対して，Wangらの手法では試験できなかった遷移条件およびレジスタ代入に対する試験系列の生成手法の研究を進めている[4]．

以降，2章では，本論文で対象としているプロトコ

† 大阪大学大学院基礎工学研究科
Graduate School of Engineering Science, Osaka University
†† 日本電信電話株式会社
NTT

ルモデルと試験の形式について，3章では遷移条件の判定に対する試験系列の生成手法について，4章ではレジスタ代入に対する試験系列の生成手法について，そして5章では試験系列生成システムを作成して実験を行った結果について述べる．

2. 準　　備

2.1 プロトコル機械モデル

本論文で対象とする通信プロトコルは図**1**で示すとおり，入出力用のゲートを介して通信を行う．入力および出力はそれぞれコマンドと整数値パラメータの組とする．また，図1中のプロトコル機械は以下で定義されるFSM-Cとして与えられるものとする．

定義1 FSM-Cを$(Q, R, \Sigma, \Delta, T, I)$で定義する．

(M1) Q: 有限制御部の状態の有限集合．

(M2) R: 整数値レジスタの有限集合．

(M3) Σ: 入力コマンドの有限集合．

(M4) Δ: 出力コマンドの有限集合．

(M5) T: アクションの有限集合．アクション$t \in T$は以下の6字組

$[u_t, C_t, I_t, O_t, R_t, v_t]$ で定義される．

- $u_t \in Q$：始状態．
- C_t：遷移条件．いくつかの差分不等式の論理積で与えられ，$d_1 \wedge d_2 \wedge \cdots \wedge d_k$（$d_i$はそれぞれ1つの差分不等式）と表される．ここで，差分不等式とは，x, yをレジスタまたは入力パラメータ，cを整定数としたとき$x - y \leq c$の形の不等式である．本論文では$x \leq c, -y \leq c$の形の式も差分不等式に含めて考える．以下では説明のために集合を用いて$C_t = \{d_1, d_2, \cdots, d_k\}$と書き表す．
- $I_t \in \Sigma$：入力コマンド．
- O_t：出力定義式．$\langle RSP_t, x + c\rangle$ もしくは $\langle RSP_t, c\rangle$．$RSP_t \in \Delta$, xはレジスタもしくは入力パラメータ，cは整定数とする．
- R_t：アクションtで実行するレジスタ代入式の集合．$r \in R$，xをレジスタまたは入力パラメータ，cを整定数としたとき，レジスタ代入式は“$r \leftarrow x + c$”，“$r \leftarrow c$”のいずれかの形の式とする．
- $v_t \in Q$：終状態．

(M6) I: 初期状態．以下の形の式により指定する．

- $q_{init} = s_0$
- 各レジスタrについて，$r_{init} = \sigma_r$

これは有限制御部の初期状態がs_0でレジスタrの初期値がσ_rであることを表す．また，あるレジスタrに対して上記の式が記述されていない場合には，rの初期値が未定義である． □

アクション$t \in [q_i, C_t, CMD, \langle RSP, x + c\rangle, R_t, q_j]$は有限制御部の状態が$q_i$であるとき，入力ゲートから$\langle CMD, ip\rangle$が入力されると，そのときのレジスタ値および$ip$の値が$C_t$を満たしている場合に実行され，レジスタ代入$R_t$を行い，$\langle RSP, x + c\rangle$を出力ゲートに出力し，有限制御部の状態を$q_j$にする．FSM-Cにその状態からの動作が定義されていない入力が行われると，FSM-Cはエラーメッセージを出力して初期状態に遷移する．このアクションをエラーアクションと呼ぶ．本論文で試験対象とするFSM-Cの状態遷移はエラーアクションによる遷移も含めると完全かつ決定性で，その実行時間は有限であるものとする．FSM-Cについて，次の補題が成り立つ．

補題1 1つの入出力系列にはちょうど1つのアクション系列が存在する．

[**略証**] 状態遷移が決定性であることより． □

例1 図**2**にFSM-Cの例を示す．この図では，有限制御部の状態を頂点で，アクションを有向辺で表している．それぞれの辺に付加されたラベルはそのアクションで実行する動作を

[{遷移条件},⟨入力コマンド⟩,⟨出力定義式⟩,{レジスタ代入式}] という形式で表している．また，遷移条件やレジスタ代入式で参照される入力パラメータは

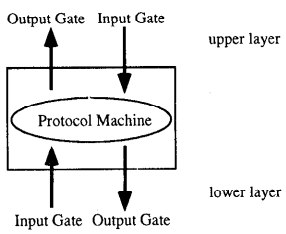


図1 入出力ゲートを用いた通信
Fig. 1 Communication with I/O gate.

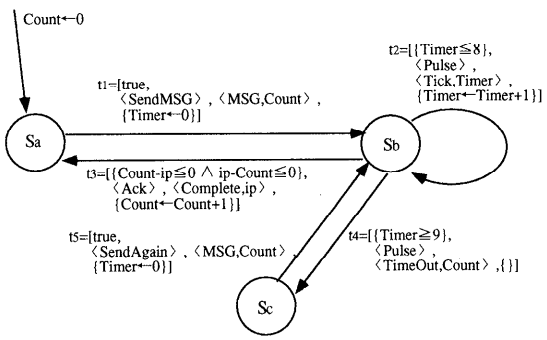


図2 FSM-Cモデルの例
Fig. 2 An example of FSM-C model protocol.

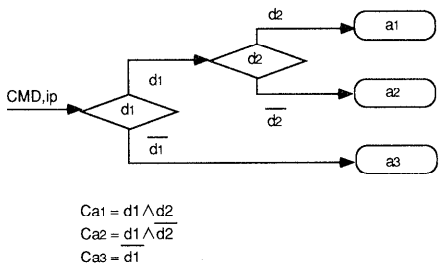


図3 遷移条件式の判定手順
Fig. 3 Method for judgement of conditions.

ip で表している．なお，この図ではエラーアクションは省略している．このFSM-Cの初期状態では，有限制御部の状態が S_a，レジスタ $Count$ の値が0，レジスタ $Timer$ が未定義である．

このプロトコルは，$SendMSG$ を受け取ると，メッセージ番号 $Count$ を付した MSG を送信し，クロックパルスが10回入力されるまで，そのメッセージ番号に対するAck信号が返ってくる（入力される）のを待つ．状態 S_a はゲートからの入力を待っている状態であり，コマンド $SendMSG$ を受け取る．状態 S_b で Ack が返ってくるのを待つ．すなわち，$Pulse$ が入力されると $Timer$ をインクリメントし，その $Timer$ の値が9を超えれば $TimeOut$ を出力し状態 S_c に遷移して再送が要求されるのを待つ．それ以前に Ack が返ってくれば状態 S_a へ遷移する． □

また，同じ状態を始状態とし，入力コマンドも同じであるアクション $t_1,\cdots,t_n$ がある場合，入力が行われた際の遷移条件の判定は**図3**で示されたように差分不等式を逐次，評価を行えるものとする．図3において，d_1，d_2 は1つの差分不等式を表す．ここで，d_i が差分不等式である場合その否定 $\overline{d_i}$ も差分不等式である．

$(d_i,\overline{d_i})$ の判定を行う場合，その分岐先のアクションの中で少なくとも1つはエラーアクションではないとする．すべてがエラーアクションである場合 d_i の判定を行わずエラーアクションを実行すればよい．

上記のような制約の下でも，一連番号やタイムアウト制御を行うプロトコルを簡潔に定義できる．

2.2 適合性試験の形式

通常，IUTの内部状態は外部から観察できない．そこで，適合性試験はブラックボックス試験という方法で行われる．ブラックボックス試験は，IUTに入力系列を与え，その出力が仕様どおりのものであるかを調べることによって行われる．

EFSMに対する試験系列の1つとして状態を同定するE-UIO系列[2]がある．これはFSMにおいて1つの状態を同定するためのUIO系列を拡張したものである．EFSMにおいて1つの状態を識別するためには，その状態に対するUIO系列の前にそのUIO系列を実行可能とするようなレジスタ値を設定する先行系列を連接しなければならない．UIO系列にこのような先行系列を連接した系列がE-UIO系列である．仕様に記述された状態がすべて存在すること，各状態遷移の始状態と終状態が仕様どおりであることはE-UIO系列を用いて文献2)の方法で試験できる．本論文で提案している手法では，系列を実行した後の有限制御部の状態が仕様どおりの状態であることを確認する必要があるため，このE-UIO系列をあわせて用いる．

3. 遷移条件の正しさを試験する試験系列

FSM-Cでは，レジスタ操作として遷移条件の判定とレジスタ代入がある．ここでは，遷移条件の判定の正しさを試験する系列について述べる．本論文では，プロトコル仕様は下記の条件1を満たし，IUTはFSM-Cと見なすことができるものとする．

[条件1] エラーアクション以外の任意の2つのアクション $t_i,t_j(i\neq j,1\leq i,j\leq|T|)$ について，その始状態，終状態および入出力コマンドがともに同じものはない． □

アクション定義中の始状態が s で入力コマンドが CMD であるアクションの集合を $T_{s,CMD}$ とし，$C_{s,CMD}=\bigcup_{t_j\in T_{s,CMD}} C_{t_j}$ とおく．このとき，すべての $d_i(\in C_{s,CMD})$ の判定が正しく実装されていることを示すことによって，すべての $C_{t_j}(t_j\in T_{s,CMD})$ が正しく実装されていることを示すことができる．図3の手順で条件判定を行っており，$d_i\in C_{s,CMD}$ なら $\overline{d_i}\in C_{s,CMD}$ であるため，$d_i,\overline{d_i}$ の組ごとに試験を行う．以下では試験対象を $C_{s,CMD}$ に含まれる不等式 $\langle d_i,\overline{d_i}\rangle(d_i:x-y\leq c)$ であるとする．また，条件判定の手順の性質より，$d_i\in C_a$ なるエラーアクションでないアクション a が存在するものとする．

3.1 定数 c が正しいことを試験する系列

定数 c の正しさを試験するためには次の2つの系列を用いる．1つは c より小さな値 c' を用いて $x-y\leq c'$ と実装されている場合に実行できない入出力系列（IOC_G）で，もう1つは c より大きな値 c'' を用いて $x-y\leq c''$ と実装されている場合に実行できない入出力系列（IOC_L）である．IOC_G，IOC_L には，以下の条件を満たすあるアクション a に対して，次のような系列を用いる．

- a の始状態は s.
- a の遷移条件は不等式 $d_i:x-y\leq c$ を含む.

• a はエラーアクションではない．

(IOC_G) 初期状態から実行可能な入出力系列であって，その最後の遷移は a によるものである．また，最後の遷移前の状態において，$x-y=c$ が成り立つ．

(IOC_L) 初期状態から実行可能な入出力系列であって，その最後の遷移は a 以外のアクションによるものである．また，最後の遷移前の状態において，a の遷移条件式のうちの d_i 以外の各遷移条件式および，$x-y=c+1$ が成り立つ．

形式的に書くと，IOC_G は以下の条件を満たす．$IOC_G = i_{G,-m}/o_{G,-m}\cdot\ldots\cdot i_{G,0}/o_{G,0}$ とおく．ただし，$i_{G,k}/o_{G,k}$ は入力と出力の組を表しており，それぞれ $i_{G,k} = \langle CMD_{G,k}, ip_{G,k}\rangle$，$o_{G,k} = \langle RSP_{G,k}, op_{G,k}\rangle$ である．また，入出力系列 IOC_G に対応するアクション系列を $TRC_G = t_{G,-m}\cdot\ldots\cdot t_{G,0}$ とおく．$s_{G,k}$，$r_{G,k}$ $(-m \le k \le 0)$ は $t_{G,k}$ 実行前の有限制御部の状態，レジスタ r の保持している値を，$s_{G,1}$，$r_{G,1}$ はそれぞれ IOC_G 実行後の有限制御部の状態，レジスタ r の保持している値を表すものとする．

(1) 有限制御部の状態に関する条件：

(a) $s_{G,-m} = q_{init}$

(b) $-m \le k \le 0$ について，

$s_{G,k} = u_{t_{G,k}}$，$s_{G,k+1} = v_{t_{G,k}}$

(2) 入出力コマンドに関する条件：

(a) $-m \le k \le 0$ について，

$CMD_{G,k} = I_{t_{G,k}}$

$RSP_{G,k} = RSP_{t_{G,k}}$

(3) アクションに関する条件：

(a) $-m \le k \le -1$ について，

$t_{G,k} \neq t_{G,0}$

(b) $k=0$ について，

$t_{G,0}$はエラーアクションではない．

$d_i \in C_{t_{G,0}}$

(4) 以下の不等式からなる連立不等式 Φ_{TRC_G} が整数解を持つ．

(a) 各レジスタ r に対して，

$r_{G,-m} = r_{init}$

(b) $-m \le k \le -1$ について：

$C_{t_{G,k}}$ 中の各不等式 "$x-y \le c$" に対して，

$x_{G,k} - y_{G,k} \le c$

$R_{t_{G,k}}$ 中の各レジスタ代入式 "$r \leftarrow x+c$" に対して

$r_{G,k+1} = x_{G,k} + c$

$R_{t_{G,k}}$ 中のいずれのレジスタ代入式の左辺にも現れないレジスタ r に対して，

$r_{G,k+1} = r_{G,k}$

出力定義式 "$\langle \mathrm{RSP}_{t_{G,k}}, x+c\rangle$" に対して，

$op_{G,k} = x_{G,k} + c$

(c) $k=0$ について：

$C_{t_{G,0}}$ 中の d_i 以外の各不等式 "$x-y \le c$" に対して，

$x_{G,0} - y_{G,0} \le c$

$R_{t_{G,0}}$ 中の各レジスタ代入式 "$r \leftarrow x+c$" に対して

$r_{G,1} = x_{G,0} + c$

$R_{t_{G,0}}$ 中のいずれのレジスタ代入式の左辺にも現れないレジスタ r に対して，

$r_{G,1} = r_{G,0}$

出力定義式 "$\langle \mathrm{RSP}_{t_{G,0}}, x+c\rangle$" に対して，

$op_{G,0} - x_{G,0} + c$

(d) 試験対象の式 $d_i : x-y \le c$ に対して，

$x_{G,0} - y_{G,0} = c$

上記の条件を満たす入出力系列 IOC_G は必ずしも存在するとは限らないが，存在する場合には図2の形のグラフに対して状態 s から有向辺を逆向きに幅優先探索を行うことによりアクション系列 TRC_G の候補を順次生成し，それぞれについて不等式 Φ_{TRC_G} を解くことで求めることができる．Φ_{TRC_G} の整数解から IOC_G の入出力パラメータの値を決定できる．不等式 Φ_{TRC_G} は，試験対象を FSM-C に限定したことにより，遷移条件，レジスタ代入式の右辺の形に制限をおいているので Bellman-Ford のアルゴリズム[5]を用いることにより，$O(lm)$（ただし，l は不等式の数，m は変数の数）で解くことができる．

IOC_G が正しく実行されているかどうかを調べるために以下の試験系列 $IOCU_G$ を用いる．

(1) $t_{G,0}$ と始状態が同じで，入出力コマンドがそれぞれ CMD，RSP であるアクションが $t_{G,0}$ 以外に存在する場合：

状態 $s_{G,1}$ を同定する系列 $UIO_{s_{G,1}}$ を作成し，IOC_G に連接する（$IOCU_G = IOC_G \cdot UIO_{s_{G,1}}$）．ただし，$UIO_{s_{G,1}}$ はその実行においてアクション $t_{G,0}$ の実行を含まず，IOC_G を先行系列として連接することにより *E-UIO* 系列となる系列である．

(2) $t_{G,0}$ と始状態が同じで，入出力コマンドがそれぞれ CMD，RSP であるアクションが $t_{G,0}$ 以外に存在しない場合：

何も連接しない．($IOCU_G = IOC_G$)

$IOCU_L$ もほぼ同様にして求めることができる．このようにして作成した系列について以下の定理1が成り立つ．

定理1 IUTにおいて，試験対象となる遷移条件式 $d_i : x - y \leq c, \overline{d_i}$ 以外は正しく実装されているものとする．このとき，d_i が c'（$c' < c$）を用いて $x - y \leq c'$ と誤って実装されている場合は $IOCU_G$ を，c''（$c'' > c$）を用いて $x - y \leq c''$ と誤って実装されている場合には $IOCU_L$ を仕様どおりに実行することができない．

[略証] $d_i' : x - y \leq c'$（$c' < c$）と誤って実装されている場合：IOC_G において入出力 $\langle \mathrm{CMD}_{G,-m}, ip_{G,-m}\rangle / \langle \mathrm{RSP}_{G,-m}, op_{G,\ m}\rangle \cdot \ldots \cdot \langle \mathrm{CMD}_{G,-1}, ip_{G,-1}\rangle / \langle \mathrm{RSP}_{G,-1}, op_{G,-1}\rangle$ を実行後の各レジスタ値は $x_{G,0} - y_{G,0} = c$ である．このとき，$d_i' : x - y \leq c'$ は成立しない．$d_i \in C_{t_{G,0}}$ であるのでIUTでは，$t_{G,0}$ 以外のアクション $t'_{G,0}$ を実行する．$t_{G,0}$ と $t'_{G,0}$ の出力コマンドが異なる場合，仕様どおりに $\langle \mathrm{CMD}_{G,0}, ip_{G,0}\rangle / \langle \mathrm{RSP}_{G,0}, op_{G,0}\rangle$ を実行できない．また，出力コマンドが互いに等しい場合，条件1より終状態が異なる．この場合，UIO 系列の性質より，$UIO_{s_{G,1}}$ を仕様どおりに実行できない．

以上より，$IOCU_G$ が実行できない．

$d_i'' : x - y \leq c''$（$c'' > c$）と誤って実装されている場合：IOC_L において入出力 $\langle \mathrm{CMD}_{L,-m}, ip_{L,-m}\rangle / \langle \mathrm{RSP}_{L,-m}, op_{L,-m}\rangle \cdot \ldots \cdot \langle \mathrm{CMD}_{L,-1}, ip_{L,-1}\rangle / \langle \mathrm{RSP}_{L,-1}, op_{L,-1}\rangle$ を実行後の各レジスタ値は $x_{L,0} - y_{L,0} = c+1$ である．このとき，$d_i'' : x - y \leq c''$ が成立するため，仕様では a を実行しないがIUTでは a を実行する．$t_{L,0}$ と a の出力コマンドが異なる場合，仕様どおりに $\langle \mathrm{CMD}_{L,0}, ip_{L,0}\rangle / \langle \mathrm{RSP}_{L,0}, op_{L,0}\rangle$ を実行できない．また，出力コマンドが互いに等しい場合，条件1より終状態が異なる．この場合，UIO 系列の性質より，$UIO_{s_{L,1}}$ を仕様どおりに実行できない．

以上より，$IOCU_L$ が実行できない． □

定理1より，系列 $IOCU_G$，$IOCU_L$ を用いることにより，差分不等式 $d_i, \overline{d_i}$ の定数 c が正しく実装されていることを試験することができる．

例2 図2のFSM-Cに対する試験系列の例を示す．ここでは，試験対象の遷移条件式を $d_i : Count - ip \leq 0$（$d_i \in C_{S_b,Ack}$）とする．このとき，IOC_G は以下の入出力系列である．$IOC_G = \langle SendMSG, 0\rangle / \langle MSG, 0\rangle, \langle Ack, 0\rangle / \langle Complete, 0\rangle$．この入出力系列を実行した際，最後に実行されるアクションは t_3 であるが，始状態が S_b，入力コマンドが Ack，出力コマンドが $Complete$ であるアクションは t_3 以外に存在しないため，$IOCU_G = IOC_G$ である．仕様では，2つ目の入力が行われた際，コマンド $Complete$ が出力される．しかし，d_i が誤って $Count - ip \leq c'(c' < 0)$ と実装されている場合，IUTではエラーアクションが実行されるため $Complete$ 以外のコマンドが出力されるため，誤りが検出される． □

3.2 レジスタが正しいことを試験する系列

条件式 $d_i : x - y \leq c$ の判定において，レジスタもしくは入力パラメータ x が正しく参照されていることを試験するためには，x と異なるレジスタまたは入力パラメータ（以下では z とする）について，以下の条件を満たすあるアクション a に対して，下記の2つの入出力系列のいずれかを用いる．

- a の始状態は s．
- a の遷移条件は不等式 $d_i : x - y \leq c$ を含む．
- a はエラーアクションではない．

（$IOC_{z,T}$） 初期状態から実行可能な入出力系列で最後のアクションが実行される前の有限制御部の状態は s．遷移条件が仕様どおりに実装されている場合には条件式 d_i を true と判定し a を実行するが，z を用いて $z - y \leq c$ と誤って実装されている場合には d_i を false と判定し a 以外のアクションを実行する．

（$IOC_{z,F}$） 初期状態から実行可能な入出力系列で最後のアクションが実行される前の有限制御部の状態は s．遷移条件が仕様どおりに実装されている場合には条件式 $\overline{d_i}$ を true と判定し，a 以外のアクションを実行するが，z を用いて $z - y \leq c$ と誤って実装されている場合には $\overline{d_i}$ を false と判定し a を実行する入出力系列．

形式的には入出力系列 $IOC_{z,T}$ は以下の条件を満たす．

$IOC_{z,T} = i_{T,-m}/o_{T,-m} \cdot \ldots \cdot i_{T,0}/o_{T,0}$ とし，$i_{T,k}$，$o_{T,k}$，$s_{T,k}$，$r_{T,k}$ は3.1と同様に定義する．また，$IOC_{z,T}$ に対応するアクション系列を $TRC_{z,T} = t_{T,-m} \cdot \ldots \cdot t_{T,0}$ とおく．

(1) 有限制御部の状態に関しては，
3.1節(1)と同様の条件を満たす．

(2) 入出力コマンドに関しては，
3.1節(2)と同様．

(3) アクションに関しては，
3.1節(3)と同様．

(4) 以下の不等式からなる連立不等式 $\Phi_{TRC_{z,T}}$ が整数解を持つ．

(a) 3.1節(4)の式のうち(a)，(b)，(c)と同様．

(b) 試験対象の式 $d_i : x - y \leq c$ に対して，

$$x_{T,0} - y_{T,0} \leq c,$$

$$y_{T,0} - z_{T,0} \leq -c - 1 \quad (\Leftrightarrow \overline{z_{T,0} - y_{T,0} \leq c})$$

以上の条件を満たす $IOC_{z,T}$ は3.1節と同様に求め

ることができる．さらに，3.1 節と同様に必要に応じて $IOC_{z,T}$ 実行後の状態が $s_{T,1}$ であることを同定する系列を連接することにより $IOCU_{z,T}$ を作成する．$IOCU_{z,T}$ と同様に $IOCU_{z,F}$ も作成できる．このようにして作成した入出力系列 $IOCU_{z,T}$，$IOCU_{z,F}$ について以下の定理 2 が成り立つ．

定理 2 IUT において，試験対象となる遷移条件式 $d_i : x - y \leq c, \overline{d_i}$ 以外は正しく実装されているものとする．このとき，d_i が x と異なるレジスタ z を用いて $d_i' : z - y \leq c$ と誤って実装されている場合，$IOCU_{z,T}$，$IOCU_{z,F}$ いずれも仕様どおり実行できない．

［略証］$IOC_{z,T}$ において $\langle \mathrm{CMD}_{T,-m}, ip_{T,-m}\rangle$ / $\langle \mathrm{RSP}_{T,-m}, op_{T,-m}\rangle \cdot \ldots \cdot \langle \mathrm{CMD}_{T,-1}, ip_{T,-1}\rangle$ / $\langle \mathrm{RSP}_{T,-1}, op_{T,-1}\rangle$ 実行後の各レジスタは，

$$x_{T,0} - y_{T,0} \leq c,$$
$$y_{T,0} - z_{T,0} \leq -c - 1$$

を満たしており，d_i は真，d_i' は偽となる．$d_i \in C_{t_{T,0}}$ であるので，仕様では $t_{T,0}$ を実行するが，IUT では異なるアクション $t'_{T,0}$ を実行する．このとき，定理 1 の略証と同じ理由で，$IOCU_{z,T}$ を仕様どおりに実行できない．

同様に $IOCU_{z,F}$ も仕様どおりに実行できない． □

定理 2 より，入出力系列 $IOCU_{z,T}$，$IOCU_{z,F}$ のどちらかを用いることにより，レジスタ x が正しく参照されていることが試験できる．レジスタ y についても同様にして試験することができる．

4. レジスタ代入の正しさの試験

アクションのレジスタ代入の正しさを試験する系列として，Wang らは試験対象となるレジスタ代入での代入された値が，直接または他のレジスタを介して出力パラメータに現れるような遷移系列を試験系列とする方法を提案している[3]．しかし，現実のプロトコル仕様ではタイマ値やウィンドサイズの設定などのように，レジスタに代入された値がその後の制御に用いられるのみで，出力値として現れないような場合も多い．この場合には Wang らの方法ではレジスタ代入が正しく行われたことを試験できない．そこで本論文では試験対象となるレジスタ代入の代入結果が直接または，他のレジスタを介して出力パラメータに現れない場合のレジスタ代入の正しさを試験する系列を提案する．

アクションのレジスタ代入の正しさを試験するために，試験対象となるレジスタ代入で正しい値がレジスタに代入されていなければ実行できない系列を試験系列として用いる．本手法では各レジスタ代入操作に対して次のような 2 つの試験系列を作成する．

（$IO_{TR} \cdot IO_{TG}$） 試験対象のレジスタ代入で正しい値より小さな値が代入された場合実行できない入出力系列

（$IO_{TR} \cdot IO_{TL}$） 試験対象のレジスタ代入で正しい値より大きな値が代入された場合実行できない入出力系列

ここで，IO_{TR}，IO_{TG}，IO_{TL} は，試験対象のレジスタ代入をアクション t の $R \leftarrow X + C$ としたときそれぞれ以下のような入出力系列となる．

IO_{TR}： 初期状態から実行可能な入出力系列で，その最後の入出力で実行されるアクションは t．

IO_{TG}： IO_{TR} 実行後に実行可能で，試験対象のアクション t でレジスタ R に $X + C$ より小さな値が代入された場合には実行できない入出力系列．

IO_{TL}： IO_{TR} 実行後に実行可能で，試験対象のアクション t でレジスタ R に $X + C$ より大きな値が代入された場合には実行できない入出力系列．

各系列は，IO_{TR} の最後で実行されるアクション t 以外にはアクション t を実行しないものとする．

形式的には IO_{TR} は以下の条件を満たす系列である．$IO_{TR} = i_{TR,-l}/o_{TR,-l} \cdot \ldots \cdot i_{TR,0}/o_{TR,0}$ とし，$i_{TR,k}$，$o_{TR,k}$，$s_{TR,k}$，$r_{TR,k}$ $(-l \leq k \leq 0)$ は 3.1 節と同様に定義する．また，$s_{TR,1}$，$r_{TR,1}$ はそれぞれ IO_{TR} 実行後の有限制御部の状態，レジスタ r の保持している値を表す．

(1) 有限制御部の状態に関しては，3.1 節の (1) と同様．

(2) 入出力コマンドに関しては，3.1 節の (2) と同様．

(3) アクションに関しては，3.1 節の (3) (a) と同様．

IO_{TG}，IO_{TL} は以下の条件を満たす．$IO_{TG} = i_{G,1}/o_{G,1} \cdot \ldots \cdot i_{G,m}/o_{G,m}$，$IO_{TL} = i_{L,1}/o_{L,1} \cdot \ldots \cdot i_{L,n}/o_{L,n}$ とおく．ここでは，IO_{TG}，IO_{TL} それぞれに対応するアクション系列をそれぞれ，$TST_G = t_{G,1} \cdot \ldots \cdot t_{G,m}$，$TST_L = t_{L,1} \cdot \ldots \cdot t_{L,n}$ とおく．特に $t_{G,m} = t_G$, $t_{L,n} = t_L$ と書くことにする．

また，$J \in \{G, L\}$ について，$ip_{J,k}$ $s_{J,k}$，$r_{J,k}$ および，$s_{G,m+1}$，$r_{G,m+1}$，$s_{L,n+1}$，$r_{L,n+1}$，はそれぞれ 3.1 節と同様に定義する．また，$r'_{J,k}$，$ip'_{J,k}$ は，アクション t 実行時に試験対象となるレジスタ代入式 “$R \leftarrow X + C$” に対して，ぞれぞれ正しい値より小さな値（$J = G$ の場合）あるいは正しい値より大きな値（$J = L$ の場合）を代入したときの，アクション $t_{J,k}$ 実行前のレジスタ r の保持している値，アクショ

ン $t_{J,k}$ 実行における入力パラメータを表す.

(1) 有限制御部の状態に関しては,
3.1 節 (1) と同様.

(2) 入出力コマンドに関しては,
3.1 節 (2) と同様.

(3) アクションに関しては,
3.1 節 (3) と同様.

(4) レジスタ, 入力パラメータに関する条件:
試験対象となるレジスタ代入式 "$R \leftarrow X + C$" として, 以下の 2 つの不等式からなる連立不等式を Φ_R とする.

$$R'_{G,1} \leq R_{TR,1} - 1$$
$$R'_{L,1} \geq R_{TR,1} + 1$$

ここで, $J \in \{G, L\}$ について, $R'_{J,k}$ は $IO_{TR} \cdot IO_{TG}$ ($IO_{TR} \cdot IO_{TL}$) を IUT で実行した際の, $t_{J,k}$ を実行する前のレジスタ R の値である. 入出力系列 IO_{TR} に対応するアクション系列を TR とし, 3.1 節の (4) (a), (b), (c) と同様に定義した Φ_{TR} の各不等式と, 以下の不等式からなる連立不等式を Φ_{TST} としたときに, $\exists V \forall R'_{G,1}, R'_{L,1}\{\Phi_R \supset \Phi_{TST}\}$ が真となる. ただし, V とは $R'_{G,1}, R'_{L,1}$ 以外のレジスタおよび変数の値を表す.

(a) 試験対象となるアクション実行後のレジスタ値に関する条件:
各レジスタ r に対して,

$$r_{G,1} = r_{TR,1}$$
$$r_{L,1} = r_{TR,1}$$

R 以外の各レジスタ r に対して,

$$r'_{G,1} = r_{TR,1}$$
$$r'_{L,1} = r_{TR,1}$$

(b) $1 \leq k \leq m-1$ について:
$C_{t_{G,k}}$ 中の各不等式 "$x - y \leq c$" に対して,

$$x_{G,k} - y_{G,k} \leq c$$
$$x'_{G,k} - y'_{G,k} \leq c$$

$R_{t_{G,k}}$ 中の各レジスタ代入式 "$r \leftarrow x+c$" に対して,

$$r_{G,k+1} = x_{G,k} + c$$
$$r'_{G,k+1} = x'_{G,k} + c$$

$R_{t_{G,k}}$ 中のいずれのレジスタ代入式の左辺にも現れないレジスタ r に対して,

$$r_{G,k+1} = r_{G,k}$$
$$r'_{G,k+1} = r'_{G,k}$$

(c) TST_G の最後のアクション t_G について:
C_{t_G} 中の各不等式 "$x - y \leq c$" に対して,

$$x_{G,m} - y_{G,m} \leq c$$

C_{t_G} 中のある不等式 "$X_G - Y_G \leq C_G$" に対して,

$$X'_{G,m} - Y'_{G,m} \geq C_G + 1$$

(d) $1 \leq k \leq n-1$ について:
$C_{t_{L,k}}$ 中の各不等式 "$x - y \leq c$" に対して,

$$x_{L,k} - y_{L,k} \leq c$$
$$x'_{L,k} - y'_{L,k} \leq c$$

$R_{t_{L,k}}$ 中の各レジスタ代入式 "$r \leftarrow x+c$" に対して,

$$r_{L,k+1} = x_{L,k} + c$$
$$r'_{L,k+1} = x'_{L,k} + c$$

$R_{t_{L,k}}$ 中のいずれのレジスタ代入式の左辺にも現れないレジスタ r に対して,

$$r_{L,k+1} = r_{L,k}$$
$$r'_{L,k+1} = r'_{L,k}$$

(e) TST_L の最後のアクション t_L について:
C_{t_L} 中の各不等式 "$x - y \leq c$" に対して,

$$x_{L,n} - y_{L,n} \leq c$$

C_{t_L} 中のある不等式 "$X_L - Y_L \leq C_L$" に対して,

$$X'_{L,n} - Y'_{L,n} \geq C_L + 1$$

試験系列を得るには, 論理式 $\forall R'_{G,1}, R'_{L,1}\{\Phi_R \supset \Phi_{TST}\}$ を真とするような V を求める. 試験対象を FSM-C に限定し, 遷移条件, レジスタ代入式の右辺の形に制限をおいていることによりその計算は 3.1 節と同様に $O(lm)$ で解くことができる. ここで得られた整数解により, 入出力パラメータの値は一意に決まる. これにより, $IO_{TR} \cdot IO_{TG}$, $IO_{TR} \cdot IO_{TL}$ を決定できる. 以下では, それぞれ IO_G, IO_L と書く.

仕様では入出力 IO_G, IO_L の実行後はそれぞれ状態 $s_{G,m+1}$, $s_{L,n+1}$ に到達する. そこで, 3.1 節と同様に, 必要に応じて UIO 系列を連接することにより IOU_G, IOU_L を作成する.

このように作成した系列 IOU_G, IOU_L について以下の定理が成り立つ.

定理 3 IUT において試験対象となるアクション t 以外のアクションが正しく実装され, t 中の試験対象であるレジスタ代入式 $R \leftarrow X + C$ 以外の処理も正しく実装されているものとする. IUT に IO_G または IO_L を適用することにより, 初期状態から $IO_{-1} = \langle \mathrm{CMD}_{TR,-l}, ip_{TR,-l} \rangle / \langle \mathrm{RSP}_{TR,-l}, op_{TR,-l} \rangle \cdot \ldots \cdot \langle \mathrm{CMD}_{TR,-1}, ip_{TR,-1} \rangle / \langle \mathrm{RSP}_{TR,-1}, op_{TR,-1} \rangle$ を実行した後, $\langle \mathrm{CMD}_{TR,0}, ip_{TR,0} \rangle / \langle \mathrm{RSP}_{TR,0}, op_{TR,0} \rangle$ の実行時に, t のレジスタ代入式 "$R \leftarrow X + C$" に対して,

レジスタ R に $X_{TR,0}+C$ と異なる値を代入しているとする．このときIUTは IOU_G，IOU_L のいずれか一方を実行できない．

[証明] アクション t のレジスタ代入式 "$R \leftarrow X+C$" に対して，IUTではレジスタ R に $X_{TR,0}+C$ $(=R_{TR,1})$ と異なる値 $(R'_{TR,1})$ を代入しているとする．つまり，状態 s_1 でのレジスタ R の値 $(R'_{TR,1})$ は $R'_{TR,1} \leq R_{TR,1}-1$ もしくは $R'_{TR,1} \geq R_{TR,1}+1$ となる．そのとき，次の(i)，(ii)がそれぞれ成り立つ．

(i) $R'_{TR,1} \leq R_{TR,1}-1$ の場合：

$R'_{TR,1}=R'_{G,1}$ なので，$R'_{G,1} \leq R_{TR,1}-1$ となる．よって，適当な $R'_{L,1}$ を選ぶことにより Φ_{TR} は真となる．すると $\Phi_{TR} \supset \Phi_{TST}$ より Φ_{TST} も真となる．ここで，Φ_{TST} 中の $r'_{G,k}$ $(1 \leq k \leq m+1)$ は，試験対象のレジスタ代入で正しい値より小さな値を代入した場合の，その後の各状態における各レジスタの値を表している．そのため Φ_{TST} の構成における(c)より，$X'_{G,m}-Y'_{G,m} \geq C_G+1$ となり，TST_G の最後のアクション t_G の遷移条件式が満たされず t_G は実行できない．このとき，定理1の略証と同じ理由で，IOU_G を仕様どおりに実行できない．

以上より，IOU_G が実行できない．

(ii) $R'_{TR,1} \geq R_{TR,1}+1$ の場合：

(i)の場合と同様に IOU_L が実行できない．

よって，レジスタ R に $X_{TR,0}+C$ と異なる値を代入しているとき，IUTは IOU_G，IOU_L のいずれか一方を実行できない． □

以上より，作成した系列 IOU_G，IOU_L を試験系列として用いることによって，レジスタ R に仕様どおりの値が代入されていることを試験できる．

しかし，この系列を1つ適用するだけでは，アクション t の試験対象の代入操作 $R \leftarrow X+C$ が $R \leftarrow X'+C'$（X' と X は異なるレジスタ）と実現されているような誤ったIUTであっても，試験系列を入力，実行した際に，IO_{TR} の最後の入力値 ip_t を受けとった時点で $X+C=X'+C'$ という関係が成り立っていると，その誤りを検出できない．そこで，上で作成した連立不等式 Φ_R, Φ_{TST} に対して次の性質を満たす複数の解を求め，その解から決定される複数の系列 $IO_{G,1},\ldots,IO_{G,k_G}$ および $IO_{L,1},\ldots,IO_{L,k_L}$ を用いる．

試験対象がアクション t におけるレジスタ代入式 $R \leftarrow X+C$ であるとき，Z を X 以外のレジスタもしくは変数，$J \in \{G,L\}$ とすると，

すべての Z に対してある2つの入出力系列 IO_i, IO_j $(1 \leq i,j \leq k_J)$ が存在して，

$$IO_i\text{実行時の } X_0-Z_0 \neq IO_j\text{実行時の } X_0-Z_0 \quad (1)$$

となる．

ただし，状態 u_t でアクション t の入力値 ip_t を受け取った時点でつねに $X-Z=c$ (c：定数) となるレジスタ Z が存在する場合はこの限りではない．

このように作成した入出力系列 $IO_{G,1},\ldots,IO_{G,k_G}$ および $IO_{L,1},\ldots,IO_{L,k_L}$ について次の定理4が成り立つ．

定理4 $IO_{G,1},\ldots,IO_{G,k_G}$ にそれぞれUIO系列を連接した試験系列と，$IO_{L,1},\ldots,IO_{L,k_L}$ にそれぞれUIO系列を連接した試験系列をすべて適用することにより，アクション t の試験対象のレジスタ代入式 $R \leftarrow X+C$ が $R \leftarrow X'+C'$（X' と X は異なるレジスタ）と実現されている誤りを検出できる．

ただし，入力値 ip_t を受け取った時点でつねに $X-X'=c$ (c：定数) となる場合は正しい実現と見なす．

[証明] 試験対象のレジスタ代入操作 $R \leftarrow X+C$ が $R \leftarrow X'+C'$（X' と X は異なるレジスタ）と実現されている場合，状態 u_t でアクション t の入力値 ip_t を受け取った時点でつねに $X-X'=c$ (c：定数) となる場合を除くと，式(1)より $J \in \{G,L\}$ について，ある2つの入出力系列 IO_i, IO_j $(1 \leq i,j \leq k_J)$ が存在して，IO_i 実行時の $X_0-X'_0 \neq IO_j$ 実行時の $X_0-X'_0$ となる．ここで，IO_i を実行した際に $X_0+C=X'_0+C'$ が成り立ちその誤りを検出できない場合は，$X_0-X'_0=C'-C$ となる．その場合，IO_j を実行すると $X_0-X'_0 \neq C'-C$ となる．つまり，$X_0+C \neq X'_0+C'$ となり，レジスタ R に正しい値（X_0+C）とは異なる値を代入するので，定理3より，その誤りを検出できる． □

上記の条件を満たす試験系列をすべてIUTに適用すると，アクション t でレジスタ R に仕様どおりの式の値を代入する代入式として実現されていることを試験できる．

また，仕様ではアクション t であるレジスタ R に対する代入式がない（R が不変）場合に，$R \leftarrow R$ という代入式があると考えて，この代入式に対する試験系列を作成し適用することにより，t で仕様には存在しない R への代入式があるような誤りを検出できる．

以上より，各レジスタ代入式（$R \leftarrow R$ も考えて）に対して試験系列 $IO_{G,1},\ldots,IO_{G,k_G}$ にそれぞれ UIO 系列を連接した試験系列と，$IO_{L,1},\ldots,IO_{L,k_L}$ にそ

れぞれ *UIO* 系列を連接した試験系列を作成し，それらをすべて適用することにより，レジスタ代入式の単一誤りを検出できる．

5. システムの作成と実験

以上で述べた試験系列生成手法に基づいて，FSM-Cの各アクションの遷移条件式およびレジスタ代入式に対する試験系列を作成する試験系列生成システムを試作し[6]，実験を行った．作成したシステムは入力としてFSM-Cとして記述されたプロトコル仕様をとり，仕様中の個々の遷移条件およびレジスタ代入に対する試験系列を出力する．試験系列が存在しない場合，無限に探索を行いシステムが停止しなくなるため，一定の深さまでで探索を打ちきり，その深さまでで試験系列が存在しないものについてはその旨のメッセージを出力する．このシステムはC言語を用いて実装し，約3500行の規模である．また，このシステムに適用するプロトコルとして，OSIセションプロトコルの主要な6機能単位を選択したものを用いた．その規模は，下記のとおりである．

- 状態数：19
- レジスタ数：12
- アクション数：126（65個のエラーアクションを含む）
- 遷移条件式に現れる不等式の数：251
- レジスタ代入式の数：58

このようなFSM-Cを試験系列生成システムに適用したところ，すべての遷移条件およびレジスタ代入に対して，試験系列を生成できた．適用実験はSun SPARC STATION 4（メモリ32 MB）上で行った．実験結果を**表1**に示す．遷移条件に対する試験系列の平均系列長とは，遷移条件式中の個々の不等式に対する試験系列の平均の長さである．また，表の括弧内には，各系列におけるE-UIO系列の長さを示した．

今回適用した例プロトコルに現れる代入操作は，いずれも，その代入結果が出力に現れない．そのため，Wangらの手法ではではその正しさを試験する系列を生成できない．しかし，この適用実験ではすべてのレジスタ代入に対してその正しさを試験する系列を生成できた．したがって，それらのレジスタ操作に対して，本論文で提案した手法を用いることによりその正しさを試験する系列を自動作成できることを確認できた．

表1 例プロトコルに対する試験系列の作成結果
Table 1 Result of an experiment of test suit generation.

	遷移条件	レジスタ代入
総系列長	1067(243)	305(58)
平均系列長	4.2(1)	5.2(1)
総CPU実行時間（秒）	283.57	53.18

6. まとめ

本論文では，EFSMのサブクラスであるFSM-Cとしてモデル化されたプロトコルに対する適合性試験の手法として，そのレジスタ操作の正しさを試験するための試験系列とその生成手法を提案した．Wangらが提案した手法を用いた試験系列ではレジスタ代入操作の代入結果が出力に現れない場合にはその正しさを試験することができなかった．本論文で提案する手法で生成される試験系列を用いることでそのようなレジスタ代入操作の正しさを試験することができる．また，提案手法に基づく試験系列生成システムを作成し，実際に例プロトコルを用いて実験を行い，確かに試験系列を生成できることを確認した．

参考文献

1) Bosik, B.S. and Uyar, M.: Finite state machine based formal methods in protocol conformance testing: From theory to implementation, *Computer Network and ISDN Systems 22*, pp.7–33 (1991).
2) Li, X. Higashino, T. Higuchi, M. and Taniguchi, K.: Automatic Generation of Extended UIO Sequences for Communication Protocols in an EFSM Model, *Proc. 7th Int'l Workshop on Protocol Test Systems*, pp.213–228 (1994).
3) Wang, C.J. and Liu, M.T.: Axiomatic Test Sequence Generation for Extended Finite State Machines, *Proc. 12th Int'l Conf. on Distributed Computing Systems*, pp.252–259 (1992).
4) 中石敬治，樋口昌宏，藤井 護：あるクラスの拡張有限状態機械におけるレジスタ操作の試験系列の生成手法，情報処理学会マルチメディア通信と分散処理研究会 95-DPS-70-10 (1995).
5) Cormen, T.H. and Rivest, R.L.: *Introduction to Algorithms*, pp.539–543, MIT Press (1990).
6) 小原 勝：レジスタ代入操作を含むプロトコルにおける試験系列生成システムの実現，大阪大学基礎工学部情報工学科特別研究報告 (1996).

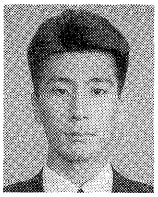
(平成 9 年 5 月 12 日受付)
(平成 10 年 1 月 16 日採録)

樋口　昌宏（正会員）

昭和58年大阪大学基礎工学部情報工学科卒業．昭和60年同大学大学院博士前期課程修了．昭和60～平成2年（株）富士通研究所勤務．平成3年大阪大学基礎工学部情報工学科助手．平成7年同講師．現在，同大学大学院基礎工学研究科情報数理系専攻講師．工学博士．通信プロトコル等の並行システムの形式的手法を用いた設計，検証，試験に関する研究に従事．

小原　　勝（学生会員）

平成8年大阪大学基礎工学部情報工学科卒業．現在，同大学大学院博士前期課程在学中．通信プロトコルの適合性試験に関する研究に従事．



中石　敬治

平成6年大阪大学基礎工学部情報工学科卒業．平成8年同大学大学院博士前期課程修了．現在，日本電信電話株式会社関西システム開発センタ勤務．在学中，通信プロトコルの適合性試験に関する研究に従事．

藤井　　護（正会員）

昭和37年大阪大学工学部電子工学科卒業．昭和39年同大学大学院修士課程修了．昭和39～42年三菱電機（株）勤務．昭和42年大阪大学基礎工学部制御工学科助手．昭和46年同大学基礎工学部情報工学科講師，昭和51年同大学大型計算機センタ助教授，昭和61年同大学基礎工学部情報工学科助教授，平成1年同大学教養部教授．平成6年同大学基礎工学部情報工学科教授，現在，同大学大学院基礎工学研究科情報数理系専攻教授．工学博士．