# データ付時間オートマトンの双模倣等価性の記号的検証法

中田明夫[†] 服部哲[††]
東野輝夫[†] 谷口健一[†]

本論文では，時間制約と入出力データに関する条件判定が同時に記述できるオートマトンモデル，データ付時間オートマトンの双模倣等価性の記号的検証法を提案する．データ付時間オートマトンの各状態はいくつかの変数を持ち，その値が遷移条件を満足するような遷移を実行する．遷移は入出力遷移と時間遷移の 2 種類があり，入力遷移は入力値を変数に代入し，出力遷移は与えられた式の値を出力し，時間遷移は経過時間を変数に代入したのち次の状態へ遷移する．各状態は時間遷移しかできない休止状態と入出力遷移しかできない活動状態に分類され，休止状態からは活動状態へ，活動状態からは休止状態へ遷移する．提案する手法は，Hennessy ら（1995）が提案した，入出力遷移のみを記述できる同種のモデルに対する双模倣等価性の記号的検証法の拡張であり，データ付時間オートマトンの任意の状態対に対して，それらが双模倣等価となるための変数に関する条件を自動導出する．

# Symbolic Verification of Bisimulation Equivalence for Timed Automata with Data Values

Akio Nakata,[†] Satoshi Hattori,[††] Teruo Higashino[†] and Kenichi Taniguchi[†]

In this paper, we propose a timed I/O automaton model and its verification method of bisimulation equivalence. In a timed I/O automaton model, a set of variables is assigned to each state, and each transition can be executed if its transition condition is satisfied by the current values of the variables. There are two kinds of transition, one is an I/O transition and the other is a time transition. An input transition can substitute the input value into a variable. A time transition may have a variable, to which the amount of the delay from the execution time of the previous I/O action is assigned. All states are divided into either idle states or active states. In an idle state, only a time transition can be executed, whereas in an active state, some I/O transitions can be executed. The proposed verification method is an extension of the method (Hennessy, et al., 1995). It derives the weakest condition for the variables to make given two states of timed I/O automata bisimilar.

## 1. まえがき

近年，通信プロトコルなどの形式的仕様記述とその検証方法に関するさまざまな研究が行われている[1),2),4)]．一般に通信プロトコルはオートマトンなどのモデルで記述されることが多い．また，それらのモデルでは，入出力データやある動作の生起時刻に依存する動作がよく現れる．このため，取り扱う入出力データの範囲や各動作の実行可能時刻などのパラメータの値を変更した場合にもシステムが同じ動作を行えるかどうかを検証できることが望ましい．しかし，データや時間の値は一般に無限にあり，原理的にはそれに応じてオートマトンの遷移も無限に生成されてしまうため，これらの検証は容易ではない．

この問題を解決するためにいくつかの提案がなされてきた．その中で文献 3) で提案された入出力データを扱えるモデル上での双模倣等価性の記号的検証法は比較的自由に条件記述が行え，検証コストもデータ値に依存しないものであった．一方，我々はこれまでに時間値を扱えるような新しいモデル（A-TSLTS）を作り，そのモデル上での双模倣等価性の記号的検証法を提案している[5)]．

本論文では，この 2 つの方法を組み合わせて，データと時間に関する制約をともに記述できるような 1 つ

† 大阪大学大学院基礎工学研究科情報数理系専攻
Department of Informatics and Mathematical Science, Graduate School of Engineering Science, Osaka University
†† 北陸先端科学技術大学院大学情報科学研究科
School of Information Science, Japan Advanced Institute of Science and Technology

のプロセスモデル（データ付時間オートマトンと呼ぶ）を提案し，その上での双模倣等価性の記号的検証法を提案する．

データ付時間オートマトンの各状態はパラメータ値などを保持するための変数をいくつか持ち，それらの値が遷移条件を満足するような遷移を実行する．初期状態の変数の値は設計者が前もって決定するものとし，途中状態の変数の値は，初期値あるいはそれ以前の状態遷移で入力/代入された値が用いられる．遷移は入出力遷移と時間遷移の 2 種類がある．入力遷移は入力値を変数に代入し，出力遷移は与えられた式の値を出力する．また，時間遷移は直前の入出力遷移の実行から次の入出力遷移の実行までの経過時間を変数に代入する．いずれの遷移においても，直前の状態までに代入された変数値（過去の入力値や経過時間）を次の状態の遷移条件の判定や出力値として用いることができる．各状態は時間遷移しかできない休止状態と入出力遷移しかできない活動状態の 2 つに分類され，休止状態からは活動状態へ，活動状態からは休止状態へと交互に遷移する．

提案する手法は，文献 3) で提案された，入出力遷移のみを記述できる同種のモデルに対する双模倣等価性の記号的検証法の拡張であり，データ付時間オートマトンの任意の状態対に対して，それらが本論文の意味において双模倣等価となるために変数間で成立すべき条件を自動導出する．

文献 3) では，我々のデータ付時間オートマトンの時間遷移を省いたモデルと同等なモデルである記号的遷移グラフ（symbolic transition graph）で記述されたシステムの任意の状態対に対して，それらが双模倣等価となるために必要十分なパラメータ値の間に成り立つべき条件式を自動導出するアルゴリズムが提案されている．本論文では，データ付時間オートマトンの時間遷移を特殊な入力動作に対応付けることによりデータ付時間オートマトンの任意の状態対が時間双模倣となるために変数間で成立すべき条件式を求める問題が文献 3) におけるパラメータ間の関係式を求める問題に帰着されることを示す．

本論文は次のように構成される．2 章ではシステムの仕様記述モデル，データ付時間オートマトンを定義する．3 章ではデータ付時間オートマトンの双模倣等価性判定問題が文献 3) におけるパラメータ間の関係式を求める問題へ帰着されることを理論的に証明する．4 章では結論と今後の課題を述べる．

## 2. データ付時間オートマトン

ここでは，仕様記述モデルとして**データ付時間オートマトン**を提案する．これは文献 5) で使用されていたモデル A-TSLTS を拡張したものである．ここで用いるデータ付時間オートマトンはある種の時間オートマトンモデルで，それぞれの状態 $s$ には $s$ において利用可能な変数の集合（$DVar(s)$）が割り当てられている．ここで $DVar(s)$ に含まれている各変数の値は，初期状態からいかなる遷移を行ってもこの状態 $s$ に到着するまでに定まっていなければならない．遷移としては $s \xrightarrow{-\gamma[P]} s'$ で表される**入出力動作遷移**と $s \xrightarrow{-e(d)[P]} s'$ で表される**時間遷移**がある．ここで $\gamma$ はある入出力動作，$d$ は遅延の長さを表す．さらに，状態を**休止状態**と**活動状態**に分け，休止状態からの遷移としては時間遷移のみを許し，遷移先は活動状態とする．活動状態からの遷移としては，入出力動作遷移のみを許し，遷移先は休止状態とする．また，初期状態は休止状態とする．このように，このモデルでは休止状態と活動状態が交互に現れる．この性質を**交替性（alternation）**と呼ぶ．$s \xrightarrow{-\gamma[P]} s'$ は $DVar(s)$ の変数の現在の値が $P$ を満たすとき，動作 $\gamma$ が実行可能であることを表す．動作の種類としては**入力動作** $c?x$，**出力動作** $c!E$ の 2 種類があり，これらの動作の実行には時間がかからないものとする．$c$ は**動作名**であり，指定された動作名集合 $Act$ の要素であるとする．また，$x$ は入力値を表す変数であり，$E$ は出力値を表す式である．$P$ は**遷移条件**を表し，$DVar(s)$ の中の任意の変数を用いて書くことができ，$\gamma$ が入力動作 $(c?x)$ であった場合，入力値を表す変数 $x$ も用いることができる（この場合，$x \notin DVar(s)$ でなければならない）．活動状態からは複数個の動作遷移の存在を許し，条件が重なった場合，それらの遷移の中から非決定的に選択される．時間遷移 $s \xrightarrow{-e(d)[P]} s'$ において，$d$ は $DVar(s)$ の変数の現在の値のもとで $P$ を満たすような値でなければならず，$P$ を満たすような $d$ の最大の値まで遅延が許される．$d$ がこの最大の値を超すような遅延は許されない．また，$d \notin DVar(s)$ でなければならない．$d$ にはこの時間遷移が実行される時点までの遅延時間が値付けされる．また，遷移条件 $P$ は変数 $d$ と $DVar(s)$ の中の任意の変数を用いて書くことができる．なお，データ付時間オートマトンのすべての状態は時間遷移をたかだか 1 つしか持たないものとする．さらに，任意の状態は入射する時間遷移もたかだか 1 つしか持たないとする．また，変数 $d$ や $x$ は遷移後の状態 $s'$ のパラメータ集合 $DVar(s')$

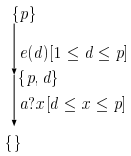


**図 1** データ付時間オートマトンの遷移
Fig. 1 Transitions in timed I/O automata.

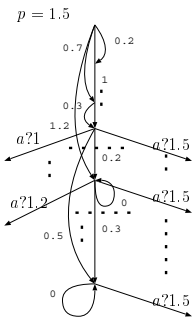


**図 2** 図 1 の意味モデル（$p = 1.5$ の場合）
Fig. 2 Semantic model for Fig. 1 (the case of $p = 1.5$).

に含まれてもよい，すなわち，その後の任意の遷移において $d$ や $x$ を遷移条件として用いてよいとする．

なお，データ付時間オートマトンでは時間遷移 $-e(d)\rightarrow$ と（時間のかからない）入出力遷移を分けてモデル化しているが，Alur らが提案した時間オートマトン[1] のように入出力遷移と同時に時間も進行するモデル化も存在する．両者のモデルの表しているものは基本的に同じであり☆，記述方法が異なるだけである．

**図 1** にデータ付時間オートマトンの遷移の例を示す．これは初期パラメータ値 $p$ が与えられたとき，$1 \le d \le p$ を満たす遅延 $d$ の後に $d \le x \le p$ を満たすような入力値 $x$ を受け付けるという動作をするモデルであり，意味的には**図 2** のようなモデルを表す．図 2 において，たとえば入力動作 $a$ による値 1 の入力を遷移 $-a?1\rightarrow$ で，$n$ 単位時間の時間（遅延）遷移を $-n\rightarrow$ で表す．一般に意味モデルにおいて入出力遷移は入出力値の範囲によって横に無限に分岐する．また，時間遷移に関しては時間の連続性により 1 つの時間遷移が $-5\rightarrow=-2\rightarrow-3\rightarrow=-1.5\rightarrow-0.5\rightarrow-2\rightarrow-1\rightarrow=\ldots$ というように，連続する複数の時間遷移の列にいくらでも分割可能である．

☆ 正確には，時間遷移を分けることによって，より細かい記述が可能となる．たとえば，動作 $a?x$ を遅延 $t_1$ から $t_2$ の間で実行するという条件と，遅延を $t_2$ 以下しか許さないという条件をあわせると（$s\ -e(d)[d \le t_2]\rightarrow\ s_1\ -a?x[t_1 \le d \le t_2]\rightarrow\ s_2$），遅延 $t_2$ の次はさらに遅延することができない，すなわち，$a?x$ を $t_1$ から $t_2$ の間に**必ず**実行するという記述になる．一方，遅延遷移の条件を任意の遅延を許すという条件に変更すると（$s\ -e(d)[true]\rightarrow\ s_1\ -a?x[t_1 \le d \le t_2]\rightarrow\ s_2$），$a?x$ を $t_1$ から $t_2$ の間に実行するか，実行できなければ無限に時間が経過するという意味の記述になる．

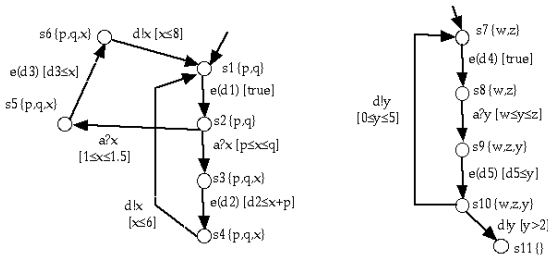


**図 3** データ付時間オートマトンの例
Fig. 3 Example of timed I/O automata.

**例 2.1** **図 3** に 2 つのデータ付時間オートマトンの例を示す．$s_i$ にそれぞれ割り当てられている集合 $\{\cdots\}$ は $DVar(s_i)$ を示し，それぞれの遷移に割り当てられている $\gamma[P]$ は動作名 $\gamma$ と遷移条件 $P$ を表している．また，2 つのオートマトンの初期状態はそれぞれ $s_1$，$s_7$ である．このモデルの動作を，左のオートマトンを例にとって説明する．動作開始時に変数（パラメータ）$p$，$q$ の値が与えられるものとする．まず初期状態 $s_1$ において任意の遅延時間（$d_1$ とする）経過後，入力動作 $a?x[p \le x \le q]$ または $a?x[1 \le x \le 1.5]$ を非決定的に選択し，実行できる．前者は $p$ から $q$ までの値 $x$ が入力可能で，後者は 1 から 1.5 までの値 $x$ が入力可能である．入力動作実行時の遅延時間の値は変数 $d_1$ に代入される．$a?x[p \le x \le q]$ が実行されたとすると，次に状態 $s_3$ に遷移する．$s_3$ は変数に $p$，$q$ に加え，$a?x$ での入力値を保持する変数 $x$ を利用できる．これは，$s_3$ 以降の動作が入力値 $x$ に依存してもよいことを意味する．$s_3$ では $x + p$ 以内の遅延が可能であり，その時間以内に次の状態 $s_4$ の出力動作 $d!x[x \le 6]$ を実行しなければならない．もし $x$ が 6 以下ならば出力動作 $d!x$ を $x + p$ 以内の遅延の後に実行可能で，そのときの遅延時間は変数 $d_2$ に代入される．$d!x$ を実行後には初期状態 $s_1$ に遷移し，最初と同様の動作を繰り返す．$s_5$ に遷移した場合，および，右のオートマトンの場合も同様である． □

## 3. 時間双模倣等価性とその検証法

本章ではデータ付時間オートマトンに対して，時間とデータをともに考慮した双模倣等価性を定義する．そして，データ付時間オートマトンの双模倣等価性検

証法が文献 3) の early bisimulation の検証法に帰着されることを示す．

**定義 3.1** 各変数への値の代入を $\rho$，$\rho'$ などで表記する．論理式 $P$ へ代入 $\rho$ を施した式が真であることを $\rho \models P$ と表記する．$\rho[x = e]$ を変数 $x$ に $e$ を代入する以外は $\rho$ と同じ代入であると定義する．論理式 $P$ における変数 $x$ のすべての自由な出現を式 $e$ に置き換えた論理式を $P\{e/x\}$ と表記する．データ付時間オートマトンの状態 $s$ と代入 $\rho$ の対 $(s, \rho)$ を $\rho(s)$ と表記し，代入 $\rho$ による $s$ の**具体的状態**と呼ぶ．具体的状態は直観的には変数に具体的な値を代入した状態のことである． □

データ付時間オートマトンの具体的な動きは具体的状態の遷移関係によって記述される．その遷移関係は直観的には 2 章で述べたとおりであるが，形式的には以下のように定義される．

**定義 3.2**

- 非負実数 $t$ に対してもし $\rho[d = t] \models P$ ならば，$s \ -e(d)[P]\rightarrow\ s'$ に対して $\rho(s) \ -t\rightarrow\ \rho[d = t](s')$，かつ $0 \leq t_1 \leq \cdots \leq t_k \leq t$ なる任意個の非負実数 $t_1, \ldots, t_k$ に対して，$\rho(s) \ -t_i\rightarrow\ \rho[d = t_i](s')$ $(1 \leq i \leq k)$ かつ $\rho[d = t_i](s') \ -(t_j - t_i)\rightarrow\ \rho[d = t_j](s')$ $(1 \leq i < j \leq k)$.
- データ値 $v$ に対してもし $\rho[x = v] \models P$ ならば，$s \ -a?x[P]\rightarrow\ s'$ に対して $\rho(s) \ -a?v\rightarrow\ \rho[x = v](s')$
- $\rho \models P$ かつ，$\rho$ による値を式 $E$ に代入して計算した値を $\rho(E)$ とするとき，$s \ -a!E[P]\rightarrow\ s'$ に対して $\rho(s) \ -a!\rho(E)\rightarrow\ \rho(s')$

以上の遷移関係をデータ付時間オートマトンの**具体的遷移関係**と呼び，この遷移関係によって定義される遷移システムを**具体的遷移システム**と呼ぶ． □

**命題 3.1** 任意のデータ付時間オートマトンの具体的遷移システムは以下の性質を満足する．

**時間決定性** $\rho(s) \ -t\rightarrow\ \rho'(s')$ かつ $\rho(s) \ -t\rightarrow\ \rho''(s'')$ ならば $\rho' = \rho''$ かつ $s' = s''$.

**時間連続性** $\rho(s) \ -t\rightarrow\ \rho'(s')$ ならば任意の $0 \leq t' < t$ に対してある $\rho''$，$s''$ が存在して $\rho(s) \ -t'\rightarrow\ \rho''(s'')$ かつ $\rho''(s'') \ -(t - t')\rightarrow\ \rho'(s')$.

**（証明）** データ付時間オートマトンの定義および定義 3.2 より容易に示される．詳細は省略する． □

我々が検証したい等価性は**時間双模倣等価性**（時間双模倣性ともいう）である．時間双模倣等価性はデータ付時間オートマトン $M$ の具体的遷移システムの状態対に対して以下のように定義される．

**定義 3.3** データ付時間オートマトン $M$ の具体的状態の対が以下の条件を満たす関係 $R_t$ に属するとき，それらは**時間的双模倣等価**であるという．

- $(\rho_1(s_1), \rho_2(s_2)) \in R_t$ ならば，任意の $a \in Act$，$v \in Val$，$\$ \in \{?, !\}$，非負実数 $t$ に対して
  - $\rho_1(s_1) \ -a\$v\rightarrow\ \rho_1'(s_1')$ ならば，ある $\rho_2'(s_2')$ が存在して $\rho_2(s_2) \ -a\$v\rightarrow\ \rho_2'(s_2')$ かつ $(\rho_1'(s_1'), \rho_2'(s_2')) \in R_t$,
  - $\rho_2(s_2) \ -a\$v\rightarrow\ \rho_2'(s_2')$ ならば，ある $\rho_1'(s_1')$ が存在して $\rho_1(s_1) \ -a\$v\rightarrow\ \rho_1'(s_1')$ かつ $(\rho_1'(s_1'), \rho_2'(s_2')) \in R_t$
  - $\rho_1(s_1) \ -t\rightarrow\ \rho_1'(s_1')$ ならば，ある $\rho_2'(s_2')$ が存在して $\rho_2(s_2) \ -t\rightarrow\ \rho_2'(s_2')$ かつ $(\rho_1'(s_1'), \rho_2'(s_2')) \in R_t$,
  - $\rho_2(s_2) \ -t\rightarrow\ \rho_2'(s_2')$ ならば，ある $\rho_1'(s_1')$ が存在して $\rho_1(s_1) \ -t\rightarrow\ \rho_1'(s_1')$ かつ $(\rho_1'(s_1'), \rho_2'(s_2')) \in R_t$

□

**例 3.1** 図 3 の 2 つのデータ付時間オートマトンは

$$P \stackrel{\text{def}}{=} [[(p = w = 0) \wedge (1.5 \leq q = z \leq 2)] \\ \vee [q < p \wedge w = 1 \wedge z = 1.5]]$$

を満足する任意の代入 $\rho$ に対して時間双模倣等価である．すなわち，$\rho \models P$ ならば $(\rho(s_1), \rho(s_7)) \in R$ なる時間双模倣 $R$ が存在する．さらに $P$ を満たさない代入 $\rho$ に対しては $s_1$ と $s_7$ が時間双模倣等価とはならない．すなわち，$P$ は $s_1$ と $s_7$ が時間双模倣等価となるための最も弱い条件である． □

文献 3) では入出力データのみを考慮したモデル，記号的遷移グラフ（symbolic transition graph）を提案し，その上の等価性として early bisimulation および late bisimulation を定義している．2 つの等価性の違いは入力動作 $-a?x\rightarrow$ の動きの解釈にあるが，ここでは深く立ち入らない．ここでは，記号的遷移グラフおよび early bisimulation の定義を文献 3) から本論文の表記に合わせた形で再掲する．

まず，記号的遷移グラフの定義を述べる．

**定義 3.4 記号的遷移グラフ**はデータ付時間オートマトンの時間遷移を省いたものである．すなわち，状態遷移関係として，$s \ -a?x[P]\rightarrow\ s'$ および $s \ -a!x[P]\rightarrow\ s'$ の形のもののみを持ち，休止状態，活動状態の区別はない． □

early bisimulation の定義は以下のとおりである．

**定義 3.5** 具体的状態の対が以下の条件を満たす関係 $R_e$ に属するとき，それらは **early bisimulation** であるという．

- $(\rho_1(s_1), \rho_2(s_2)) \in R_e$ ならば，任意の $a \in Act$，$v \in Val$，$\$ \in \{?, !\}$ に対して
  - $\rho_1(s_1) \ -a\$v\rightarrow\ \rho_1'(s_1')$ ならば，ある $\rho_2'(s_2')$ が存在して $\rho_2(s_2) \ -a\$v\rightarrow\ \rho_2'(s_2')$ かつ $(\rho_1'(s_1'), \rho_2'(s_2')) \in$

$R_e$,

– $\rho_2(s_2) \xrightarrow{-a\$v\rightarrow} \rho_2'(s_2')$ ならば，ある $\rho_1'(s_1')$ が存在して $\rho_1(s_1) \xrightarrow{-a\$v\rightarrow} \rho_1'(s_1')$ かつ $(\rho_1'(s_1'), \rho_2'(s_2')) \in R_e$ □

文献 3) の結果のうち，本論文で用いるものは以下の定理で述べられる．

**定理 3.1（Hennessy and Lin）** 記号的遷移グラフの任意の状態対 $(s_1, s_2)$ に対して，$DVar(s_1)$, $DVar(s_2)$ に属する変数を自由変数とする次のような論理式 $bisim_E(s_1, s_2)$ を求めるアルゴリズムが存在する．

$\rho \models bisim_E(s_1, s_2) \iff \rho(s_1)$ と $\rho(s_2)$ は early bisimulation □

読者の便宜のため，アルゴリズム $bisim_E()$ の具体的な内容を付録 A.1 に示す．

データ付時間オートマトンの等価性判定問題を定理 3.1 へ帰着して行う．そのための準備として，データ付時間オートマトンから記号的遷移グラフへのある写像を定義する．この写像は基本的に時間遷移 $-e(d)[P]\rightarrow$ の遷移条件 $P$ を満たす非負の遅延時間が存在する場合，その時間遷移 $-e(d)[P]\rightarrow$ をある特殊な入力遷移 $-e?d[P']\rightarrow$ で置き換えるものである．$P'$ は，特に $P$ を満たす $d$ の最大値 $max_d(P)$ が存在するならば $[0 \le d \le max_d(P)]$，最大値は存在しないが上限 $sup_d(P)$ が存在するならば $[0 \le d < sup_d(P)]$，さもなければ $[0 \le d]$ と等価な論理式であり，一般には $P' \stackrel{\text{def}}{=} [0 \le d \wedge \exists d'[d \le d' \wedge P\{d'/d\}]]$ と求められる．たとえば，$-e?d[d = 2 \vee 2.5 \le d \le 3]\rightarrow$ という時間遷移は $-e?d[0 \le d \le 3]\rightarrow$ のような入力遷移に置き換える．これは，時間連続性より，ある遅延時間 $t$ で時間遷移 $-t\rightarrow$ が可能なら，$t$ 以下のすべての遅延時間 $t'$ による時間遷移 $-t'\rightarrow$ が可能になるので☆，対応する入力遷移 $-e?t'\rightarrow$ が可能になるように $-e?d[P']\rightarrow$ を構成している．

**定義 3.6** データ付時間オートマトン $M$ から記号的遷移グラフ $M'$ への写像 $f()$ を以下のように定義する．

- $M'$ の状態，各状態の変数（パラメータ）の集合，入出力遷移は $M$ と同じである（1 対 1 対応する）．
- $M$ の動作名集合 $Act$ に対して，$f(Act) = Act \cup \{e\}$．ただし，$e$ は $Act$ に含まれない新しい動作名とする．
- $M$ の時間遷移 $s \xrightarrow{-e(d)[P]\rightarrow} s'$ に対して，$f(s \xrightarrow{-e(d)[P]\rightarrow} s') = f(s) \xrightarrow{-e?d[P']\rightarrow} f(s')$．ただし，$P' \stackrel{\text{def}}{=} [0 \le d \wedge \exists d'[d \le d' \wedge P\{d'/d\}]]$． □

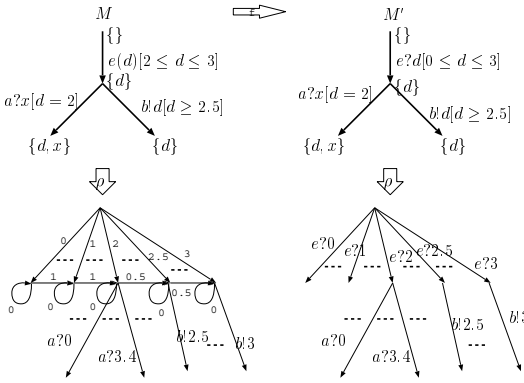


**図 4** $f()$ による $M$ と $M'$ の具体的遷移システムの対応関係
Fig. 4 Correspondence between concrete transition systems for $M$ and $M'$ by the mapping $f()$.

**図 4** に示されるように，時間遷移と入力遷移のセマンティクスの違いは，時間の概念を含む具体的遷移システムにおいては，$\rho[d = t_i](s') \xrightarrow{-(t_j - t_i)\rightarrow} \rho[d = t_j](s')$ のような時間の経過を表す（図で横向きに記した）遷移や 0 遅延に相当する自己ループ遷移が加わっている点である．

データ付時間オートマトン $M$ の任意の具体的状態 $\rho(s)$ から $t$ 単位時間経過後の状態は，もし存在するならば時間決定性より一意に定まる．よってそのような状態を $\rho(s)$ および $t$ の部分関数 $g(\rho(s), t)$ で表すことにする．一方，記号的遷移グラフ $M'$ の任意の具体的状態 $\rho(f(s))$ に対しても，$\rho(f(s)) \xrightarrow{-e?t\rightarrow} \rho'(f(s'))$ なる具体的状態 $\rho'(f(s'))$ は，遷移 $-e?t\rightarrow$ が 1 つの状態からたかだか 1 つしか出ていないことより，$\rho(f(s))$ および $t$ のみによって一意に定まる．よってそのような状態を $g(\rho(f(s)), t)$ で表す．

一般に，写像 $f()$ によって変換された記号的遷移グラフ $M'$ の具体的遷移システム上の任意の early bisimulation $R$ に対して，$R_t \stackrel{\text{def}}{=} \{(\rho_i(s_i), \rho_j(s_j)) | (\rho_i(f(s_i)), \rho_j(f(s_j))) \in R\}$ とおけば，$R_t$ がもとのデータ付時間オートマトン $M$ の具体的遷移システム上で時間双模倣となればよいが，

☆ 時間連続性の仮定によって，たとえば，$s \xrightarrow{-e(d)[t_1 \le d \le t_2]\rightarrow} s' \xrightarrow{-a?x[t_1 \le d \le t_2]\rightarrow} s''$ と $s \xrightarrow{-e(d)[0 \le d \le t_2]\rightarrow} s' \xrightarrow{-a?x[t_1 \le d \le t_2]\rightarrow} s''$ は等価と見なされる．このとき，前者・後者とも状態 $s$ から次の入力動作 $a?x$ がまだ実行できない状態（たとえば $0 \le t' < t_1$ なる $t'$ に対して $s$ から $t'$ 単位時間経過した状態）への遷移が許されることになるが，入力動作 $a?x$ は $s$ から $t_1 \sim t_2$ 単位時間経過した後の状態でしか実行できないことに変わりはない．このとき，たとえば状態 $s$ から $t'$ 単位時間経過した後の状態では，さらに $t' - t_1$ 単位時間以上経過して $a?x$ が実行できる状態に遷移することのみが許される．ただし，$s \xrightarrow{-e(d)[true]\rightarrow} s_1 \xrightarrow{-a?x[t_1 \le d \le t_2]\rightarrow} s_2$（$a?x$ を $t_2$ までに実行できなかったら無限に時間が経過する）などの記述とは等価とならない．

$(g(\rho_i(f(s_i)),t),g(\rho_j(f(s_j)),t')) \in R$ $(t \neq t')$ などの場合☆，$R_t$ は時間双模倣にならない．しかし，そのような場合でも $R$ の中から同じ入出力動作と時間経過に相当する入力遷移の系列を実行した状態対のみを残した関係 $R'$ $(R' \subseteq R)$ を構成すると，$R'$ も $M'$ の具体的遷移システム上で early bisimulation を満たすこと，ならびに，そのような early bisimulation $R'$ から上と同様に構成した $R'_t$ が，もとのデータ付時間オートマトン $M$ の具体的遷移システム上で時間双模倣となること，が証明できる．以下，これらの証明を順に示す．

**補題 3.1** $M'$ の具体的遷移システムの任意の状態対 $(\rho_1(f(s_1)),\rho_2(f(s_2)))$ に対して，$(\rho_1(f(s_1)),\rho_2(f(s_2))) \in R$ なる early bisimulation $R$ が存在するなら，次の条件を満足する early bisimulation $R'$（ただし $R' \subseteq R$ かつ $(\rho_1(f(s_1)),\rho_2(f(s_2))) \in R'$）が存在する．

$$[(g(\rho_i(f(s_i)),t),g(\rho_j(f(s_j)),t')) \in R' \Rightarrow [t=t' \wedge (\rho_i(f(s_i)),\rho_j(f(s_j))) \in R']]$$

**（証明）**

$$\begin{aligned}
R_1 \stackrel{\text{def}}{=} \{&(\rho_i(f(s_i)),\rho_j(f(s_j)))| \\
&(\rho_i(f(s_i)),\rho_j(f(s_j))) \in R \wedge \\
&\rho_1(f(s_1)) -\alpha_1\rightarrow \cdots -\alpha_k\rightarrow \rho_i(f(s_i)) \wedge \\
&\rho_2(f(s_2)) -\alpha_1\rightarrow \cdots -\alpha_k\rightarrow \rho_j(f(s_j)) \wedge \\
&\alpha_k \neq e?t\} \\
R_2 \stackrel{\text{def}}{=} \{&(g(\rho_i(f(s_i)),t),g(\rho_j(f(s_j)),t))| \\
&(g(\rho_i(f(s_i)),t),g(\rho_j(f(s_j)),t)) \in R \wedge \\
&(\rho_i(f(s_i)),\rho_j(f(s_j))) \in R_1\} \\
R' \stackrel{\text{def}}{=} &R_1 \cup R_2
\end{aligned}$$

とおくと，$R_1$ の要素はすべて時間遷移に対応する入力遷移 $e?t$ 以外の入出力遷移が入射している状態対であり，$M'$ の交替性よりそれらに $e?t$ は入射していない．したがって，$R'$ に属し $e?t$ が入射している状態対は必ず $R_2$ に属していることになる．$(g(\rho_i(f(s_i)),t),g(\rho_j(f(s_j)),t')) \in R'$ ならば $(g(\rho_i(f(s_i)),t),g(\rho_j(f(s_j)),t')) \in R_2$．$R_2$ の作り方より明らかに $R'$ は上の条件を満たす．

次に $R$ が early bisimulation ならば $R'$ も early bisimulation となることを示す．以下，$(\rho_i(f(s_i)),\rho_j(f(s_j))) \in R'$ とする．

**$[(\rho_i(f(s_i)),\rho_j(f(s_j))) \in R_1$ の場合]**

☆ ある状態対から異なる時間経過した状態どうしが入出力動作に関して等価な場合に，それらの関係も $R$ に加えた場合に相当．

$\rho_i(f(s_i)) -e?t\rightarrow \rho'_i(f(s'_i))$ ならば $(\rho_i(f(s_i)),\rho_j(f(s_j))) \in R$ と $R$ が early bisimulation であることより，$\rho_j(f(s_j)) -e?t\rightarrow \rho'_j(f(s'_j))$ なる $\rho'_j(f(s'_j))$ が存在し，$(\rho'_i(f(s'_i)),\rho'_j(f(s'_j))) \in R$．$e?t$ 遷移の決定性より $\rho'_i(f(s'_i)) = g(\rho_i(f(s_i)),t)$ かつ $\rho'_j(f(s'_j)) = g(\rho_j(f(s_j)),t)$．ゆえに $(\rho'_i(f(s'_i)),\rho'_j(f(s'_j))) \in R_2 \subseteq R'$．$M'$ の交替性より $\rho_i(f(s_i)) -a\$v\rightarrow \rho'_i(f(s'_i))$ $(a \neq e, \$ \in \{!,?\})$ の場合はありえない．$\rho_j(f(s_j)) -a\$v\rightarrow \rho'_j(f(s'_j))$ の場合も同様．

**$[(\rho_i(f(s_i)),\rho_j(f(s_j))) \in R_2$ の場合]**

$\rho_i(f(s_i)) -a\$v\rightarrow \rho'_i(f(s'_i))$ $(a \neq e, \$ \in \{!,?\})$ ならば $\rho_j(f(s_j)) -a\$v\rightarrow \rho'_j(f(s'_j))$ なる $\rho'_j(f(s'_j))$ が存在し，$(\rho'_i(f(s'_i)),\rho'_j(f(s'_j))) \in R_1 \subseteq R'$ となることは，上と同様に示せる．$\rho_j(f(s_j)) -a\$v\rightarrow \rho'_j(f(s'_j))$ を仮定しても同様．なお，交替性より $\rho_i(f(s_i)) -e?t\rightarrow \rho'_i(f(s'_i))$ や $\rho_j(f(s_j)) -e?t\rightarrow \rho'_j(f(s'_j))$ の場合はありえない． □

記号的遷移グラフ $M'$ の具体的遷移システムの各状態 $\rho(f(s))$ の集合は，交替性より $e?t$ 遷移が出ている状態の集合 $Idle(M') \stackrel{\text{def}}{=} \{\rho(f(s))|\exists t\exists\rho'\exists s'\rho(f(s)) -e?t\rightarrow \rho'(f(s'))\}$ とそうでない状態の集合 $Active(M') \stackrel{\text{def}}{=} \{\rho(f(s))|\exists t\exists\rho'\exists s'\rho'(f(s')) -e?t\rightarrow \rho(f(s))\}$ に 2 分割される．これらの集合の $f()$ による原像をそれぞれ $Idle(M)$，$Active(M)$ とする．

データ付時間オートマトン $M$ の具体的遷移システムにおいて $(\rho_1(s_1),\rho_2(s_2)) \in R$ なる時間双模倣 $R$ が存在したとする．$R' \stackrel{\text{def}}{=} \{(\rho_i(f(s_i)),\rho_j(f(s_j)))|(\rho_i(s_i),\rho_j(s_j)) \in R\}$ とおいたとき，$R'$ が $M'$ において $(\rho_1(f(s_1)),\rho_2(f(s_2))) \in R'$ なる early bisimulation となるようにしたい．しかし，$R$ として $(\rho_i(s_i),\rho_j(s_j)) \in R$ の状態対の一方が $Idle(M)$ に属し，もう一方が $Active(M)$ に属する場合，上記の関係は成り立たない．たとえば $(\rho_1(s_1),\rho_2(s_2)) \in R$，かつ $\rho_1(s_1) -0\rightarrow \rho'_1(s'_1)$ なら，時間双模倣の定義より $(\rho'_1(s'_1),\rho_2(s_2)) \in R$ であったとしても $R$ は時間双模倣である．しかし，$(\rho'_1(f(s'_1)),\rho_2(f(s_2))) \in R'$ なら，$\rho'_1(f(s'_1)) \in Active(M)$ かつ $\rho_2(f(s_2)) \in Idle(M)$ となり，交替性の性質より $R'$ は early bisimulation ではない．このため，以下ではまず $(\rho_1(s_1),\rho_2(s_2)) \in R$ なる任意の時間双模倣 $R$ に対して，「$(\rho_i(s_i),\rho_j(s_j)) \in R_t$ ならば，$\rho_i(s_i) \in Idle(M)$ であるときかつそのときに限り $\rho_j(s_j) \in Idle(M)$)」となるような時間双模倣 $R_t(\subseteq R)$ を構成できることを示す．

**補題 3.2** データ付時間オートマトン $M$ の具体的遷移システムの任意の状態対 $(\rho_1(s_1),\rho_2(s_2))$ に対して，

$(\rho_1(s_1), \rho_2(s_2)) \in R$ なる時間双模倣 $R$ が存在するならば，特に次の条件を満足するような時間双模倣 $R_t$（ただし $R_t \subseteq R$ かつ $(\rho_1(s_1), \rho_2(s_2)) \in R_t$）が存在する．

$$[(\rho_i(s_i), \rho_j(s_j)) \in R_t \Rightarrow [\rho_i(s_i) \in Idle(M) \Leftrightarrow \rho_j(s_j) \in Idle(M)]]$$

**（証明）**

$$\begin{aligned} R_t \stackrel{\text{def}}{=} \{&(\rho_i(s_i), \rho_j(s_j))| \\ &(\rho_i(s_i), \rho_j(s_j)) \in R \wedge \\ &\rho_1(s_1) -\alpha_1\rightarrow \cdots -\alpha_k\rightarrow \rho_i(s_i) \wedge \\ &\rho_2(s_2) -\alpha_1\rightarrow \cdots -\alpha_k\rightarrow \rho_j(s_j) \wedge \\ &\rho_i(s_i) \in Idle(M) \Leftrightarrow \rho_j(s_j) \in Idle(M)\} \end{aligned}$$

とおけば，$R_t$ は補題の条件を満たすことは定義より明らか．以下，$R_t$ が時間双模倣であることを示す．$(\rho_i(s_i), \rho_j(s_j)) \in R_t$ とする．$\rho_i(s_i) -a\$v\rightarrow \rho'_i(s'_i)$ ならば，$(\rho_i(s_i), \rho_j(s_j)) \in R$ および $R$ が時間双模倣より $\rho_j(s_j) -a\$v\rightarrow \rho'_j(s'_j)$ なる $\rho'_j$，$s'_j$ が存在して $(\rho'_i(s'_i), \rho'_j(s'_j)) \in R$ となる．$(\rho_i(s_i), \rho_j(s_j)) \in R_t$ より $\rho_1(s_1) -\alpha_1\rightarrow \cdots -\alpha_k\rightarrow \rho_i(s_i)$．よって $\rho_1(s_1) -\alpha_1\rightarrow \cdots -\alpha_k\rightarrow -a\$v\rightarrow \rho'_i(s'_i)$．$\rho'_j(s'_j)$ についても同様．また，$M$ の交替性と $f()$ の定義より明らかに $\rho'_i(f(s'_i))$ は $e?t$ 遷移を実行可能．すなわち，$\rho'_i(s'_i) \in Idle(M)$．同様に $\rho'_j(s'_j) \in Idle(M)$ も成り立つ．ゆえに $(\rho'_i(s'_i), \rho'_j(s'_j)) \in R_t$．$\rho_j(s_j) -a\$v\rightarrow \rho'_j(s'_j)$ を仮定しても同様．

また，$\rho_i(s_i) -t\rightarrow \rho'_i(s'_i)$ ならば，上と同様に $\rho_j(s_j) -t\rightarrow \rho'_j(s'_j)$ なる $\rho'_j$，$s'_j$ が存在して $(\rho'_i(s'_i), \rho'_j(s'_j)) \in R$ となる．$\rho_1(s_1) -\alpha_1\rightarrow \cdots -\alpha_k\rightarrow -t\rightarrow \rho'_i(s'_i)$ かつ $\rho_2(s_2) -\alpha_1\rightarrow \cdots -\alpha_k\rightarrow -t\rightarrow \rho'_j(s'_j)$ であることも明らか．また，$\rho_i(s_i) \in Idle(M)$ ならば $f()$ の定義と $\rho_i(s_i) -t\rightarrow \rho'_i(s'_i)$ より，$\rho_i(f(s_i)) -e?t\rightarrow \rho'_i(f(s'_i))$．$R_t$ の定義と $(\rho_i(s_i), \rho_j(s_j)) \in R_t$ より $\rho_j(s_j) \in Idle(M)$．したがって，同様に $\rho_j(s_j) -t\rightarrow \rho'_j(s'_j)$ より $\rho_j(f(s_j)) -e?t\rightarrow \rho'_j(f(s'_j))$．ゆえに交替性より明らかに $\rho'_i(s'_i) \notin Idle(M)$ かつ $\rho'_j(s'_j) \notin Idle(M)$．一方，$\rho_i(s_i) \notin Idle(M)$ ならば交替性より $\rho''_i(f(s''_i)) -e?t''_i\rightarrow \rho_i(f(s_i))$ なる $\rho''_i$，$s''_i$，$t''_i$ が存在する．$\rho_i(s_i) -t\rightarrow \rho'_i(s'_i)$ と $f()$ の定義と時間連続性より $\rho''_i(f(s''_i)) -e?(t''_i+t)\rightarrow \rho'_i(f(s'_i))$ が成り立つ．交替性より $\rho'_i(f(s'_i))$ は $e?t$ 遷移を持たない，すなわち，$\rho'_i(s'_i) \notin Idle(M)$．同様に $\rho_j(s_j) \notin Idle(M)$ と $\rho_j(s_j) -t\rightarrow \rho'_j(s'_j)$ より $\rho''_j(f(s''_j)) -e?(t''_j+t)\rightarrow \rho'_j(f(s'_j))$ なる $\rho''_j$，$s''_j$，$t''_j$ が存在し，ゆえに $\rho'_j(s'_j) \notin Idle(M)$ が成り立つ．したがって，$(\rho'_i(s'_i), \rho'_j(s'_j)) \in R_t$． □

本論文の主定理は以下の形で述べられる．

**定理 3.2** データ付時間オートマトン $M$ の任意の状態対 $(s_1, s_2)$ および任意の代入 $\rho$ に対して，以下の 2 つの命題は同値である．

(1) $\rho(s_1)$ と $\rho(s_2)$ が時間双模倣である．

(2) $\rho(f(s_1))$ と $\rho(f(s_2))$ が early bisimulation である．

ただし，$f()$ は定義 3.6 で定義した写像とする．

**（証明）**

**(1⇒2)** $(\rho(s_1), \rho(s_2))$ を含む時間双模倣関係を $R_t$ とする．一般性を失わず $R_t$ は補題 3.2 の条件を満足するものとする．$R_e \stackrel{\text{def}}{=} \{(\rho_i(f(s_i)), \rho_j(f(s_j)))|\ (\rho_i(s_i), \rho_j(s_j)) \in R_t\ \}$ とし，$(\rho_i(f(s_i)), \rho_j(f(s_j))) \in R_e$ かつ $\rho_i(f(s_i)) -a\$v\rightarrow \rho'_i(f(s'_i))$ と仮定する．このとき，ある $\rho'_j$ および $s'_j$ が存在して $\rho_j(f(s_j)) -a\$v\rightarrow \rho'_j(f(s'_j))$ かつ $(\rho'_i(f(s'_i)), \rho'_j(f(s'_j))) \in R_e$ であることを示す．

***a* = *e* の場合** $\rho_i(s_i) -v\rightarrow \rho'_i(s'_i)$．$R_t$ は時間双模倣関係なので，$\rho_j(s_j) -v\rightarrow \rho'_j(s'_j)$ なる $\rho'_j$，$s'_j$ が存在して $(\rho'_i(s'_i), \rho'_j(s'_j)) \in R_t$．$R_e$ の定義より $(\rho'_i(f(s'_i)), \rho'_j(f(s'_j))) \in R_e$．また，補題 3.2 の交替性の条件と $\rho_i(f(s_i)) -e?v\rightarrow \rho'_i(f(s'_i))$ より，ある $\rho''$，$s''$ が存在して $\rho_j(f(s_j)) -e?v\rightarrow \rho''(f(s''))$．$e?v$ 遷移の決定性より $\rho''(f(s'')) = \rho'_j(f(s'_j))$．

***a* ≠ *e* の場合** $f()$ の定義より $\rho_i(s_i) -a\$v\rightarrow \rho'_i(s'_i)$．$R_t$ は時間双模倣より $\rho_j(s_j) -a\$v\rightarrow \rho'_j(s'_j)$ なる $\rho'_j$，$s'_j$ が存在して $(\rho'_i(s'_i), \rho'_j(s'_j)) \in R_t$．再び $f()$ の定義より $\rho_j(f(s_j)) -a\$v\rightarrow \rho'_j(f(s'_j))$．$R_e$ の定義より $(\rho'_i(f(s'_i)), \rho'_j(f(s'_j))) \in R_e$．逆に $\rho_j(f(s_j)) -a\$v\rightarrow \rho'_j(f(s'_j))$ を仮定しても同様．

**(2⇒1)** $(\rho(f(s_1)), \rho(f(s_2)))$ を含む early bisimulation を $R_e$ とする．一般性を失わず，$R_e$ は補題 3.1 の条件を満足するものとする．$R_t \stackrel{\text{def}}{=} \{(\rho_i(s_i), \rho_j(s_j))|\ (\rho_i(f(s_i)), \rho_j(f(s_j))) \in R_e\}$ とおくと，$R_t$ は時間双模倣関係となることを示す．$(\rho_i(s_i), \rho_j(s_j)) \in R_t$ とする．$(\rho_i(f(s_i)), \rho_j(f(s_j))) \in R_e$ と $M'$ の交替性より，状態対 $(\rho_i(f(s_i)), \rho_j(f(s_j)))$ はともに時間遷移に対応する入力遷移 $-e?v\rightarrow$ のみが実行可能か，もしくはともに実行不能かのいずれかである．前者を場合 1，後者を場合 2 として以下に証明する．

**場合 1** $\rho_i(s_i) -t\rightarrow \rho'(s')$ ならば，時間決定性より $\rho'(s') = g(\rho_i(s_i), t)$．$\rho(f(s_i))$ は時間遷移に対応する入力遷移 $-e?v\rightarrow$ のみが実行可能なので，$f()$ の定義より，$\rho(f(s_i)) -e?t\rightarrow g(\rho(f(s_i)), t)$ であるはずである．$(\rho(f(s_i)), \rho(f(s_j))) \in R_e$ および $R_e$ が

early bisimulation であることより，$\rho(f(s_j)) \ -e?t\rightarrow\ g(\rho(f(s_j)),t)$ かつ $(g(\rho(f(s_i)),t), g(\rho(f(s_j)),t)) \in R_e$．再び $f()$ の定義より，$\rho(s_j) \ -t\rightarrow\ g(\rho(s_j),t)$．また，$R_t$ の定義より $(g(\rho_i(s_i),t), g(\rho_j(s_j),t))) \in R_t$．逆に $\rho_j(s_j) \ -t\rightarrow\ \rho'(s')$ を仮定しても同様．また，$\rho_i(s_i) \ -a\$v\rightarrow\ \rho'(s')$ $(a \in Act, \$ \in \{!,?\})$ の場合は $f()$ の定義と場合 1 の仮定よりありえない．$\rho_j(s_j) \ -a\$v\rightarrow\ \rho'(s')$ の場合も同様．

**場合 2** 場合 2 の仮定と $M'$ の交替性より，$\rho_i(f(s_i))$ および $\rho_j(f(s_j))$ に入射する $e?v$ タイプの遷移があるはずである☆．つまりある状態 $s_k$，$s_l$，ある代入 $\rho_k$，$\rho_l$，およびある時間値 $t_k$，$t_l$ が存在し，$\rho_k(f(s_k)) \ -\ e?t_k\ \rightarrow\ \rho_i(f(s_i))$ および $\rho_l(f(s_l)) \ -\ e?t_l\ \rightarrow\ \rho_j(f(s_j))$ が成り立つ．つまり，$\rho_i(f(s_i)) = g(\rho_k(f(s_k)), t_k)$，$\rho_j(f(s_j)) = g(\rho_l(f(s_l)), t_l)$ と表せる．このとき，$(\rho_i(f(s_i)), \rho_j(f(s_j))) \in R_e$ と補題 3.1 の条件より，$t_k = t_l$ かつ $(\rho_k(f(s_k)), \rho_l(f(s_l))) \in R_e$ が成り立つ．したがって，もし $\rho_i(s_i) \ -t\rightarrow\ \rho'_i(s'_i)$ ならば，時間決定性および時間連続性より $\rho_k(s_k) \ -\ (t_k + t)\ \rightarrow\ \rho'_i(s'_i)$ であり，場合 2 の仮定と $f()$ の定義より $\rho_k(f(s_k)) \ -e?(t_k + t)\rightarrow\ \rho'_i(f(s'_i))$ となる．$(\rho_k(f(s_k)), \rho_l(f(s_l))) \in R_e$ および $t_k = t_l$ より，$\rho_l(f(s_l)) \ -e?(t_k + t)\rightarrow\ \rho'_j(f(s'_j))$ なる $\rho'_j$，$s'_j$ が存在する．よって，$\rho_l(s_l) \ -(t_k + t)\rightarrow\ \rho'_j(s'_j)$ となる．$\rho_j(s_j) = g(\rho_l(s_l), t_k)$ および時間決定性より，$\rho_l(s_l) \ -t_k\rightarrow\ \rho_j(s_j) \ -t\rightarrow\ \rho'_j(s'_j)$ であるから，$\rho_j(s_j) \ -t\rightarrow\ \rho'_j(s'_j)$ かつ $(\rho'_i(s'_i), \rho'_j(s'_j)) \in R_t$ なる $\rho'_j$，$s'_j$ の存在が示された．逆に $\rho_j(s_j) \ -t\rightarrow\ \rho'_j(s'_j)$ を仮定しても同様．

$\rho_i(s_i) \ -\ a\$v\ \rightarrow\ \rho'_i(s'_i)$ ならば $f()$ の定義より $\rho_i(f(s_i)) \ -a\$v\rightarrow\ \rho'_i(f(s'_i))$．$(\rho_i(f(s_i)), \rho_j(f(s_j))) \in R_e$ と $R_e$ が early bisimulation より $\rho_j(f(s_j)) \ -a\$v\ \rightarrow\ \rho'_j(f(s'_j))$ かつ $(\rho'_i(f(s'_i)), \rho'_j(f(s'_j))) \in R_e$ なる $\rho_j$，$s'_j$ が存在する．したがって，再び $f()$ の定義より $\rho_j(s_j) \ -a\$v\rightarrow\ \rho'_j(s'_j)$．また，$R_t$ の定義より $(\rho'_i(s'_i), \rho'_j(s'_j)) \in R_t$．逆に $\rho_j(s_j) \ -a\$v\rightarrow\ \rho'_j(s'_j)$ を仮定しても同様． □

定理 3.1 および定理 3.2 より，ただちに以下が成立する．

**系 3.1** データ付時間オートマトンの任意の状態対 $(s_1, s_2)$ に対して，次のような論理式 $bisim_T(s_1, s_2)$ を求めるアルゴリズムが存在する．

$\rho \models bisim_T(s_1, s_2) \iff \rho(s_1)$ と $\rho(s_2)$ は時間双模倣

**（証明）** 定理 3.1 および定理 3.2 より，$bisim_T(s_1, s_2) \stackrel{\text{def}}{=} bisim_E(f(s_1), f(s_2))$ とおけばよい． □

例 3.1 で示した，図 3 の 2 つのデータ付時間オートマトンが時間双模倣となるためのパラメータに関する条件 $P$ は，系 3.1 のアルゴリズム $bisim_T()$ を用いて導出した式を手で簡約したものである．条件 $P$ の導出過程を付録 A.2 に示す．

## 4. あとがき

本論文では，時間値と入出力データを同時に含む遷移条件を記述可能なオートマトンモデル，データ付時間オートマトンを提案し，任意の 2 状態に対してそれらが双模倣等価となるような変数（パラメータ）に関する最弱の条件を自動導出する問題が，入出力データのみを扱う文献 3) の手法に帰着できることを示した．

現在のモデルでは，パラメータの値は与えられた時点，あるいは，入力された時点で決定し，それ以降変化することはない．つまり，オートマトンがレジスタのような動作とともに刻々と変化する内部変数を持つことができない．そのような内部変数が記述できるようなあるクラスの時間オートマトンモデルに対して等価性検証ができるように本研究を拡張することが今後の課題である．

## 参考文献


1) Alur, R., Courcoubetis, C. and Henzinger, T.A.: The Observational Power of Clocks, *Proc. CONCUR '94*, Lecture Notes in Computer Science, Vol.836, pp.162–177, Springer-Verlag (1994).
2) Chen, L.: An Interleaving Model for Real-Time Systems, *Proc. 2nd Int'l Symp. on Logical Foundations of Computer Science* (*LFCS '92*), Nerode, A. and Taitslin, M. (Eds.), Lecture Notes in Computer Science, Vol.620, pp.81–92, Springer-Verlag (1992).
3) Hennessy, M. and Lin, H.: Symbolic bisimulations, *Theoret. Comput. Sci.*, Vol.138, pp.353–389 (1995).
4) Holmer, U., Larsen, K. and Wang, Y.: Deciding properties of regular timed processes, *Proc. 3rd CAV*, Lecture Notes in Computer Science, Vol.575, pp.443–453, Springer-Verlag (1991).
5) Nakata, A., Higashino, T. and Taniguchi, K.: Time-Action Alternating Model for Timed LO-


☆ $e?v$ タイプの遷移を実行不能かつ入射していないような状態は，交替性の仮定より，あるとすれば初期状態か初期状態から到達不能な状態以外には考えられない．定義より初期状態は必ず休止状態であるので $e?v$ 遷移を実行可能である．また，到達不能な状態は無視してかまわない．したがって，そのような状態は存在しないとして差し支えない．

$$bisim_E(s_i,s_j) \stackrel{\text{def}}{=} match(s_i,s_j,\emptyset)$$
$$match(s_i,s_j,W) \stackrel{\text{def}}{=} \text{return } \textstyle\bigwedge_{a\in Act,\$\in\{!,?\}}\{match\_action(a\$,s_i,s_j,W)\}$$
$$match\_action(a\$,s_i,s_j,W) \stackrel{\text{def}}{=} \text{if } \$ =! \text{ then /* 出力動作 */}$$
$$\text{let } \{\ K=\{k|s_i\ -a!E_k[P_k]\rightarrow s_{i_k}\},\ L=\{l|s_j\ -a!E_l[Q_l]\rightarrow s_{j_l}\},$$
$$M_{k,l}=match(s_{i_k},s_{j_l},W\cup\{(s_i,s_j)\})\}\ \text{in}$$
$$\text{return } \textstyle\bigwedge_{k\in K}\{P_k\Rightarrow\bigvee_{l\in L}\{Q_l\wedge[E_k=E_l]\wedge M_{k,l}\}\}$$
$$\wedge\textstyle\bigwedge_{l\in L}\{Q_l\Rightarrow\bigvee_{k\in K}\{P_k\wedge[E_k=E_l]\wedge M_{k,l}\}\}$$
$$\text{else /* 入力動作 (early) */}$$
$$\text{let } \{\ K=\{k|s_i\ -a?x_k[P_k]\rightarrow s_{i_k}\},\ L=\{l|s_j\ -a?x_l[Q_l]\rightarrow s_{j_l}\},$$
$$z=new(DVar(s_i)\cup DVar(s_j)),$$
$$M_{k,l}=match(s_{i_k},s_{j_l},W\cup\{(s_i,s_j)\})\{z/x_k,z/x_l\}\}\ \text{in}$$
$$\text{return } \forall z[\textstyle\bigwedge_{k\in K}\{P_k\{z/x_k\}\Rightarrow\bigvee_{l\in L}\{Q_l\{z/x_l\}\wedge M_{k,l}\}\}]$$
$$\wedge\forall z[\textstyle\bigwedge_{l\in L}\{Q_l\{z/x_l\}\Rightarrow\bigvee_{k\in K}\{P_k\{z/x_k\}\wedge M_{k,l}\}\}]$$

ただし，変数集合 $Var$ の任意の部分集合 $V$ に対して $new(V)$ を $V$ に含まれない適当な新しい変数を返す関数とする．

**図 5** アルゴリズム $bisim_E()$
Fig. 5 Algorithm $bisim_E()$.

TOS and its Sympolic Verification of Bisimulation Equivalence, *Proc. Joint Int'l Conf. on Formal Description Techniques for Distributed Systems and Communication Protocols, and Protocol Specification, Testing, and Verification* (*FORTE/PSTV '96*), Gotzhein, R. and Bredereke, J. (Eds.), pp.279–294, IFIP, Chapman & Hall (1996).

## 付　　録

### A.1 アルゴリズム $bisim_E()$

ここでは，文献 3) で提案されたアルゴリズム $bisim_E()$ の具体的な内容について簡単に述べる．

$bisim_E()$ の内容は**図 5** のとおりである．図 5 で，関数 $match(s_i,s_j,W)$ は状態対 $(s_i,s_j)$ が集合 $W$ に属するならばその状態対はすでに訪れていると見なして true を返し，さもなければ，各入出力動作名 $a\$$ ($\$\in\{!,?\}$) に対して $match\_action(a\$,s_i,s_j,W)$ の値を求め，それらの論理積をとった式を返す．関数 $match\_action(a\$,s_i,s_j,W)$ は $s_i$ と $s_j$ がお互いに動作 $a\$$ を模倣可能で，動作 $a\$$ を実行後の状態対も双模倣等価であるための条件式を返す．具体的には次のように求める．集合 $K$，$L$ をそれぞれ，$s_i$，$s_j$ が実行可能な $a\$$ 遷移（の添字）の集合とする．出力動作の場合，

$$K=\{k|s_i\ -a!E_k[P_k]\rightarrow s_{i_k}\}$$
$$L=\{l|s_j\ -a!E_l[Q_l]\rightarrow s_{j_l}\}$$

となる．次に状態対 $(s_i,s_j)$ を訪問済として状態対 $(s_{i_k},s_{j_l})$ それぞれを双模倣等価にする条件式 $M_{k,l}$ を再帰的に求める．それらの式を用いて，以下の式

$$\textstyle\bigwedge_{k\in K}\{P_k\Rightarrow\bigvee_{l\in L}\{Q_l\wedge[E_k=E_l]\wedge M_{k,l}\}\}$$
$$\wedge\textstyle\bigwedge_{l\in L}\{Q_l\Rightarrow\bigvee_{k\in K}\{P_k\wedge[E_k=E_l]\wedge M_{k,l}\}\}$$

を $match\_action(a\$,s_i,s_j,W)$ の値として返す．この式を満たす任意の代入 $\rho$ は次の条件を満足する．

「任意の $k\in K$ に対して，もし代入 $\rho$ の下で $s_i$ から $k$ 番目の $a!$ 遷移が実行可能ならば（$P_k$），$s_j$ から実行可能なある $a!$ 遷移 $l\in L$ が存在して（$Q_l$），出力値が等しく（$[E_k=E_l]$），遷移先の状態対も等価である（$M_{k,l}$）．$i$ と $j$ および $K$ と $L$ をそれぞれ入れ替えても同様」

また，入力動作の場合，

$$K=\{k|s_i\ -a?x_k[P_k]\rightarrow s_{i_k}\}$$
$$L=\{l|s_j\ -a?x_l[Q_l]\rightarrow s_{j_l}\}$$

となる．出力動作のときと同様に状態対 $(s_{i_k},s_{j_l})$ それぞれを双模倣等価にする条件式 $M_{k,l}$ を再帰的に求める．ただし，入力値が等しいことを条件に反映させるため，$s_i$，$s_j$ のパラメータに使われていない新しい変数 $z$ を導入し，各 $M_{k,l}$ に現れる入力変数 $x_k$ および $x_l$ を $z$ に置き換える．それらの式を用いて，以下の式

$$\forall z[\textstyle\bigwedge_{k\in K}\{P_k\{z/x_k\}\Rightarrow\bigvee_{l\in L}\{Q_l\{z/x_l\}\wedge M_{k,l}\}\}]$$
$$\wedge\forall z[\textstyle\bigwedge_{l\in L}\{Q_l\{z/x_l\}\Rightarrow\bigvee_{k\in K}\{P_k\{z/x_k\}\wedge M_{k,l}\}\}]$$

を $match\_action(a\$,s_i,s_j,W)$ の値として返す．この式を満たす任意の代入 $\rho$ は次の条件を満足する．

「任意の入力値 $z$ に対して次のことが成り立つ．任意の $k\in K$ に対して，もし代入 $\rho$ の下で $s_i$ から $k$ 番目の $a?$ 遷移が実行可能かつ入力値が $z$ ならば（$P_k\{z/x_k\}$），同じ入力値 $z$ を入力可能な $s_j$ からの遷移 $l\in L$ が存在して（$P_l\{z/x_l\}$），遷移先の状態対も等価である（$M_{k,l}$）．$i$ と $j$ および $K$ と $L$ をそれぞれ入れ替えた条件も同時に成り立つ」

### A.2 例 3.1 の条件 $P$ の導出過程

例 3.1 の条件 $P$ を導出する過程を以下に示す．以下

の式変形において，$P = Q$ は $P$ から $Q$ がアルゴリズムによって機械的に求まることを表し，$P \equiv Q$ は $P$ を手で簡約化した結果 $Q$ が求まったことを表す．以下の $M_{i,j}$ はそれぞれ状態対 $(s_i, s_j)$ を等価にする条件式の中間結果であり，特に $M'_{1,7}$ は状態対 $(s_1, s_7)$ を訪問済としたときの $(s_1, s_7)$ を等価とする条件式である．

$$
\begin{aligned}
P = bisim_T(s_1, s_7) &= M_{1,7} \\
M_{1,7} &= \forall u[true \Rightarrow [true \wedge M_{2,8}\{u/d_1, u/d_4\}]] \\
&\quad \wedge \forall u[true \Rightarrow [true \wedge M_{2,8}\{u/d_4, u/d_1\}]] \\
&\equiv M_{2,8} \\
&\equiv [[q < p] \wedge [w = 1 \wedge z = 1.5]] \\
&\quad \vee [[p = 0 \wedge w = 0] \\
&\qquad \wedge [1.5 \le z \wedge z = q \wedge z \le 2]] \\
M_{2,8} &= \forall u_1[p \le u_1 \le q \Rightarrow \\
&\quad [w \le u_1 \le z \wedge M_{3,9}\{u_1/x, u_1/y\}]] \\
&\quad \wedge \forall u_1[1 \le u_1 \le 1.5 \Rightarrow \\
&\quad [w \le u_1 \le z \wedge M_{5,9}\{u_1/x, u_1/y\}]] \\
&\quad \wedge \forall u_1[w \le u_1 \le z \Rightarrow \\
&\quad [[1 \le u_1 \le 1.5 \wedge M_{5,9}\{u_1/y, u_1/x\}] \\
&\quad \vee [p \le u_1 \le q \wedge M_{3,9}\{u_1/y, u_1/x\}]]] \\
&\equiv \forall u_1[p \le u_1 \le q \Rightarrow \\
&\qquad [w \le u_1 \le z \wedge [u_1 + p = u_1] \wedge \\
&\qquad [u_1 \le 6 \iff 0 \le u_1 \le 5] \wedge \\
&\qquad [u_1 \le 2] \wedge [u_1 = u_1]]] \\
&\quad \wedge \forall u_1[1 \le u_1 \le 1.5 \Rightarrow \\
&\qquad [w \le u_1 \le z \wedge [u_1 = u_1] \wedge \\
&\qquad [u_1 \le 8 \iff 0 \le u_1 \le 5] \wedge \\
&\qquad [u_1 \le 2] \wedge [u_1 = u_1]]] \\
&\quad \wedge \forall u_1[w \le u_1 \le z \Rightarrow \\
&\qquad [[1 \le u_1 \le 1.5 \wedge [u_1 = u_1] \wedge \\
&\qquad [u_1 \le 8 \iff 0 \le u_1 \le 5] \wedge \\
&\qquad [u_1 \le 2] \wedge [u_1 = u_1]] \\
&\qquad \vee [p \le u_1 \le q \wedge [u_1 + p = u_1] \wedge \\
&\qquad [u_1 \le 6 \iff 0 \le u_1 \le 5] \wedge \\
&\qquad [u_1 \le 2] \wedge [u_1 = u_1]]]] \\
&\quad \ldots (\text{中略}) \\
&\equiv [[q < p] \wedge [w = 1 \wedge z = 1.5]] \\
&\quad \vee [[p = 0 \wedge w = 0] \wedge \\
&\qquad [1.5 \le z \wedge z = q \wedge z \le 2]] \\
M_{3,9} &= \forall u_2[u_2 \le x + p \Rightarrow \\
&\quad [u_2 \le y \wedge M_{4,10}\{u_2/d_2, u_2/d_5\}]] \\
&\quad \wedge \forall u_2[u_2 \le y \Rightarrow \\
&\quad [u_2 \le x + p \wedge M_{4,10}\{u_2/d_5, u_2/d_2\}]] \\
&\equiv [x + p = y \wedge M_{4,10}] \\
&\equiv [x + p = y] \wedge [x \le 6 \iff 0 \le y \le 5] \\
&\quad \wedge [y \le 2] \wedge [x = y] \\
M_{4,10} &= [x \le 6 \Rightarrow [[0 \le y \le 5 \wedge x = y \wedge M'_{1,7}] \\
&\quad \vee [y > 2 \wedge x = y \wedge M_{1,11}]]] \\
&\quad \wedge [0 \le y \le 5 \Rightarrow \\
&\quad [x \le 6 \wedge x = y \wedge M'_{1,7}]] \\
&\quad \wedge [y > 2 \Rightarrow [x \le 6 \wedge x = y \wedge M_{1,11}]] \\
&\equiv [x \le 6 \iff 0 \le y \le 5] \wedge \\
&\quad [y \le 2] \wedge [x = y] \\
M_{1,11} &= false \\
M'_{1,7} &= true \\
M_{5,9} &= \forall u_3[u_3 \le x \Rightarrow [u_3 \le y \wedge M_{6,10}]] \\
&\quad \wedge \forall u_3[u_3 \le y \Rightarrow [u_3 \le x \wedge M_{6,10}]] \\
&\equiv [x = y \wedge M_{6,10}] \\
&\equiv [x = y] \wedge [x \le 8 \iff 0 \le y \le 5] \\
&\quad \wedge [y \le 2] \\
M_{6,10} &= [x \le 8 \Rightarrow [[0 \le y \le 5 \wedge \\
&\quad x = y \wedge M'_{1,7}] \\
&\quad \vee [y > 2 \wedge x = y \wedge M_{1,11}]]] \\
&\quad \wedge [0 \le y \le 5 \Rightarrow \\
&\quad [x \le 8 \wedge x = y \wedge M'_{1,7}]] \\
&\quad \wedge [y > 2 \Rightarrow [x \le 8 \wedge x = y \wedge M_{1,11}]] \\
&\equiv [x \le 8 \Rightarrow [0 \le y \le 5] \wedge [x = y]] \\
&\quad \wedge [0 \le y \le 5 \Rightarrow [x \le 8] \wedge [x = y]] \\
&\quad \wedge [y > 2 \Rightarrow false \wedge [x = y]] \\
&\equiv [x \le 8 \iff 0 \le y \le 5] \wedge \\
&\quad [y \le 2] \wedge [x = y]
\end{aligned}
$$

(平成 11 年 7 月 26 日受付)
(平成 12 年 7 月 5 日採録)

**中田 明夫**（正会員）

平成 4 年大阪大学基礎工学部情報工学科卒業．平成 9 年同大大学院基礎工学研究科物理系専攻博士後期課程修了．博士（工学）．同年広島市立大学情報科学部助手．現在，大阪大学大学院基礎工学研究科助手．実時間システムや分散システムの仕様記述と検証法，プロセス代数，時相論理等の研究に従事．

**服部　哲**

平成 4 年大阪大学基礎工学部情報工学科卒業．平成 11 年同大大学院基礎工学研究科物理系専攻博士後期課程退学．同年奈良先端科学技術大学院大学情報科学研究科助手．現在，北陸先端科学技術大学院大学情報科学研究科助手．博士（工学）．項書換え系，実時間システムの仕様記述と検証，ソフトウェア工学等の研究に従事．

**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 年同大大学院博士課程修了．工学博士．同年同大助手．平成 2，6 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，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．

**谷口　健一**（正会員）

昭和 40 年大阪大学工学部電子工学科卒業．昭和 45 年同大大学院基礎工学研究科博士課程修了．工学博士．現在，同大大学院基礎工学研究科教授．この間，計算理論，ソフトウェアやハードウェアの仕様記述・実現・検証の代数的手法および支援システム，関数型言語の処理系，分散システムや通信プロトコルの設計・検証法等に関する研究に従事．