タイマシステムコールを用いるDFSMプロトコルに対する試験系列生成手法

森 亮憲[†] 徳田康平[†] 多田知正[†]
樋口昌宏[††] 東野輝夫[†]

本論文では，オペレーティングシステムのタイマ機能を利用するDFSMモデル通信プロトコルに対して，状態遷移におけるタイマ操作の単一誤りおよび終状態の単一誤りを検出する適合性試験系列生成手法を提案する．タイマ操作誤りについては，個々の誤りに対してプロトコルの仕様と実装が等価になるための十分条件と等価にならないための十分条件を与え，これらの条件を利用した試験系列生成手法を考えた．終状態誤りについては，Wp法を利用した試験系列生成手法を考えた．提案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実装し，DHCP（Dynamic Host Configuration Protocol）に適用した．その結果，試験系列が効率的に生成できることを確認した．

A Method to Generate Test Sequences for DFSM Protocol with Timer System Call

Takanori Mori,[†] Kohei Tokuda,[†] Harumasa Tada,[†]
Masahiro Higuchi[††] and Teruo Higashino[†]

In this paper, we propose a method to generate test sequences for communication protocols modeled as DFSM with timers. The test sequences can detect any single fault of timer commands or destination states in the transitions on protocol machines. For each single timer command fault, we give sufficient conditions that a given IUT is (or is not) equivalent to its specification. Based on these sufficient conditions, we give a method for generating test sequences. For each destination state fault, we give a test sequence generation method based on Wp-method. To show the usefulness of this method, we developed a system for generating test sequences, and applied it to DHCP (Dynamic Host Configuration Protocol). As a result, we generated the test sequences efficiently.

1. まえがき

近年，複数のコンポーネントが連携して動作するシステム（コンプレックスシステム）が使われるようになっている．コンプレックスシステムは，コンポーネント間のインタラクションが外部から観察・制御できない場合が多い[1]．

本論文では，コンプレックスシステムの一種として，オペレーティングシステム（OS）が提供するタイマ機能を利用してタイマ監視を行う通信プロトコルの適合性試験について議論する[2),3)]．本論文で扱うモデルは，プロトコル機械とタイマで構成される．プロトコル機械は決定性有限状態機械（DFSM）でモデル化され，システムコールによりOSのタイマ機能を利用する．プロトコル機械とタイマとのインタラクションは外部から観察・制御不可能である．

コンプレックスシステムの特定のコンポーネントに対して適合性試験を行うときは，システムを以下の2つの部分システムに分割する．(1) *Spec*：試験対象となる部分システム．(2) *Context*：システム全体のうち *Spec* 以外を表す部分システム．*Context* は正しく実装されていると仮定して試験を行う．試験対象となる実装（IUT: Implementation Under Test）と *Context* からなるシステムは，試験対象となるシステム（SUT: System Under Test）と呼ばれる．このような試験は，埋め込み型試験（embedded testing）[4)]やグレイボックス試験（gray box testing）[5)]と呼ばれる．埋め込み型試験をする場合は，以下のようなコンプレックスシステムの性質を考慮する必要がある．(1)

† 大阪大学大学院基礎工学研究科
Graduate School of Engineering Science, Osaka University
†† 近畿大学理工学部
School of Science and Engineering, Kinki University

IUTが *Spec* と異なっていてもSUTが *Spec·Context* と等価な動作をする場合がある．(2) IUTでの1つの誤りがSUTでは複数の誤りになる場合がある．

埋め込み型試験のためのさまざまな適合性試験系列生成手法が提案されている．これらの試験手法は，主としてSUTの各コンポーネントがDFSMでモデル化できる場合を対象としている[5)~7)]．本論文のモデルにおいて，タイマをFSMでモデル化すると状態数が膨大になる．そこで，これらの点を考慮して試験系列生成手法について議論する．

以降，2章では本論文で扱うモデルを説明する．3章と4章では試験系列生成手法について，5章ではDHCPに対して提案手法を適用した結果について述べる．

2. タイマシステムコールを用いるプロトコル

本論文では，複数のタイマを用いる通信プロトコル（**図1**）を考える．プロトコル機械とタイマとのインタラクションは，プロトコル機械からタイマへのタイマ操作（タイマの起動・解除）と，タイマからプロトコル機械へのタイムアウト通知の2種類である．タイマ操作はシステムコールである．たとえばLinuxでは `set_timer`, `del_timer` がある．タイムアウト通知はプロトコル機械に対するシグナルである．これらのインタラクションは外部から観察・制御できない．

2.1 タイマ

一般にOSのタイマ機能は複数のタイマを管理することができる．また，ユーザはシステムコールで個々のタイマを起動・解除することができる．タイマを起動してからタイムアウトが発生するまでの時間（タイムアウト時間）はシステムコールのパラメータで指定できる．タイマが起動されたあと，解除または再起動されることなくタイムアウト時間が経過すると，タイマがタイムアウトしてタイムアウト通知を行う．本論文では，各タイマごとにタイムアウト時間が決まっており，タイマを区別するために各タイマに対して1から順に番号がついていると考える．

タイマの仕様をベクトル T で表す．T の要素 $T[i]$ はタイマ i のタイムアウト時間を表す．タイマの状態をベクトル $\bar{\tau} = (\tau[1], \ldots, \tau[n])$ で表し，これをタイマ値ベクトルと呼ぶ．n はベクトル T の次元である．タイマ値ベクトルの要素 $\tau[i]$ はタイマ i のある時刻での値（タイマ値）を表す．$\tau[i]$ は，$0 \leq \tau[i] \leq T[i]$ の整数値および $\perp$ をとる．$\tau[i] = \perp$ は，タイマ i が起動されていないことを表す．任意の $x \in \mathcal{N}$（$\mathcal{N}$：非負整数）に対して $\perp > x$，$\perp - x = \perp$ とする．

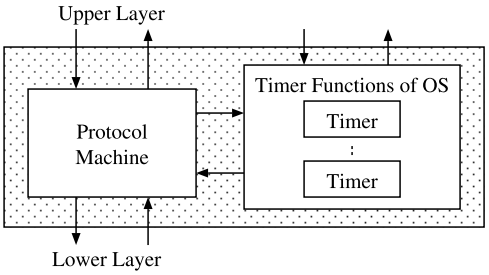


図1 タイマシステムコールを用いる通信プロトコル
Fig. 1 A model of protocols with timer system call.

タイマ i を起動または解除すると，$\tau[i]$ はそれぞれ $T[i]$，$\perp$ になる．各 $\tau[i]$ は単位時間ごとにいっせいに1ずつ減少し，$\tau[i]$ が0のとき，タイマ i はタイムアウト通知を行う．タイムアウト通知を行ったあと，タイマ値は $\perp$ になる．$\mathcal{T}$ でタイマ値ベクトルの集合 $\{\bar{\tau} \mid \tau[i] \in \{0, 1, \ldots, T[i], \perp\}\}$ を表す．

タイマ値が0のタイマがあると時間は経過しない．タイマ値が0であるタイマが複数あるときは，番号の若いタイマから順にタイムアウト通知を行う．タイマ値が0であるタイマ i が起動あるいは解除されると，タイマ値はそれぞれ $T[i]$，$\perp$ となり，そのタイマはタイムアウト通知を行わない．

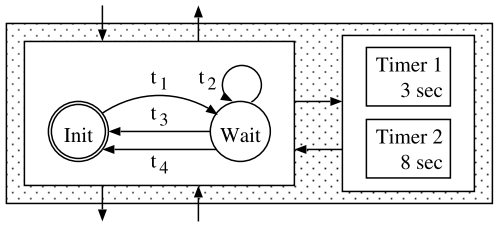
2.2 プロトコル機械

プロトコル機械はMealy型DFSM☆とし，6字組 (Q, X, n, Y, H, s_0) で定義する．

- Q ： 状態の有限集合
- X ： 外部入力記号の有限集合
- n ： 使用するタイマの数
- Y ： 外部出力記号の有限集合
- H ： 状態遷移 $(u, v, x, y, \bar{p})$ の有限集合
  - $u, v \in Q$：始状態，終状態
  - $x \in (X \cup \{1, \ldots, n\})$：入力
  - $y \in Y$：外部出力
  - $\bar{p} \in \langle \mathtt{S}, \mathtt{D}, \mathtt{N} \rangle^n$：タイマ操作ベクトル
- s_0 ： 初期状態

状態遷移は，プロトコル機械が入力（外部入力，またはタイムアウト通知）を受けたときに実行される．外部入力による状態遷移を外部入力遷移，タイムアウト通知による状態遷移をタイムアウト遷移と呼ぶ．状態遷移は，状態と入力から遷移後の状態と出力（外部出力とタイマ操作ベクトル）を一意に決定する．状態遷移は瞬時に完了すると仮定する．タイマ操作ベクト

☆ 状態と入力から出力が決定される決定性有限状態機械．



t1 : Send / Msg, (S, S) t3 : Ack / Finish, (D, D)
t2 : Timeout[1] / Msg, (S, N) t4 : Timeout[2] / Halt, (D, N)

図 2 タイムアウト再送を行う通信プロトコル
Fig. 2 Protocol re-transmitting a message when timeout occurs.

ルは，各要素が S，D，N のいずれかであるベクトル $\bar{p} = (p[1], \ldots, p[n])$ である．$p[i]$ はタイマ i への操作を表す．S はタイマを起動する操作，D はタイマを解除する操作を表す．また，N はタイマに対して何もしないことを表す．

プロトコル機械は最簡形☆であると仮定する．また，プロトコル機械は信頼できるリセット機能を持つと仮定する．プロトコル機械がリセット信号を受けると，すべてのタイマを解除して初期状態に戻る．状態遷移が定義されていない入力を受けると，エラーを表す外部出力を行いリセットと同様の動作をする．エラー遷移を含めるとプロトコル機械は完全定義☆☆である．

例 1 **図 2** は，タイムアウト再送を行う通信プロトコルである．通信相手にメッセージを送る際，タイマ 1 および 2 を起動する．Ack を受信するまでタイマ 1 のタイムアウトごとにメッセージを再送する．また，タイマ 2 がタイムアウトするとメッセージ再送を中止する．図中の Timeout[1] および Timeout[2] は，プロトコル機械へのタイムアウト通知を表している．◇

2.3 プロトコル機械とタイマからなる系

プロトコル機械 $M = (Q, X, n, Y, H, s_0)$ とタイマ仕様 T に対して，n と T の次元が等しい場合，$M \cdot T$ で M と T からなる系を表す．$M \cdot T$ の状態を状況と呼び，M の状態 $s \in Q$ とタイマ値ベクトル $\bar{\tau} \in \mathcal{T}$ の組 $\xi = \langle s, \bar{\tau} \rangle$ で表す．

$M \cdot T$ の初期状況は，M が初期状態ですべてのタイマが動作していない状況 $\xi_0 = \langle s_0, \perp^n \rangle$ とする．ただし x^n $(x \in \mathcal{N} \cup \{\perp\})$ は n 個の要素がすべて x であるベクトルを表す．状況 $\langle s, \bar{\tau} \rangle$ に対して，$\tau[i] = 0$, $\tau[j] > 0 (1 \le j < i)$ であるとき，$\langle s, \bar{\tau} \rangle$ をタイマ i がタイムアウトする状況という．タイマ i がタイムアウトする状況の集合を $\Gamma[i]$ と書く．M の状態 s と任意のタイマ値ベクトルの組からなる状況集合を $\langle s, * \rangle$ と書く．

$M \cdot T$ の状況遷移 $\eta = (\xi, \xi', x, y)$ の集合を Θ で表す．状況遷移には，以下の 3 種類がある．

- 外部入力遷移
  外部入力遷移 $(s, s', x, y, \bar{p}) \in H$ に対応する状況遷移は $(\langle s, \bar{\tau} \rangle, \langle s', \bar{\tau}' \rangle, x, y)$ である．ただし，$\tau[i] > 0$ $(1 \le i \le n)$ かつ

$$\tau'[i] = \begin{cases} T[i] & (p[i] = \mathtt{S}) \\ \tau[i] & (p[i] = \mathtt{N}) \\ \perp & (p[i] = \mathtt{D}) \end{cases}$$

- タイムアウト遷移
  タイムアウト遷移 $(s, s', \mathrm{Timeout}[k], y, \bar{p}) \in H$ に対応する状況遷移は $(\langle s, \bar{\tau} \rangle, \langle s', \bar{\tau}' \rangle, -, y)$ である．ただし，$\langle s, \bar{\tau} \rangle \in \Gamma[k]$ かつ

$$\tau'[i] = \begin{cases} T[i] & (p[i] = \mathtt{S}) \\ \tau[i] & (i \ne k \wedge p[i] = \mathtt{N}) \\ \perp & (p[i] = \mathtt{D} \vee (i = k \wedge p[i] = \mathtt{N})) \end{cases}$$

- 時間経過遷移
  時間経過による状況遷移は $(\langle s, \bar{\tau} \rangle, \langle s, \bar{\tau} - 1^n \rangle, -, -)$ である．ただし，$\tau[i] > 0 (1 \le i \le n)$.

M の状態遷移 t に対応する $M \cdot T$ の状況遷移集合を Θ_t で表す．また時間経過遷移の集合を Θ_e で表す．

状況遷移 $\eta = (\xi, \xi', x, y)$ に対して η の始状況 ξ，終状況 ξ'，入出力対 (x/y) をそれぞれ $\rho(\eta)$，$\delta(\eta)$，$IO(\eta)$ で表す．状況遷移系列 $r = \eta_1 \cdots \eta_m$ に対して，$\xi_1, \xi_2, \ldots, \xi_{m+1}$ が存在し，$k (1 \le k \le m)$ に対して $\rho(\eta_k) = \xi_k$ かつ $\delta(\eta_k) = \xi_{k+1}$ であるとき，r は ξ_1 で実行可能であるという．

例 2 図 2 のプロトコルの初期状況で以下の状況遷移系列が実行可能である．

$(\langle \mathrm{Init}, (0,0) \rangle, \langle \mathrm{Wait}, (3,8) \rangle, \mathrm{Send}, \mathrm{Msg})$
$(\langle \mathrm{Wait}, (3,8) \rangle, \langle \mathrm{Wait}, (2,7) \rangle, -, -)$
$(\langle \mathrm{Wait}, (2,7) \rangle, \langle \mathrm{Init}, (0,0) \rangle, \mathrm{Ack}, \mathrm{Finish})$ ◇

ここで，IO と δ を拡張し，状況遷移系列さらにその集合に対して適用できるようにする．たとえば δ を系列に適用した場合，系列の最後尾の状況遷移の終状態を表す．状況 ξ で実行可能な状況遷移系列の集合を $TS(\xi)$ で表す．$TS(\xi)$ のうち長さ n の状況遷移系列の集合を $TS_n(\xi)$ で表す．また，$\mathcal{R}(\xi)$ で状況 ξ から到達可能な状況集合 $\delta(TS(\xi))$ を表す．各記法で対象のプロトコル機械 M を明示するときは $TS_M(\xi)$ や $\mathcal{R}_M(\xi)$ のように書く．

☆ FSM に含まれるどの 2 つの状態も等価でない．
☆☆ すべての状態と入力の組合せに対して，出力と次の状態が指定されている．

2.4 時間オートマトンとの関係

時間を扱うプロトコルを記述するモデルとして時間オートマトン（Timed Automata）[8),9)]が知られている．時間オートマトンは複数のタイマを持つ．各タイマの値は時間経過とともに増加する．タイマの値に対する連立不等式を用いて各状態遷移に制限を付加できる．また，状態遷移で各タイマの値をリセットできる．時間オートマトンは，本論文で扱うモデルをシミュレートでき，時間オートマトンに対する解析手法が本論文のモデルに対しても有効である．

3. 単一誤りを検出する試験系列

FSMとタイマからなるプロトコルを試験するために，**図3**のような試験アーキテクチャを考える．このアーキテクチャでは，テスタもシステムコールを用いて時間を計測できる．一方，テスタはプロトコル機械とタイマとのインタラクションを観察・制御できない．また，テスタはプロトコル機械からの出力をすべて受け取ったあと，タイマからのタイムアウト通知を受け取ると仮定する．

3.1 フォールトモデル

プロトコル機械の各状態遷移について，以下のフォールトモデルを設定する．

- タイマ操作誤り
  状態遷移のタイマ操作ベクトルの1つの要素が仕様と異なる誤り．
- 終状態誤り
  状態遷移の終状態が仕様と異なる誤り．
- 外部出力誤り
  状態遷移の外部出力が仕様と異なる誤り．

本論文では単一誤りのみを検出対象とする．仕様 $M = (Q, X, n, Y, H, s_0)$ に対して，単一誤りを含む実装を (Q, X, n, Y, H', s_0) で表す．ただし，$H' = (H \setminus \{t\}) \cup \{t'\}$ であり，t と t' はタイマ操作ベクトルの1要素，または終状態，外部出力のいずれか1つのみが異なる．この実装を $M[t'/t]$ と書く．

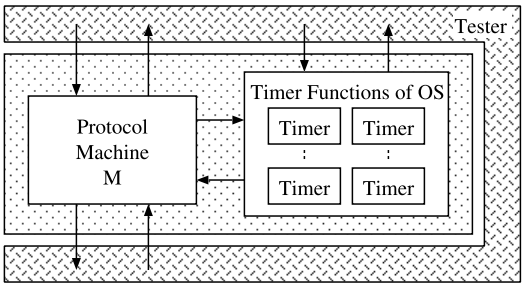


図3 試験アーキテクチャ
Fig. 3 Test architecture.

3.2 IO等価な実装

コンプレックスシステムは仕様 (S) と実装 (I) が異なっていても，コンテキスト (C) と合成したシステム $S \cdot C$ と $I \cdot C$ を外部から観察すると同じ動作をすることがある[5),6)]．これは，本論文で扱うモデルにもあてはまる．そこで，プロトコル機械 M_1，M_2 に対して以下の等価関係を定義する．

定義1 $M_1 \cdot T$ の状況 ξ_1 と $M_2 \cdot T$ の状況 ξ_2 に対して $IO(TS_{M_1 \cdot T}(\xi_1))$ と $IO(TS_{M_2 \cdot T}(\xi_2))$ が等しいとき，ξ_1 と ξ_2 はIO等価（$\xi_1 \equiv \xi_2$）であるという．△

定義2 $M_1 \cdot T$ と $M_2 \cdot T$ に対してそれぞれの初期状況がIO等価であるとき，$M_1 \cdot T$ と $M_2 \cdot T$ はIO等価（$M_1 \cdot T \equiv M_2 \cdot T$）であるという．△

仕様 M に対してタイマ操作誤りまたは終状態誤りを含む実装 $M[t'/t]$ を考えた場合，$M \cdot T$ と $M[t'/t] \cdot T$ がIO等価になることがある．一方，外部出力誤りを含む実装 $M[t'/t]$ は，誤りを含む状態遷移が実行不可能な場合を除いて $M \cdot T$ と $M[t'/t] \cdot T$ がIO等価となることはない．

例3 **図4**の仕様 M に対して実装 $M[t_1'/t_1]$ を考える．M の初期状態 s_1 でのタイマ値ベクトルは $(\perp, \perp)$ である．また，状態遷移 t_3 で状態 s_1 に遷移したあとのタイマ値ベクトルも $(\perp, \perp)$ である．一方 $M[t_1'/t_1]$ においても，s_1 でのタイマ値ベクトルはつねに $(\perp, \perp)$ である．

$M \cdot T$ と $M[t_1'/t_1] \cdot T$ の初期状況で $\eta_1 \in \Theta_{t_1}, \eta_1' \in \Theta_{t_1'}'$ を実行すると，それぞれ異なる状況 $\langle s_2, (3, \perp) \rangle$，$\langle s_2, (3, 8) \rangle$ へ遷移する．しかしタイマ1のタイムアウトにより，ともに $\langle s_3, (\perp, 8) \rangle$ へ到達する．これらの動作における入出力系列は一致する．このため $M \cdot T$ と $M[t_1'/t_1] \cdot T$ はIO等価である．◇

3.3 誤りを検出する入出力系列

仕様 M と実装 $M[t'/t]$ に対して，$M \cdot T$ と $M[t'/t] \cdot T$ の等価性を判定するには，$IO(TS_{M \cdot T}(\xi_0))$

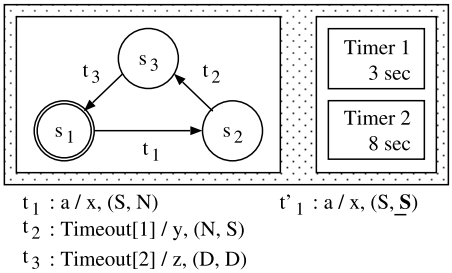


図4 誤りを含んでいても等価な動作をする実装
Fig. 4 Faulty implementation equivalent to specification.

および $IO(TS_{M[t'/t]\cdot T}(\xi_0))$ を受理する FSM☆をそれぞれ生成し，等価性を判定すればよい．これらの FSM は，時間オートマトンからリージョンオートマトンを生成する方法[8]と同様の方法で生成できる．

プロトコル機械は決定性で完全定義であると仮定している．よって，$M\cdot T$ と $M[t'/t]\cdot T$ が IO 等価でないならば，$M\cdot T$ の初期状況で実行可能であり，$M[t'/t]\cdot T$ の初期状況で実行可能でない入出力系列 io が少なくとも 1 つ存在する．また，等価性判定によってこのような io が得られる．系列 io は $M[t'/t]$ の誤りを検出する入出力系列である．

3.4 試験系列の生成

io を 3.3 節の等価性判定で得られた入出力系列であるとする．io は $M\cdot T$ に対する系列である．そこで，テスタが入力および出力のタイミングを計測できるように io を変換する．タイムアウト遷移は，1 回以上の時間経過遷移のあとに発生するので，$M\cdot T$ で実行可能な状況遷移系列はすべて，正規表現 $\{(\Theta_e^*\Theta_x)|(\Theta_e^+\Theta_o^+)\}^*\Theta_e^*$ で表される系列集合の要素である☆☆．ここで，Θ_e は時間経過遷移の集合，Θ_x は外部入力遷移の集合，Θ_o はタイムアウト遷移の集合を表す．次のように入出力系列から試験系列へ変換する．

- 状況遷移系列 $\Theta_e^*\Theta_x$ に対応する入出力系列

$$(-/-)^n(x/y) \quad \Longrightarrow \quad \begin{cases} (x/y) & n=0 \\ (\mathrm{Set}(n)/-)(\mathrm{WE/TO})(x/y) & n>0 \end{cases}$$

  $\mathrm{Set}(n)$ はテスタがタイマを n に設定することを表す．WE（Wait Enough）はテスタが何もしないで出力があるまで待つことを表す．TO は $\mathrm{Set}(n)$ に対応するタイムアウト通知である．テスタが TO を受け取れば $\mathrm{Set}(n)$ を行ってから n 単位時間経過したことが保証される．

- 状況遷移系列 $\Theta_e^+\Theta_o^+$ に対応する入出力系列

$$(-/-)^n(-/-)(-/y_1)\cdots(-/y_m) \quad \Longrightarrow \quad \begin{cases} (\mathrm{Set}(1)/-)(\mathrm{WE}/y_1)\cdots \\ \quad (\mathrm{WE}/y_m)(\mathrm{WE/TO}) & n=0 \\ (\mathrm{Set}(n)/-)(\mathrm{WE/TO}) & \\ \quad (\mathrm{Set}(1)/-)(\mathrm{WE}/y_1)\cdots & n>0 \\ \quad\quad (\mathrm{WE}/y_m)(\mathrm{WE/TO}) & \end{cases}$$

  ここで，Θ_e^+ は $\Theta_e^*\Theta_e$ と扱う．3 章で述べたように，テスタはプロトコル機械からの出力を受け取ったあと，Set(1) に対応するタイムアウトを受け取る．

- 最後尾の状況遷移系列 Θ_e^* に対応する入出力系列

$$(-/-)^n \Longrightarrow (\mathrm{Set}(n)/-)(\mathrm{WE/TO}) \quad n>0$$

以上の変換を行う手間は入出力系列長の定数倍である．

例 4 図 2 で，遷移 t_1 を t_1' と実装したとする．

$t_1 = (\mathrm{Init}, \mathrm{Wait}, \mathrm{Send}, \mathrm{Msg}, (\mathtt{S}, \mathtt{S}))$

$t_1' = (\mathrm{Init}, \mathrm{Wait}, \mathrm{Send}, \mathrm{Msg}, (\mathtt{N}, \mathtt{S}))$

この誤りを検出する入出力系列は，

$(\mathrm{Send/Msg})(-/-)(-/-)(-/-)(-/\mathrm{Msg})$

これに対応する試験系列は，次のようになる．

$(\mathrm{Send/Msg})(\mathrm{Set}(2)/-)(\mathrm{WE/TO})$

$(\mathrm{Set}(1)/-)(\mathrm{WE/Msg})(\mathrm{WE/TO})$

誤実装でこの系列を実行すると 2 個目の Msg が出力されない． ◇

4. 効率的な試験系列の生成

3.3 節で述べたように，仕様 M に対して単一誤りを含む実装 $M[t'/t]$ をすべて生成し，$M\cdot T$ と $M[t'/t]\cdot T$ の等価性を判定することで，任意の単一誤りを検出する試験系列を生成できる．しかし，この方法では手間の大きさが問題になる．そこで $M\cdot T$ よりも小さな機械を解析すること，または複数の誤りを一括して扱うことで効率良く試験系列を生成することを考える．

外部出力誤りは，状態遷移を実行し外部出力を観察することで検出できる．よって，以降ではタイマ操作誤りおよび終状態誤りを検出する試験系列について考える．どちらの誤りに対しても，最初に $M\cdot T$ の到達可能状況を解析し，$M\cdot T$ が初期状態から到達可能な各状況へ到達する状況遷移系列を求めておく．

4.1 タイマ操作の誤りに対する試験系列

仕様 M の状態遷移 $t=(u,v,x,y,\bar{p})\in H$ におけるタイマ i に対する操作 $p[i]$ について試験することを考える．M に対してタイマ操作誤りを 1 つ含む実装 $M[t'/t]$ を考える．t と t' は $p[i]$ のみが異なる．

より小さな機械を解析することで判定できる $M\cdot T$ と $M[t'/t]\cdot T$ が IO 等価になる十分条件と IO 等価にならない十分条件について議論する．このため，状況に対して次のような関係を考える．

定義 3 2 つの状況 ξ と ξ' に対して，タイマ i のタイマ値のみが異なるとき $\xi_{<i>}\xi'$ と書く． △

$\eta\in\Theta_t$，$\eta'\in\Theta'_{t'}$ かつ $\rho(\eta)=\rho(\eta')$ である状況遷移 η，η' に対して，$\delta(\eta)_{<i>}\delta(\eta')$ または $\delta(\eta)=\delta(\eta')$ である．$\xi_{<i>}\xi'$ なる 2 つの状況からのタイマ i によるタイムアウト遷移以外の状況遷移 $\eta\in\Theta_z$，$\eta'\in\Theta'_z$ を考

☆ 入出力系列の入力系列を与えたときに，対応する出力系列を出力する FSM.

☆☆ A^* は A の要素の 0 回以上の繰返し，A^+ は A の要素の 1 回以上の繰返しを表す.

表 1 $TS_{M\cdot T}(\xi_0) \overset{t,t'}{=} TS_{M[t'/t]\cdot T}(\xi_0)$ の必要十分条件
Table 1 Necessary and sufficient condition for $TS_{M\cdot T}(\xi_0) \overset{t,t'}{=} TS_{M[t'/t]\cdot T}(\xi_0)$.

$p[i]$	$p'[i]$	タイマ値ベクトル	条件		
S	N	$\bar{\tau} \in \Phi_{M[t'/t]\cdot T}(u)$	$\tau[i] \neq \perp$	$\mathcal{R}_{M[t'/t]_{\mathtt{N}}\cdot T}(\delta(\eta)) \cap \Gamma[i] = \emptyset$	$(\eta \in \Theta'_{t'} \wedge \rho(\eta) = \langle u, \bar{\tau}\rangle)$ (i)
			$\tau[i] = \perp$	$\mathcal{R}_{M_{\mathtt{N}}\cdot T}(\delta(\eta)) \cap \Gamma[i] = \emptyset$	$(\eta \in \Theta_t \wedge \rho(\eta) = \langle u, \bar{\tau}\rangle)$ (ii)
N	S	$\bar{\tau} \in \Phi_{M\cdot T}(u)$	$\tau[i] \neq \perp$	$\mathcal{R}_{M_{\mathtt{N}}\cdot T}(\delta(\eta)) \cap \Gamma[i] = \emptyset$	$(\eta \in \Theta_t \wedge \rho(\eta) = \langle u, \bar{\tau}\rangle)$
			$\tau[i] = \perp$	$\mathcal{R}_{M[t'/t]_{\mathtt{N}}\cdot T}(\delta(\eta)) \cap \Gamma[i] = \emptyset$	$(\eta \in \Theta'_{t'} \wedge \rho(\eta) = \langle u, \bar{\tau}\rangle)$
N	D	$\bar{\tau} \in \Phi_{M\cdot T}(u)$	$\tau[i] \neq \perp$	$\mathcal{R}_{M_{\mathtt{N}}\cdot T}(\delta(\eta)) \cap \Gamma[i] = \emptyset$	$(\eta \in \Theta_t \wedge \rho(\eta) = \langle u, \bar{\tau}\rangle)$
D	N	$\bar{\tau} \in \Phi_{M\cdot T}(u)$	$\tau[i] \neq \perp$	$\mathcal{R}_{M[t'/t]_{\mathtt{N}}\cdot T}(\delta(\eta)) \cap \Gamma[i] = \emptyset$	$(\eta \in \Theta'_{t'} \wedge \rho(\eta) = \langle u, \bar{\tau}\rangle)$
D	S	$\bar{\tau} \in \Phi_{M\cdot T}(u)$	$\mathcal{R}_{M[t'/t]_{\mathtt{N}}\cdot T}(\delta(\eta)) \cap \Gamma[i] = \emptyset$		$(\eta \in \Theta'_{t'} \wedge \rho(\eta) = \langle u, \bar{\tau}\rangle)$
S	D	$\bar{\tau} \in \Phi_{M\cdot T}(u)$	$\mathcal{R}_{M_{\mathtt{N}}\cdot T}(\delta(\eta)) \cap \Gamma[i] = \emptyset$		$(\eta \in \Theta_t \wedge \rho(\eta) = \langle u, \bar{\tau}\rangle)$

える．状態遷移 z のタイマ i に対する操作が S または D であれば，$\delta(\eta) = \delta(\eta')$ である．一方，タイマ i に対する操作が N であれば，依然として $\delta(\eta) <i> \delta(\eta')$ である．このように，$\xi <i> \xi'$ なる2つの状況を区別するにはタイマ i に対する操作が N の状態遷移を考えれば十分である．そこで，M に対して $M_{\mathtt{N}} = (Q, X, n, Y, H_{\mathtt{N}}, s_0)$ を考える．ただし，$H_{\mathtt{N}} = \{z = (s, s', a, b, \bar{p}) \mid (z \in H) \wedge (p[i] = \mathtt{N}) \wedge (a \neq \mathrm{Timeout}[i])\}$．

ここで状況遷移系列に対して次の関係を導入する．

定義 4 M の状態遷移集合を $H = \{t, t_1, \ldots, t_k\}$，$M[t'/t]$ の状態遷移集合を $H' = \{t', t_1, \ldots, t_k\}$ とする．$TS_{M\cdot T}(\xi_0) \cup TS_{M[t'/t]\cdot T}(\xi_0)$ に含まれる状況遷移系列 $q(= q_1 \cdots q_m)$，$r(= r_1 \cdots r_n)$ に対して，以下の条件が成り立つならば q と r が $\{t, t'\}$-等価（$q \overset{t,t'}{=} r$）であるという☆．

(1) $m = n$

(2) $\forall i \in \{1, \ldots, m\}$
$q_i \in (\Theta_{t_j} \cup \Theta'_{t_j}) \Leftrightarrow r_i \in (\Theta_{t_j} \cup \Theta'_{t_j})$ $(1 \leq j \leq k)$

(3) $\forall i \in \{1, \ldots, m\}$
$q_i \in (\Theta_t \cup \Theta'_{t'}) \Leftrightarrow r_i \in (\Theta_t \cup \Theta'_{t'})$

(4) $\forall i \in \{1, \ldots, m\}$
$q_i \in (\Theta_e \cup \Theta'_e) \Leftrightarrow r_i \in (\Theta_e \cup \Theta'_e)$

ただし，Θ_t は状態遷移 t に対応する状況遷移の集合を，Θ_e は時間経過遷移の集合を表す． △

$\{t, t'\}$-等価を拡張し状況遷移系列の集合間の関係とする．

定義 5 状況遷移系列の集合 A，B について，$\forall a \in A\ \exists b \in B\ \{a \overset{t,t'}{=} b\} \wedge \forall b \in B\ \exists a \in A\ \{b \overset{t,t'}{=} a\}$ であるとき，A と B が $\{t, t'\}$-等価（$A \overset{t,t'}{=} B$）であるという． △

状況遷移系列は一意に入出力系列に変換され，Θ_t，$\Theta_{t'}$ に属する状況遷移は同じ入出力系列に変換されるので，$\{t, t'\}$-等価に対して次の補題が成り立つ．

補題 1 $M \cdot T$，$M[t'/t] \cdot T$ に対して $TS_{M\cdot T}(\xi_0) \overset{t,t'}{=} TS_{M[t'/t]\cdot T}(\xi_0)$ ならば $M \cdot T \equiv M[t'/t] \cdot T$． □

長さ1の状況遷移系列に対して次の補題が成立する．

補題 2 $M \cdot T$，$M[t'/t] \cdot T$ の状況 ξ，ξ' について $\xi = \xi'$ または $\xi <i> \xi' \wedge (\xi, \xi' \notin \Gamma[i])$ であるとき $TS_{M\cdot T,1}(\xi) \overset{t,t'}{=} TS_{M[t'/t]\cdot T,1}(\xi')$．

略証 $\xi = \xi'$ のときは明らか．$\xi <i> \xi' \wedge (\xi, \xi' \notin \Gamma[i])$ であるときを考える．$\xi \in \Gamma[j]$ なる $j(\neq i)$ が存在すれば，$\xi <i> \xi'$ なので $\xi' \in \Gamma[j]$．また，ξ と ξ' のプロトコル機械の状態は一致するので，ξ におけるタイマ j によるタイムアウト遷移と，ξ' におけるタイマ j による遷移は $\{t, t'\}$-等価である．

一方，$\xi \in \Gamma[j]$ なる $j(\neq i)$ が存在しなければ，$\xi' \in \Gamma[j]$ となる j も存在しない．このとき，外部入力遷移と時間経過遷移が実行できる．$\xi <i> \xi'$ なので，ξ，ξ' で実行可能な長さ1の状況遷移の集合は $\{t, t'\}$-等価である． □

次に，$TS_{M\cdot T}(\xi_0)$ と $TS_{M[t'/t]\cdot T}(\xi_0)$ が $\{t, t'\}$-等価になる必要十分条件を考える．**表 1** に示す各条件について，以下の補題が成立する．表1で $\Phi_{M\cdot T}(s)$ は，M の状態が s のときにタイマがとりうるタイマ値ベクトルの集合（$\mathcal{R}_{M\cdot T}(\xi_0) \cap \langle s, * \rangle$ に含まれるタイマ値ベクトルの集合）を表す．表1の条件 (i) は，$p[i] = \mathtt{S}$，$p'[i] = \mathtt{N}$ のとき $M[t'/t]_{\mathtt{N}} \cdot T$ が状況 $\langle u, \bar{\tau}\rangle$ $(\bar{\tau} \in \Phi_{M[t'/t]\cdot T}(u) \wedge \tau[i] \neq \perp)$ で $\Theta'_{t'}$ に属する状況遷移で遷移したあと，タイマ i がタイムアウトする状況 $(\in \Gamma[i])$ に到達しないことを表している．

補題 3 M，$M[t'/t]$ の t，t' における $p[i]$，$p'[i]$ と表1の $p[i]$，$p'[i]$ が一致する行の「タイマ値ベクトル」に含まれる各ベクトルに対して「条件」が成立するときかつそのときのみ，以下が成り立つ．

(1) $TS_{M\cdot T}(\xi_0) \overset{t,t'}{=} TS_{M[t'/t]\cdot T}(\xi_0)$

☆ $\{t, t'\}$-等価に対して反射律，対称律，推移律が成り立つように定義した．本論文の議論では，$q \in TS_{M\cdot T}(\xi_0)$，$r \in TS_{M[t'/t]\cdot T}(\xi_0)$ の場合のみを使用する．

(2) $r \in TS_{M\cdot T}(\xi_0)$, $r' \in TS_{M[t'/t]\cdot T}(\xi_0)$, $r \overset{t,t'}{=} r'$ である状況遷移系列について, $\delta(r) = \delta(r')$ または $\delta(r) <_i> \delta(r') \wedge (\delta(r), \delta(r') \notin \Gamma[i])$.

略証 ($\Rightarrow$) $p[i] = \mathtt{S}$, $p'[i] = \mathtt{N}$ のときについて述べる.

状況遷移系列長に対する帰納法を用いて証明する. 初期状況で実行可能な長さ m の状況遷移系列 r, r' に対して $\delta(r) = \xi_m = \langle s_m, \bar{\tau}_m \rangle$, $\delta(r') = \xi'_m = \langle s'_m, \bar{\tau}'_m \rangle$ とする. 基底段階 $m = 0$ の場合, $M\cdot T$ と $M[t'/t]\cdot T$ の初期状況は一致するので $\xi_0 = \xi'_0$, また初期状況で実行可能な長さ 0 の状況遷移は存在しないので $TS_{M\cdot T,0}(\xi_0) \overset{t,t'}{=} TS_{M[t'/t]\cdot T,0}(\xi_0) = \emptyset$. 帰納段階として $m = k$ の場合,

$$TS_{M\cdot T,k}(\xi_0) \overset{t,t'}{=} TS_{M[t'/t]\cdot T,k}(\xi_0) \tag{3.1}$$
$$\xi_k = \xi'_k \vee \xi_k <_i> \xi'_k \wedge (\xi_k, \xi'_k \notin \Gamma[i]) \tag{3.2}$$

が成立していると仮定して $m = k+1$ について,

$$TS_{M\cdot T,k+1}(\xi_0) \overset{t,t'}{=} TS_{M[t'/t]\cdot T,k+1}(\xi_0) \tag{3.3}$$
$$\xi_{k+1} = \xi'_{k+1} \vee \xi_{k+1} <_i> \xi'_{k+1} \wedge (\xi_{k+1}, \xi'_{k+1} \notin \Gamma[i]) \tag{3.4}$$

が成立していることを示せばよい.

(3.2) と補題 2 より $TS_{M\cdot T,1}(\xi_k) \overset{t,t'}{=} TS_{M[t'/t]\cdot T,1}(\xi'_k)$ である. また (3.1) が成立しているので, (3.3) が成り立つ. 以降は $\eta_{k+1} \in TS_{M\cdot T,1}(\xi_k)$ と $\eta'_{k+1} \in TS_{M\cdot T,1}(\xi'_k)$ を $\{t, t'\}$-等価な状況遷移として, (3.4) が成立していることを示す.

- $\xi_k = \xi'_k$ のとき

  $\eta_{k+1} \in \Theta_e$ の場合, $\eta_{k+1} \overset{t,t'}{=} \eta'_{k+1}$ としているので $\eta'_{k+1} \in \Theta'_e$. よって $\xi_{k+1} = \xi'_{k+1}$.

  $\eta_{k+1} \in \Theta_z (z \in H \setminus \{t\})$ の場合, $\eta'_{k+1} \in \Theta'_z (z \in H' \setminus \{t'\})$ なので $\xi_{k+1} = \xi'_{k+1}$.

  $\eta_{k+1} \in \Theta_t$ の場合 $\eta'_{k+1} \in \Theta'_{t'}$. $\tau'_k[i] = T[i]$ であれば $p[i] = \mathtt{S}$, $p'[i] = \mathtt{N}$ なので $\xi_{k+1} = \xi'_{k+1}$. また $\tau'_k[i] \neq T[i]$ であれば $\xi_{k+1} <_i> \xi'_{k+1} \wedge \xi_{k+1} \notin \Gamma[i]$. 以下では $\xi'_{k+1} \notin \Gamma[i]$ を示す. $\tau'_k[i] = \perp$ のときは, $p'[i] = \mathtt{N}$ なので $\xi'_{k+1} \notin \Gamma[i]$. $\tau'_k[i] < T[i]$ のときは, $\delta(\eta'_{k+1}) = \xi'_{k+1} \in \mathcal{R}_{M[t'/t]_{\mathtt{N}}}(\delta(\eta'_{k+1}))$. また $\eta'_{k+1} \in \Theta'_{t'}$, $\rho(\eta'_{k+1}) = \xi'_k$, $\tau'_k \in \Phi_{M[t'/t]\cdot T}(u)$ なので, 条件 (i) より $\xi'_{k+1} \notin \Gamma[i]$.
- $\xi_k <_i> \xi'_k \wedge (\xi_k, \xi'_k \notin \Gamma[i])$ のとき

  $\eta_{k+1} \in \Theta_z (z \in H \setminus \{t\})$ の場合 $\eta'_{k+1} \in \Theta'_z$. ここで $z = (s, s', a, b, \bar{q})$ とする. $q[i]$ が $\mathtt{S}$ または $\mathtt{D}$ であるときは $\xi_{k+1} = \xi'_{k+1}$. 一方 $q[i]$ が $\mathtt{N}$ のときは $\xi_{k+1} <_i> \xi'_{k+1}$. 以下で $\xi_{k+1}, \xi'_{k+1} \notin \Gamma[i]$ を示す. $r \overset{t,t'}{=} r'$, $r \in TS_{M\cdot T}(\xi_0)$ かつ $\delta(r) = \xi_k$, $r' \in TS_{M[t'/t]\cdot T}(\xi_0)$ かつ $\delta(r') = \xi'_k$ なる状況遷移系列 $r (= \eta_1 \cdots \eta_k)$, $r' (= \eta'_1 \cdots \eta'_k)$ を考える. $M\cdot T$, $M[t'/t]\cdot T$ の初期状況は同じであり $\xi_k <_i> \xi'_k$ なので, $\eta_l \in \Theta_t (1 \le l \le k)$ かつ $\eta_j \in \Theta_{\mathtt{N}} (l < j \le k)$, $\eta'_l \in \Theta'_{t'} (1 \le l \le k)$ かつ $\eta'_j \in \Theta'_{\mathtt{N}} (l < j \le k)$ であるような l が存在する. ただし, $\Theta_{\mathtt{N}}$ は $M_{\mathtt{N}}\cdot T$ の状況遷移集合を表す.

  – $\tau'_{l-1}[i] = \perp$ のとき

    $p'[i] = \mathtt{N}$, $\eta'_j \in \Theta'_{\mathtt{N}} (l < j \le k)$ であり, z において $q[i] = \mathtt{N}$ であるので $\tau'_{k+1}[i] = \perp$. よって $\xi'_{k+1} \notin \Gamma[i]$.

    次に, ξ_{k+1} について考える. $\zeta \in \Theta_t$ かつ $\rho(\zeta) = \xi'_{l-1}$ である状況遷移 ζ を考える. $\eta_l \in \Theta_t$ かつ $p[i] = \mathtt{S}$ なので$\delta(\eta_l) = \delta(\zeta)$. また l の決め方と $q[i] = \mathtt{N}$ であることから $\xi_{k+1} \in \mathcal{R}_{M_{\mathtt{N}}\cdot T}(\delta(\eta_l))$. さらに, $\tau'_{l-1} \in \Phi_{M[t'/t]\cdot T}(u)$ でありかつ, $\tau'_{l-1}[i] = \perp$ であるので, 条件 (ii) より $\mathcal{R}_{M_{\mathtt{N}}\cdot T}(\delta(\zeta)) \cap \Gamma[i] = \emptyset$. よって $\xi_{k+1} \notin \Gamma[i]$.

  – $\tau'_{l-1}[i] \neq \perp$ のとき

    $p[i] = \mathtt{S}$, $p'[i] = \mathtt{N}$ なので $\tau_l[i] \ge \tau'_l[i]$. さらに l の決め方と $q[i] = \mathtt{N}$ であることから $\tau_{k+1}[i] \ge \tau'_{k+1}[i]$. よって $\xi'_{k+1} \notin \Gamma[i]$ を示せば十分である.

    l の決め方と $q[i] = \mathtt{N}$ であることから $\xi'_{k+1} \in \mathcal{R}_{M[t'/t]_{\mathtt{N}}\cdot T}(\delta(\eta'_l))$. また $\eta'_l \in \Theta'_{t'}$ であり, $\tau'_{l-1} \in \Phi_{M[t'/t]\cdot T}(u)$, $\tau'_{l-1}[i] \neq \perp$ であるので, 条件 (i) より $\xi'_{k+1} \notin \Gamma[i]$.

  $\eta_{k+1} \in \Theta_e$, $\eta'_{k+1} \in \Theta'_e$ の場合, $\xi_{k+1} <_i> \xi'_{k+1}$. $\xi_{k+1}, \xi'_{k+1} \notin \Gamma[i]$ は $\eta_{k+1} \in \Theta_z$, $q[i] = \mathtt{N}$ の場合と同様の議論で示せる.

  $\eta_{k+1} \in \Theta_t$, $\eta'_{k+1} \in \Theta'_{t'}$ の場合, $\xi_{k+1} <_i> \xi'_{k+1}$. $\xi_{k+1}, \xi'_{k+1} \notin \Gamma[i]$ は $\eta_{k+1} \in \Theta_z$, $q[i] = \mathtt{N}$ の場合で $l = k+1$ とすれば同様に示せる.

($\Leftarrow$) 表 1 の条件を満たさない場合を考える. $\tau[i] \neq \perp$ の場合, $\delta(r) \notin \Gamma[i] \wedge \delta(r') \in \Gamma[i]$ を満たす状況遷移系列 r, r' が存在する. ただし $r \overset{t,t'}{=} r'$. よって (2) は成立しない. 一方, $\tau[i] = \perp$ の場合, $\delta(r) \in \Gamma[i] \wedge \delta(r') \notin \Gamma[i]$ なる r, r' が存在する.

他の $p[i], p'[i]$ の組合せについても同様に議論できる. □

表 1 の条件は, M から $M_{\mathtt{N}}$ および $M[t'/t]_{\mathtt{N}}$ を生成し到達可能状況を調べることで判定できる. これらの機械は M と比較して状態遷移数が少ない.

補題 1 と補題 3 から, $M\cdot T$ と $M[t'/t]\cdot T$ が

表1の条件を満たすならば，$M \cdot T$ と $M[t'/t] \cdot T$ はIO等価である．一方，表1の条件を満たさないならば，$M \cdot T$ と $M[t'/t] \cdot T$ のどちらか一方のみが状況 $\rho(\eta)(\eta \in (\Theta_t \cup \Theta'_{t'}))$ から $\Gamma[i]$ に含まれる状況へ到達する状況遷移系列 r が得られる．r の前に $M \cdot T$ または $M[t'/t] \cdot T$ を初期状況から状況 $\rho(\eta)$ に到達させる状況遷移系列を連接して，状況遷移系列 r_0 を生成する．r_0 は，$M \cdot T$ または $M[t'/t] \cdot T$ のどちらか一方のみを初期状況からタイマ i がタイムアウトする状況へ到達させる状況遷移系列である．

補題4 $M \cdot T$，$M[t'/t] \cdot T$ に対して $r \in TS_{M \cdot T}(\xi_0)$，$r' \in TS_{M[t'/t] \cdot T}(\xi_0)$ かつ $r \overset{t,t'}{=} r'$ なる状況遷移系列 r，r' が存在し，状況 $\delta(r) = \langle s, \bar{\tau} \rangle$，$\delta(r') = \langle s, \bar{\tau}' \rangle$ が次のいずれかを満たすとき $M \cdot T \not\equiv M[t'/t] \cdot T$ である．

- $\langle s, \bar{\tau} \rangle \in \Gamma[i], \langle s, \bar{\tau}' \rangle \notin \Gamma[i], \tau[j] > 0 (i < j \leq n)$
- $\langle s, \bar{\tau} \rangle \notin \Gamma[i], \langle s, \bar{\tau}' \rangle \in \Gamma[i], \tau[j] > 0 (i < j \leq n)$

略証 補題4の条件が満たされているとする．このとき，$\delta(r)$，$\delta(r')$ のどちらか一方のみが $\Gamma[i]$ の要素である．またタイマ i 以外のタイマのタイマ値は0ではない．その結果，$M \cdot T$，$M[t'/t] \cdot T$ のどちらか一方のみが外部出力を行う． □

補題4の条件はタイマ値ベクトルの各要素を比較することで判定できる．

これら補題に基づき，以下のようにして試験系列を生成する．

step i 誤りの種類に応じて，表1の条件判定をする．条件を満たせば試験系列を生成しない．条件を満たさなければ，先に述べた状況遷移系列 r_0 が得られる．

r_0 とそれに対応する r'_0 が補題4の条件を満たせば試験系列とする．満たさなければstep iiへ．

step ii 3.3節で述べた方法で試験系列を生成する．

4.2 終状態の誤りに対する試験系列

プロトコル機械は最簡形のFSMであるので，既存のFSMに対する試験系列生成手法（Wp法[10]，UIOv法[11]）を用いて各状態を区別する状態遷移系列を生成できる．本論文ではWp法を利用して試験系列を生成する．しかし，試験系列がタイムアウト遷移を含む場合，系列が実行可能であるとは限らないので，系列の実行可能性を判定することが必要になる．

仕様 M に対して状態遷移 $t = (u, v, x, y, \bar{p}) \in H$ の終状態誤りを検出する試験系列を生成することを考える．試験系列の生成を4ステップに分け，最初は外部入力遷移のみを考慮して系列を生成する．

step 1 M を解析して，状態遷移 t の終状態 v と状態 $w_1 \in (Q \backslash \{v\})$ を区別する外部入力遷移のみからなる状態遷移系列 $r = t_1 \cdots t_m \in H^*$ を求める．r が得られた場合は，状況遷移系列 $Tr(u) \cdot \eta_t \cdot R$ を試験系列とする．ただし $\eta_t \in \Theta_t$，$R \in \Theta_{t_1} \cdots \Theta_{t_m}$．$Tr(u)$ は $M \cdot T$ を初期状況から $\rho(\eta_t)$ へ到達させる状況遷移系列である．

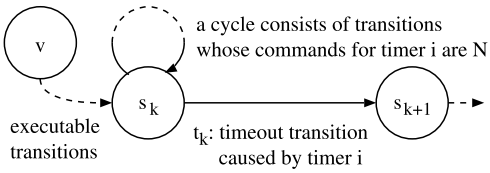


図5 系列の延長
Fig. 5 Sequence extention.

step 2 $w_2 \in (Q \backslash \{v\})$ を外部入力遷移のみで構成される状態遷移系列では区別できない状態とする．M を解析して，状態 v と状態 w_2 を区別する状態遷移系列 $t_1 \cdots t_m \in H^*$ を求める．さらに，$M \cdot T$ で実行可能な状況遷移系列 $R \in \Theta_e^* \Theta_{t_1} \cdots \Theta_e^* \Theta_{t_m}$ が存在するかを確かめる．Θ_e は時間経過遷移の集合を表す．R が存在すれば，状況遷移系列 $Tr(u) \cdot \eta_t \cdot R$ を試験系列とする．ただし $\eta_t \in \Theta_t$，$Tr(u)$ は $M \cdot T$ を初期状況から $\rho(\eta_t)$ へ到達させる状況遷移系列である．

step 3 $\Theta_e^* \Theta_{t_1} \cdots \Theta_e^* \Theta_{t_m}$ に含まれるどの状況遷移系列も実行不可能であるとは，系列中に実行できないタイムアウト遷移 t_k が存在するということである．そこで，遷移 $t_k = (s_k, s_{k+1}, \mathrm{Timeout}[i], y, \bar{p})$ の始状況でタイマ i が動作しているとき，その前に閉路 $c_1 \cdots c_n$ を加えて実行可能にすることを考える（**図5**）．閉路はタイマ i に対する操作が N の状態遷移のみで構成されるものを考える．その後，$M \cdot T$ で実行可能な状況遷移系列 $\Theta_e^* \Theta_{t_1} \cdots \Theta_e^* \Theta_{t_{k-1}} \{\Theta_e^* \Theta_{c_1} \cdots \Theta_e^* \Theta_{c_n}\}^+ \Theta_e^* \Theta_{t_k} \cdots \Theta_e^* \Theta_{t_m}$ があるかを調べる．

step 4 step 3で試験系列を生成できなかった状態 $w_3 \in (Q \backslash \{v\})$ について，3.3節で述べた方法によって試験系列を生成する．

step 1では仕様 M のみの解析，step 2と3ではタイマを考慮した仕様 $M \cdot T$ の解析，step 4では誤実装ごとにタイマを考慮したシステムの解析を行っている．

5. 試験系列生成システムを用いた実験

提案手法に基づいて試験系列生成システムを開発し，DHCP（Dynamic Host Configuration Protocol）[12]に対して適用した．

5.1 連立不等式を用いたタイマ値の表現

タイマを有限状態機械で表現すると状態数が膨大になり，仕様を解析するときの計算時間や計算領域が問題になる．プロトコル機械の各状態でタイマがとりうる値を連立不等式の集合で表すことで，複数の状況および遷移を一括して扱うことができ，仕様解析の手間を抑えることができる[13]．

タイマがとりうる値を表現するために，各タイマのタイマ値の上限と下限を表す不等式，および各タイマのタイマ値の差の上限と下限を表す不等式を用いる．各不等式は，$x - y \leq c$（x，y：タイマ値を表す変数，c：定数）の形で表される差分不等式に限定できる．差分不等式からなる連立不等式は，$O(lm)$（l：不等式の数，m：変数の数）で解を求めることができる[14]．時間オートマトンの検証手続きでも同様の方法を用いることが検討されている[9]．

5.2 例プロトコル

DHCP は TCP/IP ネットワークに接続されているホストにコンフィグレーション情報を伝達するための仕組みである．DHCP はクライアント・サーバ方式で，サーバがクライアントに対して IP アドレスを動的に割り当てる．クライアントはアドレス割当てを要求するときに貸出期間を指定できる．DHCP では時間は秒単位で扱われる．最小の IP アドレス貸出期間は 1 時間である．また，クライアントはタイムアウトに基づいてメッセージを再送する．最初の再送間隔は 4 秒で，再送ごとに間隔が 2 倍になる．

クライアントが IP アドレスの貸出期間を指定できるとすると，本論文のモデルでは記述できない．そこで IP アドレスの貸出期間を 1 時間に限定した．DHCP を本論文のモデルで記述すると，状態数 11，状態遷移数 74，タイマ数 9 となった．9 個のタイマのうち，5 個はメッセージ再送間隔の監視，また 4 個は IP アドレス貸出期間の監視に使用している．

5.3 実験結果

DHCP に対してシステムを適用し，仕様と実装との IO 等価性の判定および IO 等価でない実装に対する試験系列の生成を行った．実行した計算機の CPU は PentiumIII 600 MHz，搭載メモリは 128 MB である．システムの実行にかかった時間は約 7 分，使用したメモリは約 2 MB であった．

表 2 に，試験項目数，等価と判定した項目数および得られた試験系列数（等価でない項目数）をまとめた．また，表中の step i，ii および step 1，2，3，4 は，4 章で述べた試験系列生成アルゴリズムの各段階に対応しており，どの段階で等価性の判定および試験系列の生成を行ったかを表している．

表 2 DHCP への適用結果
Table 2 Result applying our method to DHCP.

	タイマ操作
試験項目	1332
等価	457
試験系列	875
step i	1332
step ii	0

	終状態
試験項目	740
等価	0
試験系列	740
step 1	705
step 2，3	35
step 4	0

タイマ操作誤りに対しては，step i で等価性の判定および等価でない誤りに対して試験系列の生成を行うことができた．試験項目のうち等価と判定された誤りは約 34%であった．これらのほとんどは，試験対象である状態遷移の始状態で動作していない（$\perp$ である）タイマに対する操作 `N` を操作 `D` と誤った場合であった．

終状態誤りに対する試験系列は，約 95%が step 1 で生成された．また，step 4 で試験系列を生成した誤りはなかった．

個々の誤りを含む機械を作成して，到達可能状態を解析することによって試験系列を生成した誤りはなかった．この結果，DHCP に対する試験系列を効率良く求めることができたと考えられる．

比較のために，3.3 節で述べた方法で DHCP に対して試験系列を生成した．その結果，実行時間は約 1 時間 40 分，使用メモリは約 2 MB であった．提案手法では，ほとんどの場合について，仕様の部分機械を解析することで試験系列を生成することができたので，実行時間が短くなったと考えられる．また，使用メモリに関してはあまり差がなかった．これは，提案手法で解析する仕様の部分機械が，元の仕様とあまり変わらない場合があるためと考えられる．

6. まとめ

本論文では，OS のタイマ機能を利用する FSM プロトコルに対して，タイマ操作の誤り，終状態の誤りを検出するための適合性試験系列生成手法を提案した．本論文で提案した手法に基づいて試験系列生成システムを開発し，例プロトコルに対して適用して，試験系列が効率的に生成できることを確認した．

参考文献

1) Bosik, B.S. and Uyar, M.U.: Finite State Machine Based Formal Methods in Protocol Conformance Testing: from Theory to Implementation, *Computer Networks and ISDN Systems*, Vol.22, No.1, pp.7–33 (1991).

2) 森 亮憲，樋口昌宏：タイマシステムコールを用いるFSMプロトコルの適合性試験について，情報処理学会研究報告，DPS-94-22 (1999).
3) 森 亮憲，徳田康平，樋口昌宏：タイマシステムコールを用いるFSMプロトコルに対する適合性試験系列生成手法，情報処理学会DICOMOシンポジウム，pp.655–660 (2000).
4) ISO9646: Information Technology, Open System Interconnection, Conformance Testing Methodology and Framework, ISO/IEC 9646 (1991).
5) Lima, Jr., L.P. and Cavalli, A.R.: A Pragmatic Approach to Generating Test Sequences for Embedded Systems, *Proc. IFIP 10th International Workshop on Testing of Communicating Systems* (*IWTCS'97*), pp.288–307 (1997).
6) Petrenko, A., Yevtushenko, N. and Bochmann, G.V.: Fault Models for Testing in Context, *Proc. Joint International Conference on 9th Formal Description Techniques and 16th Protocol Specification, Testing, and Verification* (*FORTE/PSTV'96*), pp.163–178 (1996).
7) Petrenko, A. and Yevtushenko, N.: Fault Detection in Embedded Components, *Proc. IFIP 10th International Workshop on Testing of Communicating Systems* (*IWTCS'97*), pp.272–287 (1997).
8) Alur, R. and Dill, D.L.: A Theory of Timed Automata, *Theoretical Computer Science*, Vol.126, No.2, pp.183–235 (1994).
9) Alur, R.: Timed Automata, *Proc. 11th International Conference on Computer-Aided Verification* (*CAV'99*), LNCS 1633, pp.8–22 (1999).
10) Fujiwara, S., Bochmann, G.V., Khendek, F., Amalou, M. and Ghedamsi, A.: Test Selection Based on Finite State Models, *IEEE Trans. Softw. Eng.*, Vol.17, No.6, pp.591–603 (1991).
11) Chan, W.Y.L., Vuong, S.T. and Ito, M.R.: An Improved Protocol Test Generation Procedure Based on UIOs, *Proc. ACM SIGCOMM'89*, pp.283–294 (1989).
12) Droms, R.: Dynamic Host Configuration Protocol, RFC 2131, Bucknell University (1997).
13) 徳田康平，森 亮憲，樋口昌宏，谷口健一：タイマを用いる有限状態機械でモデル化されたシステムの検証手続き，情報処理学会研究報告，DPS-101-15 (2001).
14) Cormen, T.H., Leiserson, C.E. and Rivest, R.L.: *Introduction to Algorithms*, chapter 7, pp.539–543, The MIT Press (1990).

(平成13年 6 月 8 日受付)
(平成13年10月16日採録)

森 亮憲（学生会員）

平成10年大阪大学基礎工学部情報工学科卒業．平成12年同大学大学院博士前期課程修了．現在，同大学大学院博士後期課程在学中．通信プロトコルの適合性試験，検証法等の研究に従事．

徳田 康平（正会員）

平成11年大阪大学基礎工学部情報工学科卒業．平成13年同大学大学院博士前期課程修了．現在，シャープ株式会社勤務．在学中，通信プロトコルの適合性試験，検証法等の研究に従事．

多田 知正（正会員）

平成5年大阪大学基礎工学部情報工学科卒業．平成7年同大学大学院博士前期課程修了．平成10年同大学大学院博士後期課程退学．現在，同大学大学院基礎工学研究科助手．博士（工学）．分散システム，分散プログラミング環境に関する研究に従事．

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．(株）富士通研究所勤務，大阪大学助手，講師を経て，現在近畿大学理工学部電気工学科助教授．博士（工学）．通信プロトコル等の並行処理系の検証，試験に関する研究に従事．

東野 輝夫（正会員）

昭和54年大阪大学基礎工学部情報工学科卒業．昭和59年同大学大学院博士課程修了．同年同大学助手．平成2年，平成6年モントリオール大学客員研究員．現在，大阪大学大学院基礎工学研究科教授．工学博士．分散システム，通信プロトコル等の研究に従事．電子情報通信学会，ACM各会員．IEEE Senior Member.