マルチメディア通信と分散処理 70−10
（1995. 5. 25）

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

中石 敬治 樋口 昌宏 藤井 護

大阪大学基礎工学部

e-mail: nakaishi@ics.es.osaka-u.ac.jp

拡張有限状態機械（EFSM）としてモデル化された通信プロトコルの適合性試験を行なうためには試験系列を作成する必要がある．本稿では，EFSM のサブクラスであるカウンタ付有限状態機械（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を自動で生成する手法を提案する．FSM-C モデルとは有限状態機械に，代入，整定数の加減算，大小比較の機能を持つ整数値レジスタを付けたプロトコル機械のモデルである．本手法を用いて作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について，試験対象のレジスタ操作が仕様通りに実現されていることを試験できる．また，OSI セションプロトコルの一部に対して本手法を適用した結果についても述べる．

A Method for Generating Test Cases for a Register Assignment of Models of a Subclass of EFSM

Takaharu Nakaishi Masahiro Higuchi Mamoru Fujii

Faculty of Engineering Science,
Osaka University

e-mail: {nakaishi, higuchi, fujii }@ics.es.osaka-u.ac.jp

Generating test cases is significant for conformance testing of communication protocols. In this paper, we propose a method for generating test cases for communication protocols modeled as FSM with counters (FSM-C), a subclass of Extended Finite-State Machines (EFSM). FSM-C has several integral registers. Operations on registers are restricted within assignments, additions and subtractions of an integer, and comparisons of two registers. By using proposing method, it is possible to generate a set of test cases that can detect a single fault of conditons and register assignments. Lastly we illustrate our method with an example.

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通りに動作するかどうかの試験は重要である（プロトコル適合性試験）．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通りであるかどうかを観察することによって行なう．試験系列の生成は従来人手により行なわれていた．しかし，試験効率をあ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）モデルとしてモデル化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その手法がほぼ確立されている．しかし，多くの通信プロトコルはFSMの有限制御部にレジスタを持たせた拡張有限状態機械（EFSM）としてモデル化される．EFSMに対する試験系列の生成手法として文献[1]では，状態の存在及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，各遷移で実行するアクションの正しさを試験する系列の生成手法は，Wangらが文献[2]において提案している．しかし，Wangらの手法ではレジスタ操作の一部において，その正しさを試験出来ないものがある．本稿ではEFSMの部分クラスであるカウンタ付FSMに対して，Wangらの手法では試験できなかった部分を補い，文献[1]の手法を利用してレジスタ操作の正しさを試験する試験系列の生成手法を提案する．

以降，2.では試験系列生成手法を提案するための準備を，3.でレジスタ操作の正しさを試験する系列について，そして4.で本稿で提案する手法を実際の例に適用した結果を示す．

2 準備

2.1 プロトコル機械モデル

本稿ではプロトコル機械のモデルとして拡張有限状態機械の部分クラスであるカウンタ付FSM（FSM-C）をとりあつかう．FSM-Cは，有限状態機械（FSM）に，代入，整定数の加減算，大小比較の機能を持つ整数値レジスタを付けたプロトコル機械のモデルである．FSM-Cでは，入力，条件判定，出力，レジスタ代入を実行できる．FSM-Cの状態遷移は，アクションの有限集合で定義される．図1は，FSM-Cモデルの例である．このプロトコルの説明は4.で行なう．

図1では，状態をノードで，アクションを枝で表している．それぞれの枝についているラベルは，そのアクションで実行する動作を表しており，(遷移条件式, 入力/出力, レジスタ代入式) という形式で表してある．

状態遷移は，あるアクションの遷移条件を満たす入力が行なわれると，出力，レジスタ代入の順でそのアクションの実行を行ない，次の状態に遷移する．入出力はそれぞれコマンドと整数値の対で行なわれ，入力は ?CMD,ip，出力は !RSP,$x+c$ と表される．遷移条件は，$x-y \leq c$ という形をした不等式からなる連立不等式であり，出力またはレジスタに代入される式は，$x+c$ という形の式に限る．ここで，CMD, RSP, ip, c はそれぞれ入力コマンド，出力コマンド，入力データを表す変数，整定数値を表し，x, y は入力データを表す変数またはレジスタ名を表す．

本稿で扱うプロトコル機械の仕様は以上のようなFSM-Cを用いて記述されているものとし，その状態遷移は完全かつ決定性で実行時間は有限であるものとする．そして，ある一つの状態を始状態とする状態遷移 $t_1, \cdots, t_n$ があるとき，任意の二つの状態遷移 t_i, t_j $(i \neq j, 1 \leq i, j \leq n)$ について，その終状態もしくは出力のいずれかが異なるものとする．また，IUTもFSM-Cモデルとして実現されているものとする．以上のように限定したモデルでも一連番号やタイマを扱うプロトコルをモデル化できる．

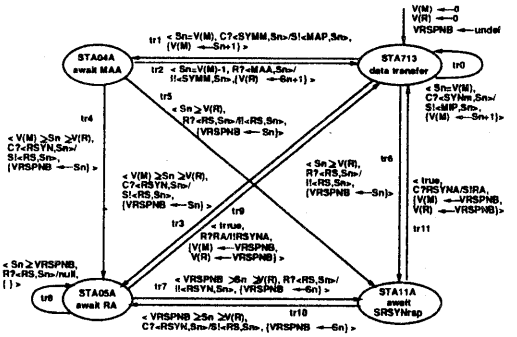

図1: OSIセションプロトコルの一部を抽出したプロトコル機械

2.2 ブラックボックス試験

適合性試験は通常ブラックボックス試験という方法で行なわれる．ブラックボックス試験では，IUTの内部状態が外部から観察できない．そこで，IUTに適当な入力系列を与え，その際の出力が仕様通りのものであるかを調べることによって試験を行なう．

2.3 E-UIO系列を用いた試験

状態を同定する系列として，E-UIO系列[1]がある．これはFSMにおいて一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において一つの状態を識別するには，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．

本稿で提案する手法において，ある状態 u のUIO系列を用いる際には，初期状態から状態 u までの系列が決定済みなので，その系列を先行系列として用いる．

3 レジスタ操作の正しさを試験するテスト系列

FSM-Cでは，アクション中の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，それらの正しさを試験するテスト系列について考える．以下では，FSM-Cモデルのプロトコル機械の仕様が与えられているものとする．

3.1 遷移条件の正しさの試験

アクションの各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．以下では試験の対象となるなるアクションを $t = \langle u, v, C_t, \langle \mathrm{CMD}, ip\rangle/\langle \mathrm{RSP}, op\rangle, R_t\rangle$ とする．すなわち 始状態が u，終状態が v，実行される動作が $\langle C_t, \langle \mathrm{CMD}, ip\rangle/\langle \mathrm{RSP}, op\rangle, R_t\rangle$ というアクション t を試験対象とする．アクション t が正しく IUT に実装されていることを確認するため，文献 [1] では以下の条件を満たすテスト系列対を生成することを提案している．

(1). 初期状態から u へ遷移させる入出力系列 IO に，状態 u を同定する系列 UIO_u を連接した系列．ただし，初期状態から IO を実行後，ある入力 $\langle \mathrm{CMD}, ip\rangle$ に対して t の遷移条件が満たされている．

(2). IO に $\langle \mathrm{CMD}, ip\rangle|\langle \mathrm{RSP}, op\rangle$，状態 v を同定する系列 UIO_v を連接した系列．ただし ip は t の遷移条件を満たす整数値であり，op はそのときの出力パラメータ．

しかし，上記の $IO{\cdot}UIO_u$, $IO{\cdot}\langle \mathrm{CMD}, ip\rangle|\langle \mathrm{RSP}, op\rangle{\cdot}UIO_v$ を用いた試験では IUT において 状態遷移 t の始状態と終状態がそれぞれ u と v であることが試験できるだけであり，t の遷移条件については IUT において初期状態から IO を実行したあとに遷移 t を実行するための条件が成立していることを保証しているのみである．IUT においてアクションの遷移条件が正しく判定されているかどうかを試験するための系列組の生成について，以下で議論する．

遷移条件が正しく判定されていることの基準として，IUT においてそれぞれのアクションの遷移条件が仕様とくらべてきつ過ぎないという条件を考える．あるアクション t の遷移条件が仕様よりきついものとなっている場合，ある入力に対して仕様では t を実行することになっているにもかかわらず，IUT では別のアクション t' を実行してしまう．例えば，C_t 中の差分制約 $x_1 - x_2 \le c$ が IUT において $x_1 - x_2 \le c'$ (ただし $c' < c$) と実装されている場合，$c' < x_1 - x_2 \le c$ であるとき，アクション t を実行すべきところを，IUT では他のアクションを実行してしまうことになる．仕様が決定性かつ完全であるという仮定より，すべてのアクションの遷移条件について IUT が仕様よりきつすぎないことは，すべてのアクションの遷移条件について IUT が仕様よりゆるすぎないことと同値であり，IUT においてすべてのアクションの遷移条件が正しく実装されていることを意味する．

以下では，遷移条件中の差分制約 $x_1 - x_2 \le c$ が IUT において $x_1 - x_2 \le c'$ (ただし $c' < c$) となっているような実装誤りを検出する試験系列について述べる．

C_t 中の差分制約 $x_1 - x_2 \le c$ に対して，初期状態から u へ遷移させる系列として以下のような入力系列 IO を選ぶ．

1. IO実行中にアクション t の実行は行なわれない．
2. 初期状態から IO を実行後，ある入力 $\langle \mathrm{CMD}, ip\rangle$ に対して $x_1 - x_2 = c$ 及び t の他の遷移条件を満たす．

仕様上で初期状態から IO を実行後，$x_1 - x_2 = c$ 及び t の他の遷移条件を満たす入力 $\langle \mathrm{CMD}, ip\rangle$ に対して，そのときの出力を $\langle \mathrm{RSP}, op\rangle$ とするとき，IOを用いて，以下の試験系列を考える．

AC1 t の他に u を始状態とし，入力コマンドが CMD，出力コマンドが RSP であるアクションが存在しない場合：IOに，$\langle \mathrm{CMD}, ip\rangle|\langle \mathrm{RSP}, op\rangle$ を連接した系列．

AC2 t の他に u を始状態とし，入力コマンドが CMD，出力コマンドが RSP であるアクションが存在する場合：IOに，$\langle \mathrm{CMD}, ip\rangle|\langle \mathrm{RSP}, op\rangle$ と状態 v を同定する系列 UIO_v を連接した系列．

定理 1 IUT において，遷移条件の判定以外はすべて正しく実装され，かつ試験対象となるアクション t を除くすべてのアクションの遷移条件が仕様と比べてきつすぎないものとする．このとき，アクション t の遷移条件 $x_1 - x_2 \le c$ が IUT において $x_1 - x_2 \le c'(c' < c)$ となっている場合，試験系列 AC1，AC2 によりその誤りを検出できる．

[証明] IO が t の実行を含まず，IUT において，遷移条件の判定以外はすべて正しく実装され，かつ試験対象となるアクション t を除くすべてのアクションの遷移条件が仕様と比べてきつすぎないことより，IUT は IO を仕様通りに実行することができ，実行後の状態も仕様通りのものとなっている．このとき，アクション t の遷移条件が仕様通り実装されていれば，AC1，AC2 を正しく実行することができる．アクション t の遷移条件 $x_1 - x_2 \le c$ が IUT において $x_1 - x_2 \le c'(c' < c)$ となっている場合，IUT は入力 $\langle \mathrm{CMD}, ip\rangle$ に対して他のアクション t' を実行する．この場合，仮定より t' の出力コマンド，終状態の少なくとも一方が t のものと異なる．出力コマンドが異なる場合は AC1 を正しく実行できず，終状態が異なる場合は AC2 を正しく実行できないので，その誤りを検出することができる． □

定理 1 より，試験系列 AC1，AC2 を IUT に適用すると遷移条件の単一誤りを検出することができる．

ここで，系列 IOを作成するには，遷移を幅優先探索して系列長の短いものから順に調べていけば良い．また，入出力値の決定のために連立不等式を解く計算は，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[4] により $O(lm)$ (但し lは不等式の数，m は変数の数) で解くことができる．

3.2 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方法として，Wang らは試験対象となるレジスタ代入で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[2]．しかし，現実のプロトコル仕様ではレジスタに代入された値がその後遷移の制御に用いられるのみで，出力値として現れるようなことがないような場合も多い．ここでは，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について提案する．

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．本稿で対象としているプロトコル機械のモデルでは遷移条件が不等式により与えられるので，正しい値より大きな値が代入された場合実行できない系列，正しい値より小さな値が代入された場合実行できない系列を用いて試験することを考える．すなわち，試験対象となるレジスタ代入式を アクションt のレジスタ代入式 "$R \leftarrow X+C$" として，初期状態から試験対象となるアクションが実行可能となる状態へ遷移し，試験対象となるアクションを実行させる入出力系列，及びそれに引き続く入出力系列として，アクションt でレジスタ R に $X+C$ が代入される場合は実行できるが，$X+C$ より小さな値が代入された場合実行できない系列，アクションt でレジスタ R に $X+C$ が代入される場合は実行できるが，$X+C$ より大きな値が代入された場合実行できない系列を用いて試験を行なうことを考える．

まず，初期状態から実行可能で，その最後にのみ試験対象であるアクションt をもつアクションの系列$TR = t_{-l} \cdot \ldots \cdot t_{-1} \cdot t_0$ ($-l \leq k \leq -1$ に対して，$t_k \neq t, t_0 = t$) を選ぶ．すなわち，以下の条件を満たすTRを選ぶ．ここで，ip_k, op_k はTR に沿ったある実行における，アクション t_k 実行における入力パラメータ，出力パラメータを表し，s_k , r_k はアクション t_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表すものとし，s_1 , r_1 は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ものとする．

1. 有限制御部の状態に関する条件:

   (a) $s_{-l} = s_{init}$,

   (b) $-l \leq k \leq 0$ について，

   $$s_k = u_{t_k},\ s_{k+1} = v_{t_k}.$$

2. 以下の不等式からなる連立不等式Φ_{TR}が整数解を持つ．

   (a) 各レジスタ r に対して

   $$r_{-l} = r_{init}$$

   (b) $-l \leq k \leq 0$ について：

   C_{t_k} 中の各不等式 "$x - y \leq c$" に対して，

   $$x_k - y_k \leq c,$$

   出力定義式 "$(\mathrm{RSP}_{t_k}, x + c)$" に対して，

   $$op_k = x_k + c,$$

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

   $$r_{k+1} - x_k = c,$$

ここで，Φ_{TR}の一つの整数解は TR に沿った一つの実行履歴を表すことになる．

次に，アクション t でレジスタ R に $X+C$ が代入される場合は実行できるが，$X+C$ より小さな値が代入された場合実行できないアクションの系列 $TST_G = t_{G,1} \cdot \ldots \cdot t_{G,m}$ ($1 \leq k \leq m$ に対して $t_{G,k} \neq t$), アクション t でレジスタ R に $X+C$ が代入される場合は実行できるが，$X+C$ より大きな値が代入された場合実行できない系列 $TST_L = t_{L,1} \cdot \ldots \cdot t_{L,n}$ ($1 \leq k \leq n$ に対して $t_{L,k} \neq t$), を選ぶ．以下では，特に $t_{G,m} = t_G, t_{L,n} = t_L$ と書くことにする．

TST_G, TST_Lについての条件は以下のように表すことができる．ここで，$Y \in \{G, L\}$ について，$ip_{Y,k}$ は TST_Y に沿ったある実行における，アクション $t_{Y,k}$ 実行における入力パラメータを表し，$s_{Y,k}$, $r_{Y,k}$ はアクション $t_{Y,k}$ 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表すものとし，$s_{G,m+1}$, $r_{G,m+1}$, $s_{L,n+1}$, $r_{L,n+1}$ はそれぞれ TST_G, TST_L 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．また，$Y \in \{G, L\}$ について，$r'_{Y,k}$, $ip'_{Y,k}$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，それぞれ正しい値より小さな値 ($Y = G$ の場合) あるいは正しい値より大きな値 ($Y = L$ の場合) を代入したときの，アクション $t_{Y,k}$ 実行前のレジスタ r の保持している値，アクション $t_{Y,k}$ 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する条件:

   (a) $s_{G,1} = s_{L,1} = s_1$,

   (b) $1 \leq k \leq m$ について，

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

   (c) $1 \leq k \leq n$ について，

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

2. 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．

   $$R'_{G,1} \leq R_1 - 1$$

   $$R'_{L,1} \geq R_1 + 1$$

   t_G の遷移条件 C_{t_G} 中のある不等式 "$X_G - Y_G \leq C_G$", t_L の遷移条件 C_{t_L} 中のある不等式 "$X_L - Y_L \leq C_L$" について，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，$\exists Q \forall R'_{G,1}, R'_{L,1} \{\Phi_R \supset \Phi_{TST}\}$ が解を持つ．ただし，Q とは $R'_{G,1}, R'_{L,1}$以外のレジスタ値を表す．

   (a) 入力パラメータが同じであることを表す条件:

   $$ip_{G,k} = ip'_{G,k} (1 \leq k \leq m)$$

   $$ip_{L,k} = ip'_{L,k} (1 \leq k \leq n)$$

(b) 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

$$r_{G,1} = r_1$$

$$r_{L,1} = r_1$$

R 以外の各レジスタ r に対して,

$$r'_{G,1} = r_1$$

$$r'_{L,1} = r_1$$

(c) $1 \leq k \leq m-1$ 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

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

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

(d) 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,

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

不等式 "$X_G - Y_G \leq C_G$" に対して,

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

(e) $1 \leq k \leq n-1$ 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

$$r_{L,k+1} - x_{L,k} = c,$$

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

(f) 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,

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

不等式 "$X_L - Y_L \leq C_L$" に対して,

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

上記の条件を満たす $TR \cdot TST_G, TR \cdot TST_L$ について論理式 $\exists Q \forall R'_{G,1}, R'_{L,1}\{\Phi_R \supset \Phi_{TST}\}$ を真とするような Q を求める. この論理式は, 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プレスブルガー一文となるため Q を求める計算は比較的容易である. FSM-C の定義より, 入力パラメータの値はアクション系列中の各状態でのレジスタの値により一意に決まるので, 上で求めた Q より入力パラメータを決定する. この入力パラメータは $TR \cdot TST_G$ と $TR \cdot TST_L$ に沿った二つの実行履歴 (それぞれ IO_G, IO_L とする) を表すことになる. また, 逆に出力パラメータの値とレジスタの値は, それ以前の入力パラメータにより一意に決まるので, $\exists Q \forall R'_{G,1}, R'_{L,1}\{\Phi_R \supset \Phi_{TST}\}$ を真とする解を指定するには, $ip_{-l}, \ldots, ip_0, ip_{G,1}, \ldots, ip_{G,m}, ip_{L,1}, \ldots, ip_{L,n}$, の値を指定すれば十分である.

次に, IO_G, IO_L にもとずいて, 文献 [1] で提案されたのと同様の方法により, $TR \cdot TST_G$ 実行後に状態 $s_{G,m+1}$ にいることを同定する系列 UIO_G と, $TR \cdot TST_L$ 実行後に状態 $s_{L,n+1}$ にいることを同定する系列 UIO_L を求める.

このように作成した系列 $IO_G \cdot UIO_G$, $IO_L \cdot UIO_L$ について以下の定理が成り立つ.

定理 2 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. ことのき IUT は $IO_G \cdot UIO_G$, $IO_L \cdot UIO_L$ のいずれか一方を実行できない.
[証明] 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 (R'_1) を代入しているとする. その時, 状態 s_1 でのレジスタ Rの値は $R_1 - 1$ 以下もしくは $R_1 + 1$ 以上の値となる. そして, 次の (i) もしくは (ii) が成り立つ.

(i) $R_1 - 1$ 以下の場合, アクション t の実行後のレジスタ Rの値を $R'_{G,1}$ とすると, $R'_{G,1} \leq R_1 - 1$ となる. よって, 適当な $R'_{L,1}$ を選ぶことにより Φ_R は真となる. すると, $\Phi_R \supset \Phi_{TST}$ 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間違った代入を行なった後の各状態における各レジスタの値を表していることになる. Φ_{TST} 中のレジスタ, 入力値に関する条件の (d) より, $X'_{G,m} - Y'_{G,m} \geq C_G + 1$ が成り立つことより, TST_G の最後のアクション t_G を実行できない. つまり TST_G は実行できない. ここで仮定より, $s_{G,m}$ から出る t_G 以外のアクションは, その出力が t_G の出力と異なるかその遷移先が t_G の遷移先 ($s_{G,m}$) とは異なる. そのため $s_{G,m}$ から t_G と出力が同じであるアクションを実行した場合は、$s_{G,m+1}$ が異なるため UIO_G が実行できない. 以上より, $IO_G \cdot UIO_G$ が実行できない.

(ii) R_1+1以上の場合も(i)の場合と同様に$IO_L \cdot UIO_L$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$IO_G \cdot UIO_G$, $IO_L \cdot UIO_L$ のいずれか一方を実行できない. □

以上より, 作成した系列 $IO_G \cdot UIO_G$, $IO_L \cdot UIO_L$ を試験系列として用いることによって, レジスタ Rに仕様通りの値が代入されていることを試験できる.

しかし, この系列を一つ適用するだけでは, 試験対象の代入操作 $R \leftarrow X+C$が $R \leftarrow X'+C'$ $(X' \neq X)$ 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, 状態 u_tでアクション t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_{TR}, Φ_{TST}に対して次の条件を満たす複数の解 $IO_1, IO_2, ..., IO_k$を求める. ただし, 試験対象を $R \leftarrow X+C$とし, Zを X以外のレジスタもしくはアクション tでの入力値とする.

[条件] すべてのZに対してある二つの試験系列 IO_i, IO_j $(1 \leq i,j \leq k)$ が存在して,

$$IO_i\text{実行時の } X_0 - Z_0 \neq IO_j\text{実行時の } X_0 - Z_0$$

となる.

このような試験系列 $IO_1, IO_2, ..., IO_k$を全て適用することにより, 試験対象のレジスタ代入操作が仕様通りのレジスタを用いて, 仕様通りの代入式として実現されていることを試験できる.

4 例への適用

ここでは, 図1に示したプロトコルモデルの例に対して本稿で提案した試験系列生成手法を適用した結果を述べる. 図1のプロトコル機械は, OSIセションプロトコル[3]のデータ転送フェーズで, カーネル機能単位, 全二重機能単位, 大同期機能単位, 小同期機能単位, 再同期機能単位の五つの機能単位を扱う. このプロトコル機械は, 次の四つのゲートを持つ.

C ：上位層からの入力を行なう.
I ：上位層に出力を行なう.
S ：下位層へ出力を行なう.
R ：下位層からの入力を行なう.

このプロトコル機械は, 以下のように簡単化されている.

- 小同期確認 PDUとレジスタ $V(A)$ は対象外とする.
- 対象とするプロトコルは大同期トークンと小同期トークンを持ち, これらのトークンは移動しない.
- 再同期の3オプションのうち再スタートのみを扱う.

このプロトコル機械に対して, 提案した手法を適用し, 遷移 tr_0から tr_{11}にある13個のレジスタ代入操作および, 遷移 tr_0から tr_{11}の各遷移条件についてそれぞれの正しさを試験する系列を作成した. その個数および長さは表1に示すとおりである.

	レジスタ代入	遷移条件
作成した系列の総数	44	13
各操作の平均系列数	3.4	1.1
一操作に対する最多系列数	4	2
総系列長	189	37
平均系列長	4.3	2.8
最長の系列	7	4

表 1: 例プロトコルに対する試験系列の作成結果

今回適用した例にある代入操作はいずれもその代入結果が出力に現れないので, Wangらの手法ではその正しさを試験することができない. そのようなレジスタ操作に対しても, 本稿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試験系列を自動で作成することができた.

5 まとめ

本稿では, EFSMのサブクラスであるFSM-Cモデ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らが提案する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や, 入力した試験系列によって誤った実現であるにもかかわらず仕様と同じ動作をしてしまう場合など, その正しさの試験ができない部分があった. 本稿で提案した手法を用いることにより, そのようなレジスタ操作の正しさを試験できるようになった. また, 本手法をOSIセションプロトコルの一部機能単位におけるレジスタ操作に対して適用し, 確かに試験系列を生成できることを示した.

今後, 提案する手法を実現し, いくつかの例題に適用してその有効性について議論する予定である.

参考文献

[1] X.Li, T.Higashino, M.Higuchi and K.Taniguchi: "Automatic Generation of Extended UIO Sequences for Communication Protocols in an EFSM Model", in Proc 7th Int'l Workshop on Protocol Test Systems, pp.213-228, Nov. 1994.

[2] C.J.Wang and M.T.Liu: "Axiomatic Test Sequence Generation for Extended Finite State Machines", in Proc 12th Int'l Conf. on Distributed Computing Systems, pp.252-259, June 1992.

[3] ISO: "Information Processing System - Open Systems Interconnection - Basic Connection Oriented Session Protocol Specification",IS 8327 , 1987.

[4] Cormen,T.H., et al.: "Introduction to Algorithms", The MIT Press, pp.539-543, 1990.