マルチメディア通信と分散処理 89－4
（1998. 6. 3）

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

新田 直子† 山口 弘純† 中田 明夫‡ 東野 輝夫† 谷口 健一†

†大阪大学 大学院基礎工学研究科 情報数理系専攻
〒560-8531 大阪府豊中市待兼山町 1-3
‡広島市立大学 情報科学部 ソフトウェア工学講座
〒731-3194 広島市安佐南区大塚東 3-4-1

本稿では M.Hennessy らが提案したデータ付きプロセスの双模倣等価性の記号的検証法と，我々が提案した時間プロセスの双模倣等価性の記号的検証法を組み合わせて，データと時間に関する制約を両方記述できるような一つのプロセスモデル (データ付時間オートマトンと呼ぶ) 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双模倣等価性を検証する際に，動作に入出力値などのデータや時間に関する制約が付けられていると，意味モデル上では無限個の遷移を生成してしまう．そこで遷移先が同じであるような動作群を一つの遷移にまとめ，もとの無限グラフを有限グラフに変換して検証する．双模倣等価性 $t \simeq^b u$ は，ブール式 b を満たすようなデータ値や時間値の組に対してのみ，状態 t と u は双模倣等価であることを表す．本研究では与えられた2つの状態 t と u に対して $t \simeq^b u$ を満たすような最も弱いブール式 $mgb(t,u)$ を求めるアルゴリズムを示す．本手法の適用例として，マルチメディア通信用多重化プロトコルの遷移条件の変更に際して非時間的双模倣等価性が保たれるかどうかを検証した．

Symbolic Verification of Bisimulation Equivalence for Timed Automata with Data Values

Naoko NITSUTA† Hirozumi YAMAGUCHI† Akio NAKATA‡
Teruo HIGASHINO† Kenichi TANIGUCHI†

†: Dept. Information and Computer Sciences, Faculty of Engineering Science,
Osaka University, Machikaneyama 1-3, Toyonaka, Osaka 560-8531, Japan
‡: Dept. of Computer Science, Faculty of Information Sciences, Hiroshima City University,
Ozuka-higashi 3-4-1, Asaminami-ku, Hiroshima 731-3194, Japan

In this paper, we propose a symbolic verification method to verify bisimulation equivalence of two timed automata with data values by combining M. Hennessy's method to verify bisimulation equivalence of two data-passing processes and our proposed method to verify bisimulation equivalence of two timed processes. We use the timed automata with data values to describe data-passing timed processes. In general, it costs infinitely to verify bisimulation equivalence of two timed automata with data values, if the range of data values and/or time domain is infinite. Therefore, we classify the infinite number of transitions from a state into the finite number of groups, each of which moves the same state. We use $t \simeq^b u$ to express that t is bisimulation equivalent to u for any interpretation which satisfies b. In this paper, we propose an algorithm which obtains the weakest condition $mgb(t,u)$ such that $t \simeq^{mgb(t,u)} u$ holds for a given state-pair t and u. Using the proposed method, we can verify timed and untimed bisimulation equivalence for timed automata with data values. A practical example is also given.

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なわれている [3, 4, 5]．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，これらの値が変更され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るため，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．

この問題を解決するためにいくつかの提案がされてきた．その中で M. Hennessy らが提案し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 [1] 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 [2]．しかし，前者では時間値を扱うことはできず，後者では入出力データを扱うことができない．

そこで本稿では，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共に記述できるような一つのプロセスモデル (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．プロセスの等価性にはいくつかの定義が存在するが，本研究では通信プロセスの基本的な等価性である双模倣等価性の一つである

late 双模倣等価性 [1] に基づいて考える．この方法により，入出力データと時間の両方に依存するような遷移条件を持ったシステムを扱うことができ，それらの値を変更した場合に等価な動作をするかどうかの判定を機械的に行うことができる．

本稿で用いるデータ付時間オートマトンモデルでは，時間遷移を遅延時間 d を用いて $\xrightarrow{e(d)}$ で，入出力動作などの動作遷移 γ を $\xrightarrow{\gamma}$ で表す．また，状態は休止状態と活動状態の 2 種類から成る．ここで，休止状態は遷移として時間遷移のみを持ち遷移先は活動状態である状態を示し，活動状態は遷移として動作遷移のみを持ち遷移先は休止状態である状態を示す．各状態は 0 個以上のパラメータ (x,y など) を持つ．また，これらの遷移は遷移条件をもち，この条件は遅延時間値や遷移前の状態のパラメータ値，入力値に関する任意の述語を記述できる．等価性検証の機械化のために，それらの述語からなる任意の一階述語論理式の決定可能性のみを要求する．

ここで提案するアルゴリズムは有限状態のデータ付時間オートマトンとその状態対 (t,u) を入力とし，その状態対を双模倣等価にするような最も弱い条件 $mgb(t,u)$ を出力する．$mgb(t,u)$ が求まれば，その状態対があるパラメータの値について等価であるかどうかはその値が $mgb(t,u)$ を満たすかどうかで検証できる．また，このアルゴリズムはデータ付時間オートマトンの類似の双模倣等価性である early 双模倣等価性 [1] や非時間的双模倣等価性 [2] の検証などにも簡単に拡張できる [7].

提案したアルゴリズムの適用例として，映像，音声などのデータをパケット化し，それを可変長パケットに多重化にして送出するようなプロトコル [6] をあげる．

以下，2 章で仕様記述モデルとして用いるデータ付時間オートマトンモデルを定義し，3 章で双模倣等価性の定義を行う．4 章でデータ付時間オートマトンの等価性の検証アルゴリズムを提案する．5 章でその検証方法を使った具体的な応用例を示し，6 章でまとめを述べる．

2 データ付時間オートマトン

仕様記述モデルとしてデータ付時間オートマトンを使う．これは [2] 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パラメータ変数の集合 ($DVar(s)$) 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なってもこの状態 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 γ はある動作，d は遅延の長さを表す．状態を，遷移として時間遷移のみを持ち遷移先は活動状態であるような休止状態 (idle state) と，遷移として動作遷移のみを持ち遷移先は休止状態であるような活動状態 (active state) の 2 種類に分ける．よって，このモデルでは休止状態と活動状態が交互に現れる．$s \xrightarrow{\gamma,P} s'$ は $DVar(s)$ の変数の現在の値のもとで P を満たすとき動作 γ が実行されることを表す．動作の種類としては入出力を伴わない動作 a，出力動作 $c!x$，入力動作 $c?x$ の 3 種類があり，これらの動作の実行には時間はかからないものとする．また x は入出力値を表す変数である．また，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，また γ 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)．状態 s 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 がこの最大の値を越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 にはこの遷移が実行された時点でそ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．

本研究において任意のモデルは時間決定性である．すなわち，すべての状態は時間遷移を最大でも 1 つしか持たないものとする．また，変数 d や x は $DVar(s')$ に含まれてもよい．つまり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良い．このようなモデルをデータ付時間オートマトンモデルと呼ぶことにする．

例 1 図 1 に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 γ と遷移条件 P を表している．

このモデルは次のように動作する．まず，初期状態 s_1 において実行開始時からの遅延が 1 単位時間以上になるとゲート c からの入力が可能になる．この時この入力値が正のときのみ入力を受け付ける．変数 $d1$ には実際に入力を受け付けるまでにかかった遅延時間が割り当てられる．入力が行われると状態 s_3 に遷移する．その時点からの遅延が 5 単位時間以内に動作 a または出力 $d!x$ を実行しなければならない．その遅延時間を $d2$ とし，$d1+d2$ が 4 単位時間内に x の値をゲート d から出力した場合初期状態にもどる．$d1+d2$ が 3 単位時間以上になると動作 a が実行可能になり a の実行後状態 s_5 に遷移する．

3 双模倣等価性

双模倣等価性とは 2 つのプロセスがお互いに同じ動作を模倣し続けることができる性質のことをいう．あるプロセスを双模倣等価であるプロセスと置き換

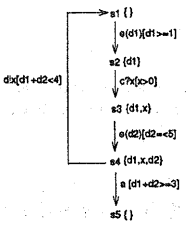

図 1: データ付時間オートマトンの例

えても外部の通信相手に対しては全く同様に振る舞うことが保証される．データ付きプロセスの双模倣等価性には，入力値を前もって予測した模倣を許す early 双模倣等価性と，現実のプロセスと同様に入力値を予測できないとする late 双模倣等価性の二種類が存在する．

また，時間プロセスの等価性にも時間的双模倣等価性と非時間的双模倣等価性の二種類がある．時間的双模倣等価性では模倣において各動作の生起時刻が一致しなければならないのに対して非時間的双模倣等価性では生起時刻は一致しなくてもよい．

ここでは紙面の都合により，late 時間的双模倣等価性の形式的な定義のみを与える．詳細は文献 [7] 参照．

定義 1 (late 時間的双模倣等価性の定義)
変数への値づけを $\rho, \rho', \cdots$ で表す．ある状態 s のパラメータにある値をつけた (ρ) 時，それを $\rho(s)$ と表す．関係 R が *late* 時間的双模倣等価であるとは，もし $(\rho_i(s_i), \rho_j(s_j)) \in R$ なら，次の条件が成り立つことである．

- 任意の $\gamma \in a, c!x, e(d)$ について

– $\rho_i(s_i) \xrightarrow{\gamma} \rho_i'(s_i')$ ならば，ある ρ_j'，s_j' が存在して $\rho_j(s_j) \xrightarrow{\gamma} \rho_j'(s_j)$ かつ $(\rho_i'(s_i'), \rho_j'(s_j')) \in R$ を満たす．
– $\rho_j(s_j) \xrightarrow{\gamma} \rho_j'(s_j')$ ならば，ある ρ_i'，s_i' が存在して $\rho_i(s_i) \xrightarrow{\gamma} \rho_i'(s_i)$ かつ $(\rho_i'(s_i'), \rho_j'(s_j')) \in R$ を満たす．

- 任意の $c?x$ について

– $\rho_i(s_i) \xrightarrow{c?x} \rho_i'(s_i')$ ならば，ある ρ_j'，s_j' が存在して任意の x について $\rho_j(s_j) \xrightarrow{c?x} \rho_j'(s_j)$ かつ $(\rho_i'(s_i'), \rho_j'(s_j')) \in R$ を満たす．
– $\rho_j(s_j) \xrightarrow{c?x} \rho_j'(s_j')$ ならば，ある ρ_i'，s_i' が存在して任意の x について $\rho_i(s_i) \xrightarrow{c?x} \rho_i'(s_i)$ かつ $(\rho_i'(s_i'), \rho_j'(s_j')) \in R$ を満たす．

ここで，$(\rho_i(s_i), \rho_j(s_j)) \in R)$ であるような関係 R があった場合 $\rho_i(s_i)$ と $\rho_j(s_j)$ は双模倣等価であるといい，$\rho_i(s_i) \simeq_t \rho_j(s_j)$ と表す．$\rho(s_i) \simeq_t \rho(s_j)$ である時，s_i と s_j は ρ に関して *late* 時間的双模倣等価であるという．

以下ではここで定義した late 時間的双模倣等価性について述べ，late 時間的双模倣等価性のことを単に双模倣等価性ということにする．

4 双模倣等価性の検証

定義 2 値づけ ρ の下で条件 P が成り立つ時かつその時のみ $\rho \models P$ と表す．あるブール式 b に対して $\rho \models b$ であるようなすべての ρ に関して $\rho_i(s_i) \simeq_t \rho_j(s_j)$ である時かつその時のみ $s_i \simeq_t^b s_j$ と表す．

任意の状態対 (s_i, s_j) において $\rho \models P$ ならば ρ において s_i と s_j が等価であるようなもっとも弱い条件 P を (s_i, s_j) の mgb と呼び，$mgb(s_i, s_j)$ で表す．つまり，$s_i \simeq_t^b s_j$ であるような任意の b に対し $b \rightarrow mgb(s_i, s_j)$ となり，$s_i \simeq_t^{mgb(s_i,s_j)} s_j$ である．この時，この条件が恒真であれば任意のパラメータについてこの状態対は双模倣等価であり，この条件が充足可能であればこの条件を満たすようなあるパラメータ値についてこの状態対は双模倣等価であるという．もし任意の状態対について mgb が求められれば，$\rho(s_i)$ と $\rho(s_j)$ の等価性検証は $\rho \models mgb(s_i, s_j)$ の検証に帰着できることになる．

そこで，[1]，[2] のアルゴリズムを組み合わせて，有限状態のデータ付時間オートマトンとその状態対 (t, u) からその状態対を双模倣等価にするようなもっとも弱い条件 $mgb(t, u)$ を求めるアルゴリズムを提案する．

集合 V について $new(V)$ とは $x \notin V$ であるような適当な x を返す関数であり，$s_i[x \rightarrow z]$ とは状態 s_i 後に出てくる x をすべて z に置き換えることを表すとし，次に提案するアルゴリズムを示す．

4.1 アルゴリズムの概略

mgb を求めるアルゴリズムは図 2 に示す通りである．$mgb(s_i, s_j)$ はデータ付時間オートマトンの任意の状態対 s_i と s_j を引数にとり，$mgb1(s_i, s_j, \emptyset)$ の値を求める mgb として返す．$mgb(s_i, s_j, W)$ は s_i, s_j, W を引数にとる．ここで，W はすでに扱った状態対の集合である．もし，$s_i, s_j \in W$ なら true を返す．また，もし s_i，s_j が休止状態の対なら $match\_delay(s_i, s_j, W)$ を，活動状態の対なら $match\_action(s_i, s_j, W)$ を返す．ここで，$match\_delay(s_i, s_j, W)$,$match\_action(s_i, s_j, W)$ はどちらも (s_i, s_j) に対する mgb を求める関数である．$match\_delay(s_i, s_j, W)$ で，まず s_i と s_j からの遅延を同じものにするために新しく変数 $d(= new(DVar(s_i) \cup DVar(s_j)))$ を作る．ある値付け ρ が与えられた時に，もし s_i と s_j が等価であれば s_i から遅延 v での遷移が可能であれば s_j からも遅延 v での遷移が存在し，かつ遷移先の $s_{i'}$ と $s_{j'}$ も等価でなければならない．これが，s_i と s_j を逆にした場合にも成り立たなければならないのでこの値付け ρ において s_i と s_j が等価であるための条件は

$$\forall d[P_i\{d/d_i\} \Rightarrow [P_j\{d/d_j\} \wedge M_{i',j'}]]$$

$$
\begin{array}{rcl}
mgb(s_i,s_j) & \stackrel{\text{def}}{=} & mgb1(s_i,s_j,\emptyset) \\
mgb1(s_i,s_j,W) & \stackrel{\text{def}}{=} & if (s_i,s_j) \in W \text{ then return } true \\
 & & else\ if\ (s_i,s_j)\ is\ a\ pair\ of\ idle\ states, then\ return\ match\_dalay(s_i,s_j,W) \\
 & & else\ if\ (s_i,s_j)\ is\ a\ pair\ of\ active\ states, then\ return\ match\_action(s_i,s_j,W) \\
 & & else\ return\ false \\
match\_delay(s_i,s_j,W) & \stackrel{\text{def}}{=} & if\ s_i \stackrel{e(d_i),P_i}{\longrightarrow} s_{i'}\ and\ s_j \stackrel{e(d_j),P_j}{\longrightarrow} s_{j'} \\
 & & then\ let\ \{d = new(DVar(s_i) \cup DVar(s_j)), \\
 & & \quad M_{i',j'} = mgb1(s_{i'}[d_i \to d], s_{j'}[d_j \to d], W \cup \{(s_i,s_j)\})\}\ in \\
return & & \forall d[P_i\{d/d_i\} \Rightarrow [P_j\{d/d_j\} \wedge Mi',j']] \wedge \forall d[P_j\{d/d_j\} \Rightarrow [P_i\{d/d_i\} \wedge Mi',j']] \\
 & & else\ if\ s_i \stackrel{e(d_i),P_i}{\not\longrightarrow}\ and\ s_j \stackrel{e(d_j),P_j}{\not\longrightarrow} \\
 & & then\ return\ true\ else\ return\ false \\
match\_action(s_i,s_j,W) & \stackrel{\text{def}}{=} & return \bigwedge_{\gamma \in SyAct} \{match\_action1(\gamma,s_i,s_j,W)\} \\
match\_action1(a,s_i,s_j,W) & \stackrel{\text{def}}{=} & let\{K = \{k \mid s_i \stackrel{a,P_k}{\longrightarrow} s_{i_k}\}, \\
 & & \quad L = \{l \mid s_j \stackrel{a,Q_l}{\longrightarrow} s_{j_l}\}, \\
 & & \quad M_{k,l} = mgb1(s_{i_k},s_{j_l},W \cup \{(s_i,s_j)\})\}\ in \\
return & & \bigwedge_{k \in K}\{P_k \Rightarrow \bigvee_{l \in L}\{Q_l \wedge M_{k,l}\}\} \wedge \bigwedge_{l \in L}\{Q_l \Rightarrow \bigvee_{k \in K}\{P_k \wedge M_{k,l}\}\} \\
match\_action1(c!,s_i,s_j,W) & \stackrel{\text{def}}{=} & let\{K = \{k|s_i \stackrel{c!e_i,P_k}{\longrightarrow} s_{i_k}\}, \\
 & & \quad L = \{l|s_j \stackrel{c!e_j,Q_l}{\longrightarrow} s_{j_l}\}, \\
 & & \quad M_{k,l} = mgb1(s_{i_k},s_{j_l},W \cup \{(s_i,s_j)\})\}\ in \\
return & & \bigwedge_{k \in K}\{P_k \Rightarrow \bigvee_{l \in L}\{Q_l \wedge e_i = e_j \wedge M_{k,l}\}\} \wedge \bigwedge_{l \in L}\{Q_l \Rightarrow \bigvee_{k \in K}\{P_k \wedge e_i = e_j \wedge M_{k,l}\}\} \\
match\_action1(c?,s_i,s_j,W) & \stackrel{\text{def}}{=} & let\{z = new(DVar(s_i) \cup DVar(s_j)), \\
 & & \quad K = \{k|s_i \stackrel{c?x,P_k}{\longrightarrow} s_{i_k}\}, \\
 & & \quad L = \{l|s_j \stackrel{c?y,Q_l}{\longrightarrow} s_{j_l}\}, \\
 & & \quad M_{i_k,j_l} = mgb1(s_{i_k}[x \to z], s_{j_l}[y \to z], W \cup \{(s_i,s_j)\})\}\ in
\end{array}
$$

$$
return\ \{\bigwedge_{k \in K}\{\bigvee_{l \in L}\{\forall z\{P_k\{z/x\} \Rightarrow \{Q_l\{z/y\} \wedge M_{i_k,j_l}\}\}\}\} \wedge \{\bigwedge_{l \in L}\{\bigvee_{k \in K}\{\forall z\{Q_l\{z/y\} \Rightarrow \{P_k\{z/x\} \wedge M_{i_k,j_l}\}\}\}\}\}
$$

図 2: アルゴリズム

$$
\wedge \forall d[P_j\{d/d_j\} \Rightarrow [P_i\{d/d_i\} \wedge M_{i',j'}]] \qquad (1)
$$

となる (ここで，$M_{i',j'} = mgb(s_{i'}[d_i \to d], s_{j'}[d_j \to d])$)．また，$s_i$ と s_j が等価でない場合，ある遅延 v' に対して s_i からは遷移が可能であるのに s_j からは可能でない，または $s_{i'}$ と s_j が等価でないかのどちらかである．どちらの場合でも (1) は成り立たないので式 (1) が $\rho(s_i)$，$\rho(s_j)$ を等価にするための最も一般的な条件式，つまり (s_i,s_j) に対する mgb となる．次に，$match\_delay(s_i,s_j,W)$ は $M_{i',j'} = mgb1(s_{i'},s_{j'},W \cup \{(s_i,s_j)\})$ を再帰的に計算し，最終的に (s_i,s_j) に対する mgb を返す．$match\_action1(a,s_i,s_j,W)$ は次のような動作をする．任意の値付け ρ，任意の入出力を伴わない動作 a に対し s_i と s_j が等価である時，$\rho \models P_k$ であるような P_k に対して $s_i \stackrel{a,P_k}{\longrightarrow} s_{i_k}$ という遷移があり，動作 a が実行可能であれば $\rho \models Q_l$ であるような Q_l に対して $s_j \stackrel{a,Q_l}{\longrightarrow} s_{j_l}$ という遷移が存在し，かつ遷移先の s_{i_k} と s_{j_l} も等価でなければならない．これも同様に，s_i と s_j を逆にした場合にも成り立たなければならないのでこの値付け ρ において s_i と s_j が等価であるための条件は $K = \{k|s_i \stackrel{a,P_k}{\longrightarrow} s_{i_k}\}$, $L = \{l|s_j \stackrel{a,Q_l}{\longrightarrow} s_{j_l}\}$, $M_{i_k,j_l} = mgb1(s_{i_k},s_{j_l},W \cup \{(s_i,s_j)\})$ とすると

$$
\begin{array}{l}
\bigwedge_{k \in K}\{P_k \Rightarrow \bigvee_{l \in L}\{Q_l \wedge M_{i_k,j_l}\}\} \\
\wedge \bigwedge_{l \in L}\{Q_l \Rightarrow \bigvee_{k \in K}\{P_k \wedge M_{i_k,j_l}\}\}
\end{array} \qquad (2)
$$

となる．また，$match\_action1(c!,s_i,s_j,W)$ は任意の値付け ρ，任意の出力動作 $c!$ に対し動作 a の場合と同じ様に考えられる．ただ，出力されるものが 2 つの状態対で同じでなければならないので，式 (2) は $K = \{k|s_i \stackrel{c!e_i,P_k}{\longrightarrow} s_{i_k}\}$, $L = \{l|s_j \stackrel{c!e_j,Q_l}{\longrightarrow} s_{j_l}\}$, $M_{i_k,j_l} = mgb1(s_{i_k},s_{j_l},W \cup \{(s_i,s_j)\})$ とすると

$$
\begin{array}{l}
\bigwedge_{k \in K}\{P_k \Rightarrow \bigvee_{l \in L}\{Q_l \wedge e_i = e_j \wedge M_{i_k,j_l}\}\} \\
\wedge \bigwedge_{l \in L}\{Q_l \Rightarrow \bigvee_{k \in K}\{P_k \wedge e_i = e_j \wedge M_{i_k,j_l}\}\}
\end{array} \qquad (3)
$$

となる．$match\_action1(c?, s_i, s_j, W)$ も任意の値付け ρ，任意の入力動作 $c?$ に対し同様に考えるが，ここで入力される変数は状態 s_i，s_j で同じものかつそれぞれ値が与えられていないものでなければならないので，新しく変数 $z(\notin (DVar(s_i) \cup DVar(s_j))$ を作る．この上で式 (2) は $K = \{k | s_i \xrightarrow{c?x, P_k} s_{i_k}\}$, $L = \{l | s_j \xrightarrow{c?y, Q_l} s_{j_l}\}$, $M_{i_k,j_l} = mgb1(s_{i_k}[x \to z], s_{j_l}[y \to z], W \cup \{(s_i, s_j)\})$ とすると

$$\bigwedge_{k \in K} \{ \bigvee_{l \in L} \{ \forall z \{ P_k \{z/x\} \Rightarrow \{ Q_l \{z/y\} \wedge M_{i_k,j_l} \} \} \} \}$$
$$\wedge \bigwedge_{l \in L} \{ \bigvee_{k \in K} \{ \forall z \{ Q_l \{z/y\} \Rightarrow \{ P_k \{z/x\} \wedge M_{i_k,j_l} \} \} \} \} \quad (4)$$

となる．これをすべての動作 γ に対してあてはめたものの和をとったものが値付け ρ に対して s_i と s_j が等価であるための条件となる．

一方，s_i と s_j が等価でない場合，ある動作 γ' に対して $s_i \xrightarrow{\gamma', P_k} s_{i_k}$ という遷移が可能な時，$s_j \xrightarrow{\gamma', Q_l} s_{j_l}$ という遷移が存在しない，または s_{i_k} と s_{j_l} が等価でないかのどちらかである．どちらの場合でも (2)，(3)，(4) は成り立たないので，式 (2)，(3)，(4) をすべての動作 γ に対して当てはめたものの和が $\rho(s_i)$ と $\rho(s_j)$ を等価にするための最も一般的な条件式，つまり (s_i, s_j) に対する mgb となる．$match\_action1(\gamma, s_i, s_j, W)$ は M_{i_k,j_l} を再帰的に計算し，最終的に式 (2)(ある動作 γ に対する (s_i, s_j) の mgb) を返す．$match\_action(s_i, s_j, W)$ は $match\_action1(\gamma, s_i, s_j, W)$ の和をとり，(s_i, s_j) に対する mgb を返す．

5 実用プロトコルへの適用

本研究で提案したアルゴリズムを適用するような実世界に存在するような具体例を示す．

映像，音声，データの3つのメディアからなるユーザデータを各メディアごとにパケット化し，それを可変長パケットに多重化にして送出するパケット多重化方式 H.223 [6] を簡単化した例である．

この方式では一パケットに多重化されるサイズはメディアごとに異なる．したがってメディアごとにパケット化遅延が異なる．なお，ここではパケットは一定周期ごとに送信すると仮定し，ビットレートなどの理由でパケット化が間に合わないメディアに関しては次パケットで送信するとしている．

以下を仮定する．

- 送信するパケットは可変長であるとする．H.223 方式ではパケット化遅延を吸収するため，16種類の多重化のパターン（どのメディアがパケットのどの位置に格納されるかを表す）を予めパターンテーブルとして決定しておき，H.245 方式に基づくプロトコルにより受信側に予め送信しておく．送信側では各パケットごとにパターンテーブルの多重化パターンのいずれかを選択することができる．ここでは受け取ったパケットサイズに近い多重化パターンが定義されているものとする．
- 受信側で安定した再生を行うためには到着レートが一定であることが望ましい．従って送信側で送信レートが一定となるよう，パケットは一定の間隔 p で送信される．
- 各メディアストリームのパケット化単位は予め決定されている最大パケット長を超えないとする．あるメディアのパケット化遅延が大きくなったため3つのメディアすべてを送出時刻までにパケット化することができないと判断した場合，残りのメディアは受け取らず，パターンの残りのデータ部分にダミー情報を入れて送信する．

動作の概略を説明する．音声データを p_a，映像データを p_v，それ以外のデータを p_d とおき，送信するパケットの最大パケット長を s_{max}，ヘッダサイズを h とする．仮定より，各メディアの最大パケット長をそれぞれ $s_{a_{max}}$，$s_{v_{max}}$，$s_{d_{max}}$ とすると，$s_{a_{max}} + s_{v_{max}} + s_{d_{max}} + h \leq s_{max}$，$|p_a| \leq s_{a_{max}}$，$|p_v| \leq s_{v_{max}}$，$|p_d| \leq s_{d_{max}}$ でなければならない．初期状態から3つのうちどれかのデータを受けとると，そのデータを送出は一定周期 p ごとに起こるので時間 p 以内にパケット化しなければならない．次に，今パケット化したデータ以外の種類のデータが入力されるとその入力を受け付けて，同様に送出の周期の関係より (p− 最初のデータのパケット化にかかった時間) 以内にパケット化する．ここで，(最初のデータのパケット化にかかった時間＋次の入力待ちの時間) がある一定値 $e(\leq p)$ を越えても次の入力が来なかった場合パケットを送信して良い．(パケットを送信した後は初期状態に戻る.) 2番目のデータをパケット化した後，残った種類のデータが入力されるとその入力を受け付けて (p− 最初の2つのデータのパケット化にかかった時間) 以内にパケット化する．このときも，(最初の2つのデータのパケット化にかかった時間＋次の入力待ちの時間) がある一定値 $e(\leq p)$ を越えても次の入力が来なかった場合パケットを送信して良い．(パケットを送信した後は初期状態に戻る.) 3番目のデータをパケット化した後はそのパケットを送信して初期状態に戻る．

このシステムの状態遷移表を表1に示す．

このようなシステムの場合，遷移表からも分かるように入出力データと時間値の両方を扱うようなシステムであるため，このシステムの遷移条件などの変更時に動作が変わらないことを確かめるために本手法が役立つ．例えば，表1のシステムを S,S の遷移 $s_5 \to s_0$ および $s_5 \to s_8$ の遷移条件をそれぞれ $true$, $d_a < e \wedge |p_v| \leq s_{v_{max}}$ に変更したシステムを S' とする．S の各状態 s_i に対して対応する S' の状態を s'_i として，例えば状態対 (s_5, s'_5) について等価性を検証する．

$$\begin{aligned} mgb(s_5, s'_5) &= mgb1(s_5, s'_5, \emptyset) \\ &= \forall z[|z| \leq s_{v_{max}} \Rightarrow [true \wedge M_{00'} \vee \\ &\quad d_a < e \wedge |z| \leq s_{v_{max}} \wedge M_{88'}]] \\ &= \forall z[e \leq d_a \Rightarrow [true \wedge M_{00'} \vee \end{aligned}$$

現在の状態	パラメータ	動作	遷移条件	遷移先
s_0	$\{s_{a_{max}}, s_{v_{max}}, s_{d_{max}}, p, e\}$	true	$e(d_0)$	s_1
s_1	$\{s_{a_{max}}, s_{v_{max}}, s_{d_{max}}, p, e\}$	$c?p_a$ $c?p_v$ $c?p_d$	$\lvert p_a\rvert \leq s_{a_{max}}$ $\lvert p_v\rvert \leq s_{v_{max}}$ $\lvert p_d\rvert \leq s_{d_{max}}$	s_2 s_3 s_4
s_2	$\{s_{a_{max}}, s_{v_{max}}, s_{d_{max}}, p, e, p_a\}$	$e(d_a)$	$d_a \leq p$	s_5
s_3	$\{s_{a_{max}}, s_{v_{max}}, s_{d_{max}}, p, e, p_v\}$	$e(d_v)$	$d_v \leq p$	s_6
s_4	$\{s_{a_{max}}, s_{v_{max}}, s_{d_{max}}, p, e, p_d\}$	$e(d_d)$	$d_d \leq p$	s_7
s_5	$\{s_{a_{max}}, s_{v_{max}}, s_{d_{max}}, p, p_a, d_a\}$	$c?p_v$ $c?p_d$ $d!p_a$	$\lvert p_v\rvert \leq s_{v_{max}}$ $\lvert p_d\rvert \leq s_{d_{max}}$ $e \leq d_a$	s_8 s_9 s_0
s_6	$\{s_{a_{max}}, s_{v_{max}}, s_{d_{max}}, p, p_v, d_v\}$	$c?p_a$ $c?p_d$ $d!p_v$	$\lvert p_a\rvert \leq s_{a_{max}}$ $\lvert p_d\rvert \leq s_{d_{max}}$ $e \leq d_v$	s_{10} s_{11} s_0
s_7	$\{s_{a_{max}}, s_{v_{max}}, s_{d_{max}}, p, p_d, d_d\}$	$c?p_a$ $c?p_v$ $d!p_d$	$\lvert p_a\rvert \leq s_{a_{max}}$ $\lvert p_v\rvert \leq s_{v_{max}}$ $e \leq d_d$	s_{12} s_{13} s_0
s_8	$\{s_{a_{max}}, s_{v_{max}}, s_{d_{max}}, p, e, p_a, d_a, p_v\}$	$e(d_{av})$	$d_a + d_{av} \leq p$	s_{14}
s_9	$\{s_{a_{max}}, s_{v_{max}}, s_{d_{max}}, p, e, p_a, d_a, p_d\}$	$e(d_{ad})$	$d_a + d_{ad} \leq p$	s_{15}
s_{10}	$\{s_{a_{max}}, s_{v_{max}}, s_{d_{max}}, p, e, p_v, d_v, p_a\}$	$e(d_{va})$	$d_v + d_{va} \leq p$	s_{16}
s_{11}	$\{s_{a_{max}}, s_{v_{max}}, s_{d_{max}}, p, e, p_v, d_v, p_d\}$	$e(d_{vd})$	$d_v + d_{vd} \leq p$	s_{17}
s_{12}	$\{s_{a_{max}}, s_{v_{max}}, s_{d_{max}}, p, e, p_d, d_d, p_a\}$	$e(d_{da})$	$d_d + d_{da} \leq p$	s_{18}
s_{13}	$\{s_{a_{max}}, s_{v_{max}}, s_{d_{max}}, p, e, p_d, d_d, p_v\}$	$e(d_{dv})$	$d_d + d_{dv} \leq p$	s_{19}
s_{14}	$\{s_{a_{max}}, s_{v_{max}}, s_{d_{max}}, p, e, p_a, d_a, p_v, d_{av}\}$	$c?p_d$ $d!(p_a p_v)$	$\lvert p_d\rvert \leq s_{d_{max}}$ $e \leq d_a + d_{av}$	s_{20} s_0
s_{15} 以降は省略				

表 1: パケット多重化プロトコルの状態遷移表

$$
\begin{aligned}
& d_a < e \wedge |z| \leq s_{v_{max}} \wedge M_{88'}]] \\
= \ & \forall z[d_a < e \wedge |z| \leq s_{v_{max}} \Rightarrow \\
& [|z| \leq s_{v_{max}} \vee e \leq d_a]] \\
= \ & \forall z[true \Rightarrow [|z| \leq s_{v_{max}} \vee e \leq d_a]] \\
= \ & \forall z[|z| \leq s_{v_{max}} \vee e \leq d_a].
\end{aligned}
$$

これより,$mgb(s_5, s_5')$が偽, すなわち $|p_v| > s_{v_{max}} \wedge e > d_a$ である時 S と S' は等価でないといえる. また,S の遷移 $s_5 \rightarrow s_8$ の遷移条件のみを $d_a < e \wedge |p_v| \leq s_{v_{max}}$ に変更したシステムを S'' とする. S の状態 s_i に対応する S'' の状態を s_i'' として同様に状態対 (s_5, s_5'') について等価性を検証すると,

$$
\begin{aligned}
mgb(s_5, s_5'') &= mgb1(s_5, s_5'', \emptyset) \\
&= \forall z[|z| \leq s_{v_{max}} \Rightarrow [e \leq d_a \wedge M_{00''} \vee \\
&\quad d_a < e \wedge |z| \leq s_{v_{max}} \wedge M_{88''}]] \\
&= \forall z[e \leq d_a \Rightarrow [e \leq d_a \wedge M_{00''} \vee \\
&\quad d_a < e \wedge |z| \leq s_{v_{max}} \wedge M_{88''}]] \\
&= \forall z[d_a < e \wedge |z| \leq s_{v_{max}} \Rightarrow \\
&\quad [|z| \leq s_{v_{max}} \vee e \leq d_a]] \\
&= \forall z[e \leq d_a \Rightarrow [|z| \leq s_{v_{max}} \vee e \leq d_a]] \\
&= \forall z[true].
\end{aligned}
$$

となるので, S と S'' は等価であるといえる.

6 あとがき

本研究では [1] のデータ付オートマトンの双模倣等価性の記号的検証アルゴリズムと [2] の時間オートマトンの双模倣等価性の記号的検証のアルゴリズムを組み合わせて入出力データおよび時間値の両方を扱える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のアルゴリズムを提案した.

この方法では, 遷移条件を任意の方法で書くことができ, 入出力データおよび時間値の具体的な値に関わらないコストで双模倣等価性の検証を行なうことができる. また, 本手法が適用できるような実世界に沿った例を示した.

今後の課題としてはこのアルゴリズムを実装し,5章に示したようなモデルに対して機械的に検証結果を出すことである.

参考文献

[1] M. Hennessy and H. Lin, "Symbolic Bisimulations," *Theoretical Computer Science*, 138, pp. 353–389 (1995).

[2] A. Nakata, T. Higashino and K. Taniguchi, "Time-Action Alternating Model for Timed Processes and Its Symbolic Verification of Bisimulation Equivalence," 電子情報通信学会英論文誌 (A), Vol. E80-A, No. 2, pp. 400–406 (1997).

[3] U. Holmer, K. Larsen and Y. Wang, "Deciding Properties of Regular Timed Processes," *Proc. of 3rd Int. Conf. on Computer Aided Verification*, Lecture Notes in Computer Science, Vol. 575, pp. 443–453 (1991).

[4] L. Chen, "An Interleaving Model for Real-time Systems," *Proc. of 2nd Int. Symp. on Logical Foundations of Computer Science(LFCS'92)* Lecture Notes in Computer Science, Vol.620, pp. 81–92 (1992).

[5] R. Alur, C. Courcoubetis and T.A. Henzinger, "The Observational Power of Clocks," *Proc. of CONCUR'94*, Lecture Notes in Computer Science, Vol. 836, pp. 162–177 (1994).

[6] ITU-T, "低ビットレートマルチメディア通信用多重化プロトコル," ITU-T 勧告 H.223, TTC 標準 JT-H 221 (1996).

[7] 新田 直子, "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," 平成 9 年度大阪大学基礎工学部情報工学科卒業研究報告 (1998).