The University of Osaka
Institutional Knowledge Archive

Title	外部入力値のみを保持できる整数変数をもつFSMに対する記号モデル検査法
Author(s)	竹中, 崇; 岡野, 浩三; 東野, 輝夫 他
Citation	電子情報通信学会論文誌D. 2004, J87-D1(4), p. 462-470
Version Type	VoR
URL	https://hdl.handle.net/11094/27431
rights	Copyright © 2004 IEICE
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

論　文

外部入力値のみを保持できる整数変数をもつFSMに対する記号モデル検査法

竹中　崇[†*]　岡野　浩三[†]　東野　輝夫[†]　谷口　健一[†]

Symbolic Model Checking of Extended Finite State Machines with Linear Constraints over Integer Variables

Takashi TAKENAKA[†*], Kozo OKANO[†], Teruo HIGASHINO[†], and Kenichi TANIGUCHI[†]

あらまし　整数変数をもつ有限状態機械（FSM/int）に対する記号モデル検査法を提案する．入力値を保持する変数の値は次に同一遷移で新たな値が読み込まれるまで変更されない，などの制約を満たす，与えられたFSM/intに対し，整数変数をもつCTL式を満たすか否かをこの記号検査アルゴリズムは判定する．この記号モデル検査アルゴリズムを実装し，ブラックジャックディーラ回路とパケット多重化プロトコルに適用した．そして，100状態，10整数変数程度の規模のシステムであれば数秒程度（最悪時で数分程度）で検証できることが確かめられた．

キーワード　記号モデル検査，CTL，プレスブルガー文

1. まえがき

記号モデル検査法はハードウェア設計の分野においては有用な方法として定着している [1], [2]．記号モデル検査法の成功の理由の一つとしてBDD [3], [4] 等のコンパクトで効率的な符合化法を採用していることが挙げられる．しかしながら，BDDによる表現方法の欠点の一つとして，任意桁の整数変数を扱えないことが挙げられる．

一方，拡張有限状態機械や通常のプログラムのように，無限状態をもつシステムに対するモデル検査アルゴリズムも重要である [5]〜[7]．本論文では整数変数をもつFSMを対象とした記号モデル検査アルゴリズムを提案する．提案するアルゴリズムでは，遷移条件と途中の計算過程の両方を整数上の線形不等式の論理結合で表現することにより，無限状態をもつシステムに対する安全性や活性を評価できる．本論文で提案する手法では，各整数変数が初期パラメータ値か若しくは外部入力値のみを保持するようなクラスのEFSMを検証の対象としている．初期パラメータを保持する変数の集合と，外部入力を保持する変数の集合は共通集合をもたない．初期パラメータを保持する変数は動作開始からその値を変更しない．一方，外部入力を保持する変数は，外部入力があったときにその値を記憶し，次の新しい値が読み込まれるまで変更されない．

同様なクラスのEFSMを検証の対象とする手法としては文献 [6] がある．文献 [6] で対象とするクラスのEFSMでは，過去の入力値を記憶しておくことはできるが，現在の入力値と過去の入力値の比較を遷移の実行条件として使用することができない．また，新しい外部入力が一つでも与えられたときには，それまで与えられた過去の入力値をすべて忘れ，それ以降の遷移の実行条件で使用できない．これらの制限により，過去の入力値の系列に依存する動作記述を行うことができない．

これに対して，本論文で対象とするEFSMでは，過去の入力値を記憶し，この過去の入力値と現在の入力値との比較を遷移の実行条件として使用することができる．そのため，過去の外部入力系列に依存した動作記述を行うことができる．このクラスのEFSMを以降ではFSM/intと呼ぶこととする．FSM/intの各状態

† 大阪大学大学院情報科学研究科，豊中市
Graduate School of Information and Science Technology, Osaka University, Machikaneyama, Toyonaka-shi, 560–8531 Japan
* 現在，NECマルチメディア研究所

遷移の遷移条件は整数変数及び外部入力パラメータ上の不等式の論理結合で表される．記号モデル検査アルゴリズムのもう一つの入力は FSM/int で表現されたシステムに対する時相性質である．これらは，整数変数値で状態指定ができる CTL [2] 風の式（CTL/int）で表される．

先に述べたように，FSM/int では過去の入力系列に依存する動作記述を行うことができるが，このモデルに対するモデル検査アルゴリズムは一般には存在しない．そこで，本論文では過去の入力系列に依存する動作記述を行うことができるという記述能力は保持しつつも，モデル検査が可能であるような FSM/int のサブクラスを提案し，このサブクラスに対するモデル検査手法を提案する．このサブクラスは，有限機械上に存在する閉路に着目し，この閉路の実行回数がそれ以降のモデルの動作に影響を与えないための制限を与えることにより得られる．この制限により，モデル検査で考慮すべき有限機械の実行系列の数が有限に抑えられるため，モデル検査が可能となる．

提案手法の有用性を確認するために，記号モデル検査を行うプログラムを実装し，ブラックジャックディーラ回路 [8] とパケット多重化プロトコル H.223 [9] に適用した．その結果，100 状態，10 変数程度のシステムであれば数秒で記号モデル検査ができることが確認できた．本論文では以降，**2.** と，**3.** において，FSM/int と CTL/int 式を定義する．**4.** で，アルゴリズムを述べる．**5.** に，評価結果を記し，**6.** でまとめる．

2. モ デ ル

FSM/int $\mathcal{M} = (S, T, R, s_0, rc_0)$ は，状態の有限集合 S，遷移の有限集合 T，整数変数の有限集合 R，初期状態 $s_0 \in S$ と，初期パラメータ値指定条件 rc_0 からなる．R 中の変数は初期パラメータと（外部入力値の保持のみできる）データ変数に分かれる．初期パラメータの値は計算開始時に決定され，以降 $\mathcal{M}$ の計算終了時まで不変とする．遷移 $t \in T$ は 4 字組 $\langle s_i, C_t, I_t, s_j \rangle$ で指定される．ここで，s_i と s_j はそれぞれ遷移元と遷移先の状態名を表す．C_t は t の遷移条件であり，I_t は遷移 t でセットされる変数の集合である．遷移条件 C_t は整数変数上の不等式の論理結合であり，使用される変数は，初期パラメータ，データ変数である．C_t に現れる変数 r は C_t の判定時には値が定まっているものと仮定する(注1)．

$\mathcal{M}$ における全状態(注2)は状態名と（初期パラメータとデータ変数の）整数値ベクトル $\vec{v}$ の組で与えられる．

全状態 $q = \langle s_i, \vec{v} \rangle$ における遷移 $t = \langle s_i, C_t, I_t, s_j \rangle$ の実行可能性は以下のように定義される．

t において外部から入力される（すなわち r はデータ変数であり，I_t に含まれる）各データ変数 $r_1, r_2, \ldots, r_k \in I_t$ について，それぞれ，ある外部入力値 $v_1, v_2, \ldots, v_k$ が存在し，かつ，それ以外の変数においては全状態 q における値 $\vec{v}$ に対して，条件 C_t が成り立つこと．

実行可能な遷移 t が実行されると状態 s_j に遷移する．このとき全状態は状態 s_j 及び，整数ベクトル $\vec{v}[r_1, r_2, \ldots, r_k \backslash v_1, v_2, \ldots, v_k]$ の組となる．ここで $\vec{v}[r_1, r_2, \ldots, r_k \backslash v_1, v_2, \ldots, v_k]$ はベクトル $\vec{v}$ 中の変数 r_i の位置の値を v_i で置き換えたものとする．

$\mathcal{Q}_{\mathrm{init}}$ で初期全状態集合を表すこととする．$\mathcal{Q}_{\mathrm{init}}$ の各全状態は以下を満たす．初期状態が s_0 であり，初期パラメータの取り得る値は初期レジスタ値指定条件 rc_0 を満たす．FSM/int $\mathcal{M}$ は，$\mathcal{Q}_{\mathrm{init}}$ 中のいずれの初期全状態からでも遷移を開始できるものとする．

$\mathcal{M}$ は $\mathcal{Q}_{\mathrm{init}}$ を満たす任意の初期全状態から開始し実行可能な遷移を繰り返すことによって実行していく．

2.1 FSM/int に対するクラス制限

容易に分かるように，入力のない 2-カウンタオートマトンの動作を模倣させることができる外部入力列が存在するので一般の FSM/int に対する記号モデル検査は一般に決定不能であり，記号モデル検査のアルゴリズムを構築するには，モデルに対するサブクラスを指定する必要がある．その方法の一つとして，有意な履歴のパターンを有限に抑えるような制限を与えることが考えられる．本論文では，FSM/int の実行制御のループ構造に着目し，ループの実行回数がその後の動きに影響を与えないサブクラスを与える．まず，依存グラフの定義を行い，この依存グラフを用いて閉路に関する制約を課すことによりクラス制限をする．

与えられた FSM/int $\mathcal{M}$ と，データ変数 r に対し，以下を満たす遷移の系列 $\sigma = s_i \overset{t_{k_0}}{\rightarrow} s_{i'} \overset{t_{k_1}}{\rightarrow} \cdots \overset{t_{k_{n-1}}}{\rightarrow} s_n$ を考える．

（1） σ は FSM/int $\mathcal{M}$ の有限状態部を単純に有限状態機械とみなして得られる，任意の状態 s_i から始まる遷移の系列である．

(注1)：C_t と I_i は共通のデータ変数を含んでもよい．
(注2)：状態名及び，各レジスタ値が異なる組合せをすべて区別した状態のそれぞれを “全状態” (total state) と呼ぶこととする．

（2） 遷移 t_{k_0} においてデータ変数 r に入力値がセットされる．

（3） 変数 r の値は σ において遷移 t_{k_0} 以外ではセットされない．

（4） 遷移 $t_{k_{n-1}}$ において変数 r の値が遷移条件で参照される．

このような場合，データ変数 r の値が遷移系列 σ において活性であると呼ぶ．また，遷移系列 σ を r に対する活性遷移系列と呼ぶ．$\mathcal{M}$ 上の（状態を挟んで入射，出射する）二つの遷移 t_i と t_{i+1} に対し，もし，両方の遷移がある変数 r の活性遷移系列に属するならば，すなわち，それぞれの活性な変数集合に共通のデータ変数をもつならば，t_{i+1} は t_i に依存するという．$\mathcal{M}$ の依存グラフ $\mathcal{G}_{\mathcal{M}}$ は，t_j が t_i に依存するとき，かつそのときに限り，$t_j \rightarrow t_i$ の有向枝をもつと定義する．依存グラフ $\mathcal{G}_{\mathcal{M}}$ 中の閉路中の状態でかつ二つ以上の遷移が入る状態を合流状態 (*confluent state*) と呼ぶ．閉路が正則閉路 (*regular cycle*) と呼ばれるのは，その閉路中の任意の合流状態 s について，s を含む次の活性遷移系列 σ が存在しない場合である．

- 活性遷移系列 σ に関する変数 r に代入するような遷移がその閉路中に存在する．

本論文では以下の制約を FSM/int におく．

- 依存グラフ $\mathcal{G}_{\mathcal{M}}$ に閉路が存在すれば，それは正則閉路でなくてはいけない．

正則閉路中の遷移が何度繰り返し実行されても，閉路外でセットされた変数で活性な変数の値が変更されることはない．

この制限を満たすクラスを r-FSM/int と呼ぶ．

［例 1］（FSM/int の例） 図 1 は FSM/int の例である(注3)．初期パラメータ集合は $\{p, q\}$ であり，データ変数集合は $\{i_0, j_0, i_2, j_2\}$ である．初期状態は s_1 である．図 1 中の各遷移 t_i において，I_t と c_t はラベルとして，集合及び論理式として表されている．省略時はそれぞれ空集合，真を意味する．

$\mathcal{M}$ の動作は例えば以下のように進行していく．初期状態で初期パラメータ値 p と q がそれぞれ 0, 10 とする．初期状態 s_1 において実行可能な遷移は t_0 である．この遷移は常に実行可能であり，実行により，データ変数 i_0 と j_0 に外部から値が代入される．この値を仮に 4 と 5 とする．このとき全状態は $(0, 10, 4, 5, \star, \star) \times s_2$ となる（i_2 と j_2 はこの時点では不定値）．この状態において実行可能な遷移は条件式 $p + i_0 < j_0 < q$ が真となる遷移 t_1 である．t_1 の実行により状態 s_3 に

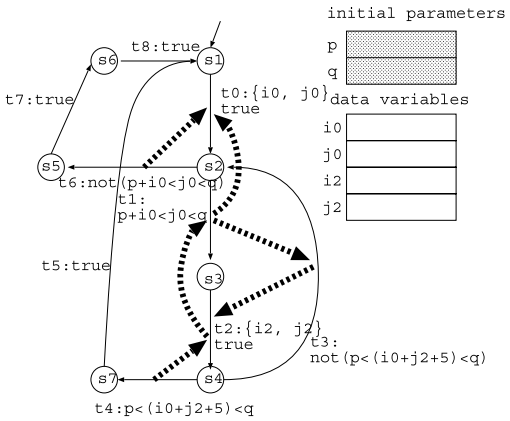

図 1 FSM/int の説明のための例
Fig. 1 An example: FSM/int.

遷移し，更に遷移 t_2 が実行され状態 t_4 に移る．この際，データ変数 i_2 と j_2 に外部から値が代入される．この値を仮に 6 と 7 とすると，この時点での全状態は $(0, 10, 4, 5, 6, 7) \times s_4$ となる．この全状態では遷移 t_3 が実行可能となる．

依存グラフ $\mathcal{G}_{\mathcal{M}}$ を図 1 中に破線で表している．遷移 t_1 から遷移 t_2 への依存辺が存在する理由は以下のとおりである．変数 i_0 の活性遷移系列は $s_1 \xrightarrow{t_0} s_2 \xrightarrow{t_1} s_3 \xrightarrow{t_2} s_4 \xrightarrow{t_4} s_7$ であり，遷移 t_1 と t_2 が変数 i_0 の活性遷移系列に属するためである．

これに対して，遷移 t_5 から t_4 への依存辺は存在しない．これは，遷移 t_4 が，変数 i_0, j_2 の活性遷移系列に属するのに対して，遷移 t_5 がどの変数の活性遷移系列にも属していないためである．

図 1 を見て分かるように $\mathcal{G}_{\mathcal{M}}$ は遷移 t_1, t_2, t_3 からなる閉路を一つもつ．この閉路中の合流状態は s_2 一つである．s_2 を含む活性遷移系列に関するどのデータ変数 i_0 と j_0 も閉路中で値が変化しないので，この閉路は正則閉路である．よって，図 1 の $\mathcal{M}$ は r-FSM/int に属する．

3. 検証性質

本論文では CTL を拡張し，各状態における整数変数値のベクトルを状態指定の一部として指定できるように拡張している．

状態式 を $[\alpha_{s_i}]@s_i$ と表記する．ここで，s_i は状態名であり α_{s_i} は（対象とする FSM/int の）整数変数上の不等式の論理結合である．直感的には，状態式は，

(注3)：FSM/int の諸概念の説明のための図であり，この FSM 自体は意味のある動作を行うわけではない．

状態が s_i で整数変数値が α_{s_i} を満たす全状態集合を表す．

［定義 1］（状態式の構文） 状態式 SF の文法の定義を以下のように与える：

$$\begin{aligned}
&\mathrm{SF} ::= [\mathrm{sp}(s)]@s \\
&\mathrm{sp}(s) ::= g(s) \mid (\mathrm{sp}(s)) \mid \mathrm{sp}(s) \wedge \mathrm{sp}(s) \\
&\qquad\qquad \mid \neg\mathrm{sp}(s) \mid \mathrm{true} \mid \mathrm{false} \\
&g(s) ::= t(s) < t(s) \\
&t(s) ::= (t(s)) \mid t(s) + t(s) \mid -t(s) \mid x \mid 1 \mid 2 \mid \ldots,
\end{aligned}$$

ここで x は整数変数である．また，$\mathrm{sp}(s)$ をそのような整数変数上不等式の論理結合とする．

［定義 2］（状態式の意味定義） 状態式 $[\mathrm{sp}(s)]@s$ は全状態の集合を意味し，その集合は以下のように与える：

$$[\mathrm{sp}(s)]@s = \{(s, \vec{v}) \mid \mathrm{sp}(s)\},$$

ここで $\vec{v}$ は整数変数 r_i の値ベクトルである．

CTL/int は CTL [2] を直接拡張したものである．

［定義 3］（CTL/int の構文） CTL/int 式を以下のように定義する：

$$\begin{aligned}
f ::= \ &\mathrm{SF} \mid \neg f \mid f_1 \cap f_2 \mid f_1 \cup f_2 \mid \\
&\exists\bigcirc f \mid \exists\Box f \mid f_1 \ \exists\mathcal{U} \ f_2.
\end{aligned}$$

CTL/int の時相演算子の意味を通常どおり FSM/int $\mathcal{M}$ のパスや状態を用いて定義する．パス $(q_0, q_1, \ldots)$ は全状態の系列である．パス中の連続する二つの全状態 q_i と q_{i+1}（それぞれに対応する状態を s_i と s_{i+1} とする）に対し，遷移 $\langle s_i, C_{i,i+1}, I_{i,i+1}, s_{i+1}\rangle \in T$ が存在しなければいけない．ここで，もし，$I_{i,i+1}$ が空であれば，$C_{i,i+1}$ は全状態 q_i で成立する必要がある．それ以外のときはある外部入力値 $i_1, i_2, \ldots, i_k$ $(\in I_{i,i+1})$ があって，それらが $C_{i,i+1}$ を満たさなければいけない．

［定義 4］（CTL/int の意味定義） 与えられた FSM/int $\mathcal{M}$ と CTL/int 式 f に対して，CTL/int の意味を以下のように与える．ここで，q と q_i を $\mathcal{M}$ の全状態とする：

$q \models \mathrm{SF}$ iff $q \in \mathrm{SF}$

$q \models \neg f$ iff $q \not\models f$

$q \models f \cap g$ iff $q \models f$ and $q \models g$

$q \models f \cup g$ iff $q \models f$ or $q \models g$

$q_0 \models \exists\bigcirc f$ iff あるパス $(q_0, q_1, \ldots)$ 上で，$q_1 \models f$

$q_0 \models \exists\Box\ f$ iff あるパス $(q_0, q_1, \ldots)$ 中のすべての全状態 q_j において，$q_j \models f$

$q_0 \models f \ \exists\mathcal{U}\ g$ iff あるパス $(q_0, q_1, \ldots)$ に対して “i” が存在して $q_i \models g$ かつ すべての $j(< i)$ に対して，$q_j \models f$

$\mathcal{M}$ が f を満たすこと，すなわち，$\mathcal{M} \models f$ を $\mathcal{Q}_{\mathrm{init}} \subseteq \mathcal{Q}_f$ で定義する．ここで $\mathcal{Q}_f (= \{q \mid q \models f\})$ は CTL/int 式 f を満たす全状態集合である．

$\forall\bigcirc$ や，$\exists\Diamond, \forall\Box, \forall\Diamond, \forall\mathcal{U}$ のような時相演算子はこれまで定義した演算子を用いて通常どおり定義する．

［例 2］（CTL 式） 以下の式を考える．

$$\exists\Diamond[(0 < j_0 + 3) \wedge (i_0 < i_2) \wedge (i_2 < p + j_0)]@s_7.$$

図 1 中の FSM/int $\mathcal{M}$ はこの性質を満たす．すなわち $\mathcal{M}$ は，ある外部入力データによって，いつかは次の全状態に到着する．状態が s_7 で各整数変数は $(0 < j_0 + 3) \wedge (i_0 < i_2) \wedge (i_2 < p + j_0)$ を満たす．この説明は **4.3** を参照のこと．

4. 記号モデル検査アルゴリズム

ここで記号検査アルゴリズムを与える．$\mathcal{M}$ が f を満たすことを調べるために，CTL/int 式 f の部分式を満たす全状態を f の構成に従って帰納的に調べることにより記号検査を行う．f を満たす全状態を表現する「データ構造」を構成した後はその全状態が $\mathcal{M}$ の初期状態を含むかどうかをチェックすればよい．このデータ構造として，我々は単純にプレスブルガー文を用いる．以降では，与えられた CTL/int に対し，それを表現するプレスブルガー文をどう構成するかを説明する．

4.1 式 $\exists\bigcirc f$ の扱い

CTL/int 部分式 f を満たす全状態集合に相当するプレスブルガー文 $\mathcal{Q}_f$ が既に得られていると仮定する(注4)．部分式 $\exists\bigcirc f$ を満たす全状態集合に相当するプレスブルガー文 $\mathcal{Q}_{\exists\bigcirc f}$ の構成方法は以下のとおりである．まず，$\mathcal{Q}_f$ が $\bigcup_{s_j \in S}[\alpha_{s_j}]@s_j$ に等しいと仮定する．遷移 $t_{ij} = \langle s_i, C_{ij}, I_{ij}, s_j\rangle$ を考える．我々のクラス制限のもとでは，$r \notin I_{ij}$ の変数値は t_{ij} が実行されても変化しないことが保証される．よって $\mathcal{Q}_{\exists\bigcirc f}$ を以下のように（プレスブルガー文として）導出することができる：

(注4)：以降混乱のない限りプレスブルガー文とそれに対応する全状態集合に同じ記号を用いる．

$$\mathcal{Q}_{\exists\bigcirc f} = \bigcup_{s_j \in S} \bigcup_{\substack{s_i \in S \\ \langle s_i, C_{ij}, I_{ij}, s_j \rangle \in T}} \left([\exists i \in I_{ij}.(C_{ij} \wedge \alpha_{s_j})]@s_i \right),$$

ここで $\exists i \in I_{ij}.(C_{ij} \wedge \alpha_{s_j})$ は

$$\begin{array}{ll} (C_{ij} \wedge \alpha_{s_j}) & \text{if} \quad I_{ij} = \emptyset \\ \exists i_1, i_2, \ldots, i_k.(C_{ij} \wedge \alpha_{s_j}) & \text{if} \quad I_{ij} = \{i_1, i_2, \ldots i_k\}. \end{array}$$

例えば，図 1 の $\mathcal{M}$ に対して，$\mathcal{Q}_f$ を $[(p < i_0 < q + j_0)]@s_2$ とする．すると，$\mathcal{Q}_{\exists\bigcirc f} = [\alpha_{s_1}]@s_1 \cup [\alpha_{s_4}]@s_4$ が導かれる．

4.2 式 $f \exists\mathcal{U} g$ と $\exists\Box f$ の扱い

同様に，$\mathcal{Q}_{f \exists\mathcal{U} g}$ と $\mathcal{Q}_{\exists\Box f}$ の全状態集合を求める方法について述べる．

4.2.1 式 $f \exists\mathcal{U} g$ に対する基本アルゴリズム

部分式 f と g に対応する全状態集合 $\mathcal{Q}_f$ と $\mathcal{Q}_g$ がそれぞれ得られているとする．

$\mathcal{Q}_{f \exists\mathcal{U} g}$ は以下で定義される．

$$Q_0 = \mathcal{Q}_g, \quad Q_{n+1} = Q_n \cup (\mathcal{Q}_f \cap \exists\bigcirc Q_n),$$
$$\mathcal{Q}_{f \exists\mathcal{U} g} = Q_\infty.$$

Q'_n を以下のように定義する．$Q'_n = Q_n - Q_{n-1}$．Q'_n を順次求めていくことにより目的の式を得る．長さが n のパスを n-path と呼ぶ．Q'_n は $\mathcal{Q}_g$ 中の各全状態から各 n-path に沿って遷移を後ろ向きにたどり，途中の状態が $\mathcal{Q}_f$ に含まれるような状態を求めることによって得られる．

$\mathcal{Q}_g$ を $\bigcup[\alpha_{s_j}]@s_j$ と表現する．上記の後ろ向き探索は，各部分集合 $[\alpha_{s_j}]@s_j$ に対して，状態 s_j に至る遷移系列の各々に対して後ろ向き探索を行うことに相当する．例えば，ある状態 s_j へ n 個の遷移で到達可能な遷移系列 $\sigma_{ij} = s_i \xrightarrow{t_1} \cdots \xrightarrow{t_{n-1}} s_l \xrightarrow{t_n} s_j$ で到達可能な全状態の集合は $[\beta_{\sigma_{ij}}]@s_i$ となる．ここで，$\beta_{\sigma_{ij}}$ は各遷移の遷移条件及び途中の状態 s_m において（経由するすべての全状態が $\mathcal{Q}_f$ に含まれるために）満たすべき条件 γ_{s_m} と存在限定子が順にネストした条件

$$\begin{aligned} \beta_{\sigma_{ij}} = (\gamma_{s_i} \wedge [\exists i \in I_n.(C_n \wedge \cdots \\ \cdots \wedge (\gamma_{s_l} \wedge [\exists i \in I_1.(C_1 \wedge \alpha_{s_j})]) \cdots)]) \end{aligned}$$

である．このように，（$\mathcal{Q}_g$ 中の）状態 s_j に至る各 n-path を逐次的に後ろ向きにたどることにより Q'_n を得ることができる．

したがって，$\mathcal{Q}_{f \exists\mathcal{U} g}$ を得るためには，このような後ろ向き探索を新たな全状態が得られなくなるまで続ければよい．

後ろ向き探索を終えるには，

（1） 後ろ向き探索の最後の遷移系列が実行不能，または，

（2） 新たな状態がこれ以上増えない，

という条件でよい．条件 (1) は $\beta_{\sigma_{ij}}$ が充足不能であればよい．条件 (2) については新たに得られた集合 $[\beta_{\sigma_{ij}}]@s_i$ が集合 $[\beta_{\sigma_{i'j}}]@s_{i'}$ に含まれることを判定すればよい．ここで，$[\beta_{\sigma_{i'j}}]@s_{i'}$ は次の 2 条件を満たす全状態の集合である．

(2-1) 状態 s_i が $s_{i'}$ と同じ，かつ，

(2-2) $\beta_{\sigma_{ij}} \rightarrow \beta_{\sigma_{i'j}}$ が成り立つ．

なお，$\sigma_{i'j}$ は $\beta_{\sigma_{ij}}$ の接頭系列であり $\sigma_{i'j} = s_{i'} \xrightarrow{t_{k+1}} \cdots \xrightarrow{t_n} s_j$ で定義される．

まとめると条件 (2) すなわち，条件 $[\beta_{\sigma_{ij}}]@s_i \subseteq [\beta_{\sigma_{i'j}}]@s_{i'}$ は条件 (2-1) と (2-2) に置き換えることができる．

条件 (2-1) は後ろ向き探索中に状態 s_i に 2 度訪れたことを意味する．条件 (2-1) と (2-2) が成り立てば，後ろ向き探索で得られた全状態は σ_{ij} を後ろ向きにたどって得られた全状態集合に既に含まれる．したがってこれ以上探索する必要はない．

なお，2 度訪れた s_i に対して $\beta_{\sigma_{ij}} \rightarrow \beta_{\sigma_{i'j}}$ が常に成り立つわけではない．

上で述べたように $\beta_{\sigma_{ij}}$ と $\beta_{\sigma_{i'j}}$ は条件や限定子が交互にネストしている．したがって，$\beta_{\sigma_{ij}}$ は $\beta_{\sigma_{i'j}}$ を用いて以下のように表記される：

$$\begin{aligned} \beta_{\sigma_{ij}} = (\gamma_{s_i} \wedge [\exists i \in I_n.(C_n \wedge \cdots \\ \wedge (\gamma_{s_{k+1}} \wedge [\exists i \in I_{k+1}.(C_{k+1} \wedge \beta_{\sigma_{i'j}})]) \cdots)]). \end{aligned}$$

もし，$\beta_{\sigma_{ij}}$ で束縛される自由変数が $\beta_{\sigma_{i'j}}$ にあれば，$\beta_{\sigma_{ij}} \rightarrow \beta_{\sigma_{i'j}}$ が成り立たないこともある．

例えば図 2 を考える．例 2 で与えた式に対して，もし $\beta_{\sigma'}$ を遷移系列 $\sigma' = s_4 \xrightarrow{t_3} s_2 \xrightarrow{t_1} s_3 \xrightarrow{t_2} s_4 \xrightarrow{t_4} s_7$ に基づいて計算すると，$\beta_{\sigma'}$ は $\exists i_2, j_2.(p < i_0 + j_2 + 5 < q \wedge 0 < j_0 + 3 \wedge i_0 < i_2 < p + j_0) \wedge (p + i_0 < j_0 < q) \wedge \neg(p < i_0 + j_2 + 5 < q)$ になる．これは自由変数 j_2 をもち，この変数は状態 s_2 まで計算を続けると限定子に束縛されることとなる．それゆえ，少なくとも状態 s_4 では計算を停めることはできない．したがって，そのつど原則としてチェックが必要となる．図 3 に $\beta_{\sigma'}$ の計算過程を表している．通常の限定子消去の方法により，図 3 の最後の式を得ることができる．

ここで注意すべきこととしては，r-FSM/int であっ

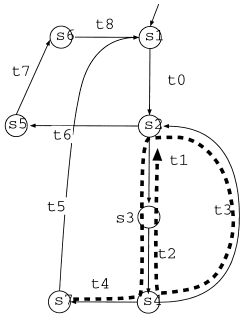

図 2　σ_{ij} の後ろ向き探索中に状態 s_2, s_4 に 2 回訪問する例
Fig. 2　The case where we arrive at states s_2 and s_4 twice during backward trace of σ_{ij}.

$$
\begin{aligned}
&[(0 < j_0 + 3) \wedge (i_0 < i_2 < p + j_0)]@s_7 \\
&\qquad \uparrow t_4 \\
&[(0 < j_0 + 3) \wedge (i_0 < i_2 < p + j_0) \wedge (p < (i_0 + j_2 + 5) < q)]@s_4 \\
&\qquad \uparrow t_2 \\
&\exists i_2, j_2 [(0 < j_0 + 3) \wedge (i_0 < i_2 < p + j_0) \wedge (p < (i_0 + j_2 + 5) < q)]@s_3 \\
&= \exists j_2 [(0 < j_0 + 3) \wedge (i_0 + 1 < p + j_0) \wedge (p - i_0 - 5 < j_2 < q - i_0 - 5)]@s_3 \\
&= [(0 < j_0 + 3) \wedge (i_0 + 1 < p + j_0) \wedge (p + 1 < q)]@s_3 \\
&\qquad \uparrow t_1 \\
&[(0 < j_0 + 3) \wedge (i_0 + 1 < p + j_0) \wedge (p + 1 < q) \wedge (p + i_0 < j_0 < q)]@s_2 \\
&\qquad \uparrow t_3 \\
&[(0 < j_0 + 3) \wedge (i_0 + 1 < p + j_0) \wedge (p + 1 < q) \wedge (p + i_0 < j_0 < q) \\
&\quad \wedge \neg(p < i_0 + j_2 + 5 < q)]@s_4
\end{aligned}
$$

図 3　$\beta_{\sigma'}$ の計算
Fig. 3　Calculation of $\beta_{\sigma'}$.

ても，なお，2 度訪れた s_i に対して $\beta_{\sigma_{ij}} \rightarrow \beta_{\sigma_{i'j}}$ が常に成り立つわけではないこと，すなわち，いつでも探索を打ち切れるわけではないことである．

4.2.2　式 $\exists\Box f$ に対する基本アルゴリズム

CTL 式 $\exists\Box f$ を扱うためのアルゴリズムについては簡単に述べる．$\mathcal{Q}_{\exists\Box f}$ は $\mathcal{Q}_f$ 中の各状態から遷移を前向きにたどることにより計算する．ここで

$$\mathcal{Q}_f = \bigcup_{s_i \in S} [\delta_{s_i}]@s_i$$

である．

全状態の集合 $\mathcal{Q}_f$ 中のある全状態の集合 $[\delta_{s_i}]@s_i$ について，状態 s_i からある遷移系列 $\sigma_{ij} = s_i \xrightarrow{t_1} \cdots \xrightarrow{t_n} s_j$ に沿って順に探索することにより，この遷移系列に沿った実行系列中に経由する全状態がすべて $\mathcal{Q}_f$ に含まれる全状態の集合 $[\eta_{\sigma_{ij}}]@s_i$ （$[\delta_{s_i}]@s_i$ の部分集合である）が得られる．ただし，$\eta_{\sigma_{ij}}$ は経由する遷移の遷移条件と存在限定子が順にネストした条件

$$
\begin{aligned}
\eta_{\sigma_{ij}} \;=\; & \delta_{s_i} \wedge [\exists i \in I_1.(C_1 \wedge \delta_1 \wedge \cdots \\
& \cdots \wedge [\exists i \in I_n.(C_n \wedge \delta_{s_j})] \cdots)]
\end{aligned}
$$

である．全状態の集合 $\mathcal{Q}_{\exists\Box f}$ は，このような各状態からのすべての探索を全状態の集合がそれ以上小さくならなくなるまで続けてそれぞれ得られた全状態の集合和となる（$\mathcal{Q}_{f\,\exists\mathcal{U}\,g}$ とは異なり，探索途中で得られた抽象状態を逐次追加することはしない）．

探索を更に続ける必要があるか否かの判定は，以下の判定で行う．

（1）　その探索で経由した遷移系列が実行不能か否か，及び，

（2）　今後探索を続けて新たに求められる全状態がそれまでの探索で既に求められている全状態の集合より小さくなる可能性があるか否か．

（1）は，条件 $\eta_{\sigma_{ij}}$ が充足可能か否かの判定で行うことができる．（2）は，探索を行った遷移系列 σ_{ij} に対してその前半の部分系列 $\sigma_{ij} = s_i \xrightarrow{t_1} \cdots \xrightarrow{t_k} s_{j'}$ を考えたときに，

(2-1)　s_j と $s_{j'}$ が同じ状態であり（これは，遷移系列 σ_{ij} に沿った探索中に同じ状態 s_j を 2 度訪問していることに相当する），かつ，

(2-2)　$\eta_{\sigma_{ij}}$ が $\eta_{\sigma_{ij'}}$ と同じ条件であるか，すなわち $\eta_{\sigma_{ij}} \leftrightarrow \eta_{\sigma_{ij'}}$ がすべての自由変数の任意の解釈のもとで恒真であるか，

の 2 条件の判定で行うことができる．ここで，遷移系列 σ_{ij} 実行後に更に遷移系列 $s_{j'} \xrightarrow{t_{k+1}} \cdots \xrightarrow{t_n} s_j$ からなる閉路を無限に実行する遷移系列 σ_{ij+} を考える．上記の 2 条件が成り立つとき，このような遷移系列 σ_{ij+} をいくら続けて探索を行っても，得られる状態集合は $[\eta_{\sigma_{ij}}]@s_i$ より小さくならず，それ以上の探索を行わない．このとき，$[\eta_{\sigma_{ij}}]@s_i$ 中の任意の全状態から，無限長の遷移系列 σ_{ij+} の実行したとき途中で経由する全状態はすべて $\mathcal{Q}_f$ に含まれるので，$[\eta_{\sigma_{ij}}]@s_i \in \mathcal{Q}_{\exists\Box f}$ となる．

充足可能性や同値性の判定をプレスブルガー文の真偽判定問題に帰着し行うこと，探索中に同じ状態に 2 度訪問したからといって，常にそれ以上のトレースを行わなくてもよいとは限らないことは $\mathcal{Q}_{f\,\exists\mathcal{U}\,g}$ を求めるときと同様である．

4.2.3　アルゴリズムの停止性

この項では，正しく $\mathcal{Q}_{f\,\exists\mathcal{U}\,g}$ を有限ステップで得られることの略証を与える．

最初に，後ろ向き探索が終了することを提案モデルの構成に従って示す．

r-FSM/int $\mathcal{M}$ では，σ_{ij} をたどる際，クラス制限によって，s_i を 2 回訪問し，次の二つの条件のうち一

つでも成り立てば．$[\beta_{\sigma_{i'j}}]@s_i \supseteq [\beta_{\sigma_{ij}}]@s_i$ が成り立ち，後ろ向き探索を終了できる．

（1） 状態 s_i への入射遷移 t_{k+1} が出次遷移 t_k に依存しない場合（この状態 s_i を依存性中断状態と呼ぶ）．例えば，図 4 の探索 (a) では状態 s_1 は依存性中断状態である（図 5(a)）．

（2） 状態 s_i が正則閉路の合流状態である場合．例えば，図 4 の探索 (b) では状態 s_2 はそのような合流状態である（図 5(b)）．

この条件を停止条件と呼ぶ．停止条件のいずれかが成り立てば次の等式が成り立つ．

$$\begin{aligned}\beta_{\sigma_{ij}} &= (\gamma_{s_i} \wedge [\exists i \in I_n.(C_n \wedge \cdots \\ &\quad \wedge(\gamma_{s_{k+1}} \wedge [\exists i \in I_{k+1}.(C_{k+1} \wedge \beta_{\sigma_{i'j}})]) \cdots)]) \\ &= (\gamma_{s_i} \wedge [\exists i \in I_n.(C_n \wedge \cdots \\ &\quad \wedge(\gamma_{s_{k+1}} \wedge [\exists i \in I_{k+1}.(C_{k+1})] \cdots)]) \wedge \beta_{\sigma_{i'j}},\end{aligned}$$

これは，（依存関係や正則な閉路の定義より）$\beta_{\sigma_{i'j}}$

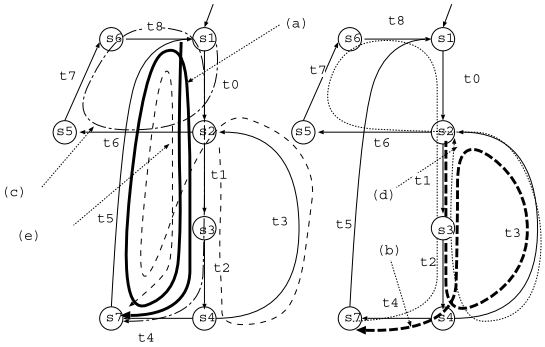

図 4 例 1 に対する記号モデル検査の探索系列
Fig. 4 Sequences traced for model checking of $\mathcal{M}$ in Example 1.

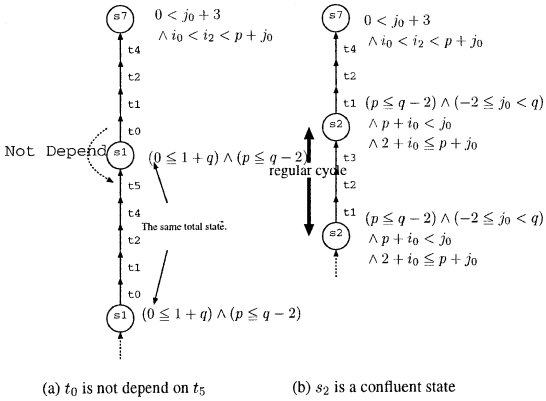

図 5 σ_{ij} の後ろ向き探索
Fig. 5 Backward trace of σ_{ij}.

の自由変数の値を前半の部分系列でセットしない（$I_n, \ldots, I_{k+1}$ に出現しない）ので成り立つ．

次に，上記の性質を利用して，$\mathcal{Q}_{f\,\exists\mathcal{U}\,g}$ を求める計算では無限長の後ろ向き探索を考えなくてもよいことを示す．r-FSM/int では，依存グラフに正則な閉路以外の閉路が存在しないので，依存遷移グラフ上で正則な閉路となる遷移の系列や，正則な閉路を含まないような順に依存し合う遷移系列はともにその長さが有限となる．したがって，抽象状態の集合 $[\alpha_{s_j}]@s_j$ に対して状態 s_j からの後ろ向き探索を考えるとき，次の理由より無限長の探索は存在しない．

仮に無限長の後ろ向き探索が存在するとする．正則な閉路となる遷移系列や，正則な閉路を含まないような順に依存し合う遷移系列はともにその長さが有限であるので，この無限長の後ろ向き探索中には前後の遷移が依存しないような状態や正則閉路中の合流状態であるような状態が無限に存在する．FSM/int には，依存性中断状態の組合せは有限個しか存在しない．また，正則な閉路の開始状態も有限個しか存在しない．したがって，無限の後ろ向き探索中には依存性中断状態や正則閉路中の合流状態であるような状態のうちのどれかは 2 度以上出現する．このような状態を 2 度訪問したときは，その探索を終了することができるため，後ろ向き探索は有限長となり，仮定に反する．直感的に以上のことは，同じ遷移系列（閉路）を 2 度以上繰り返して逆に探索することがないことを表している．

したがって，r-FSM/int では全状態の集合 $\mathcal{Q}_g$ 中の各全状態の集合 $[\alpha_{s_j}]@s_j$ に対して，状態 s_j からの後ろ向き探索を考えるとき，すべての後ろ向き探索が有限長になるため部分式 $f\,\exists\mathcal{U}\,g$ を満たす状態集合を求める計算が有限時間で終了する．

同様の議論により $\mathcal{Q}_{\exists\Box\,f}$ が有限ステップで得られることが議論できる．

4.3 記号モデル検査の例

例 1 の FSM/int$\mathcal{M}$ のもとで例 2 の CTL/int 式 f が成り立つことを調べるモデル検査の例を示す．

まず，$\exists\Diamond[(0 < j_0{+}3) \wedge (i_0 < i_2) \wedge (i_2 < p{+}j_0)]@s_7$ を満たす全状態の集合を求めることを考える．考慮すべき状態 s_7 からの後ろ向き探索は，**4.2.1** の条件より，図 4 の遷移系列 (a)，(b)，(c)，(d)，(e) を経由するような後ろ向き探索である．

例えば，状態 s_2 では，図 2 について，状態 s_2 から s_7 まで，後半部分のみを経由する条件，遷移系列 (b) のすべてを経由する条件ともに，$\beta_0 = (p \leq$

$q-2) \wedge (j_0 < q) \wedge (p+i_0 < j_0) \wedge (2+i_0 \leq p+j_0) \wedge (0 \leq 3+i_0+2j_0) \wedge (7+2i_0+j_0 \leq q)$ であり，同じ条件になっていることが分かる．すなわち，ここで探索を打ち切ってもよいということが分かる．これらすべての遷移系列について後ろ向き探索を行うことにより，初期状態 s_1 で満たすべき条件 $(2-3q \leq q) \wedge (p \leq q-2)$ が得られる．

初期パラメータレジスタ値がこの条件を満たすとき，f を満たす抽象状態の集合 $\mathcal{Q}_f$ に初期抽象状態の集合が含まれ，$\mathcal{M} \models f$ が成り立つ．

5. 検証実験

例題として，TPCD94 ベンチマークセット [8] より Black-Jack Dealer（以下 bjd と呼ぶ）を一部修正した回路を取り上げる．bjd は，カードゲームのスコアの判定を行う回路である．カードの値（1 から 13 の整数値）の系列が入力されたときにそのスコアを計算し，16 以下 21 以上の有効なスコアとなっているか，あるいはスコアが 21 以上の無効となっているかをそれぞれ `Stand`，`Broke` の真偽値として出力する．5 枚のカードの値が入力されるか，あるいは `Stand` または `Broke` が出力されたときスコアの値を初期化し最初から繰り返す．スコア計算では 1 のカードは “1” または “11” のいずれかの値として使用できる．どちらとして使用するかはプレーヤーにとって有利となるように自動的に判別する．このような回路を r-FSM/int の仕様として記述した（以下 bjd1 と呼ぶ）．有限制御部の状態数は 88，入力レジスタの数は 5 個であった．また，初期パラメータレジスタはもたない．このような回路に対して，以下のような性質を検証した．

（性質 1）　初期状態より到達可能な任意の状態に対し，いつか初期状態に戻るような実行系列が存在する．この性質は $\forall\Box(\exists\Diamond([\mathrm{true}]@s_0))$ のような CTL/int 式で表される．ただし，s_0 は初期状態を表す．

（性質 2）　ゲーム終了時にスコアが 21 より大きく，かつ，`Broke` が偽である状態に到達することがない．この性質は以下のような CTL/int 式で表される．

$$\neg\exists\Diamond(\bigcup_{s_i \in S_{\neg\mathrm{broke}}} [\mathrm{score}_{s_i} > 21]@s_i)$$

ただし，$S_{\neg\mathrm{broke}}$ はゲーム終了時に `broke` が偽である有限状態の集合，score_{s_i} は状態 s_i でのスコアの値を表す．

（性質 3）　ゲーム終了時にスコアが 16 より小さく，かつ，`Stand` が真である状態に到達するすることがない．

（性質 4）　カードが 5 枚入力され，かつ，それらがすべて同じであることはない．

（性質 5）　カードが 5 枚入力されたときのスコアが 11 より小さくなることがない．

また，カードの最大値，及び有効なスコアの範囲を初期パラメータとして扱い，（$\mathcal{Q}_{\mathrm{init}}$ 中の）任意の初期パラメータ値（整数値）に対して定義されるすべての回路が上記性質を満たすことの検証も行った．カードの最大値のみを初期パラメータで指定した回路を bjd2 とし，それに加え有効なスコアの範囲を初期パラメータで指定した回路を bjd3 としている．これらのシステムは有限状態機械モデルではうまく扱えない．

検証実験には，Perl 及び yacc を用いて試作した検証支援系を用いた（記述は全体で 2000 行程度）．プレスブルガー文真偽判定手続きには米メリーランド大学で開発された Omega [10] を利用した．この検証支援系では自明な判定を省くなどの工夫を行っている．検証実験を行った結果を表 1 に示す．

有限状態機械モデルではうまく扱うことのできない他の例題として，映像，音声，データの三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して送出するパケット多重化方式 H.223 [9] の仕様例に対しても検証実験を行った．これはパケット到着ごとにパケットの種類，そのサイズとパケット化遅延が入力されたとき，これらのパケットを周期的に多重化して送出するような例である．パケットを送出する周期や，送出パケットの最大サイズ等が初期パラメータとして与えられる．データレジスタでは転送タイムアウト値や各種パケット内のパラメータを保持する．有限状態部ではパケット内のパラメータ値によって異なる処理を行うことを繰り返し，パケット内のパラメータ値によっては正常にパケット送出が行えないこともあり得る．有限制御部の状態数は 32，初期パラメータの数は 8 個，データレジスタの数は 6 個であった．このような例の，（$\mathcal{Q}_{\mathrm{init}}$ 中の）任意の初期パラメータ値（整数値）に対

表 1　検証実験の結果
Table 1　Results of the experiments.

	性質 1	性質 2	性質 3	性質 4	性質 5
bjd1	77.5	2.9	2.7	2.6	4.8
bjd2	101.3	2.9	2.7	2.7	4.9
bjd3	436.5	92.0	2.7	3.3	92.0

Intel PentiumII 450 MHz, 512MB Memory (単位 秒)

して定義されるすべての FSM/int が，例えば「初期状態からすべてのメディアを含むパケットを送出する状態へ到達する入力系列が存在する」という性質を満たすことを 1.5 秒程度で検証することができた．

6. む す び

本論文では，初期パラメータ値を保持する整数レジスタと入力値を保持することができる整数レジスタをもち，遷移条件を整数レジスタ上の整数線形不等式の論理結合で指定できる拡張有限状態機械モデルに対する記号モデル検査手法を提案し，例題回路を用いて検証に要する時間などを評価した．

本手法ではモデルが現実的なサイズで表現可能か否かではなく，式の真偽判定が現実的な時間で行えるか否かが問題となる．プレスブルガー文の真偽判定手続きによる判定時間は抽象状態の集合を表す式の式長に大きく依存し，レジスタがとり得る値にはさほど依存しない．よって，本手法は与えられた回路をそのデータパスも含めて検証したい場合など，レジスタがとり得る値の範囲が広い（有限状態機械モデルに基づく手法ではうまく扱えないような）回路に対して有効であると考えられる．また，有限状態機械モデルでは，各レジスタがとり得る値の範囲を初期パラメータ（整数値）として与えるようなシステムはうまく扱うことができないが，本手法では扱うことができる．

文　　献

[1] E.M. Clarke and R.P. Kurshan, “Computer-aided verification,” IEEE Spectr., vol.33, pp.61–67, June 1996.

[2] E.M. Clarke, O. Grumberg, and D.A. Peled, Model Checking, MIT Press, 1999.

[3] R.E. Bryant, “Graph-based algorithms for boolean function manipulation,” IEEE Trans. Comput., vol.C-35, no.8, pp.677–691, Aug. 1986.

[4] S. Minato, Binary Decision Diagrams and Applications for VLSI CAD, Kluwer Academic Publishers, 1996.

[5] T. Bultan, R. Gerber, and W. Pugh, “Symbolic model checking of infinite state systems using Presbuerger arithmetic,” Proc. 9th Int’l Conf. on Computer Aided Verification (CAV), vol.1254 of LNCS, pp.400–411, Springer-Verlag, 1997.

[6] W. Chan, R. Anderson, P. Beame, and D. Notkin, “Combining constraint solving and symbolic model checking for a class of systems with non-linear constraints,” Proc. 9th Int’l Conf. on Computer Aided Verification (CAV), vol.1254 of LNCS, pp.316–327, Springer-Verlag, 1997.

[7] R. Alur, T.A. Henzinger, and P.-H. Ho, “Automatic symbolic verification of embedded systems,” IEEE Trans. Softw. Eng., vol.22, no.3, pp.181–201, 1996.

[8] T. Kropf, “Benchmark-circuits for hardware-verification,” Proc. 2nd Int’l Conf. on Theorem Provers in Circuit Design (TPCD), vol.901 of LNCS, pp.1–12, Springer-Verlag, 1994.

[9] ITU-T, “Multiplexing protocol for low bit rate multimedia communication,” ITU-T Recommendation H.223.

[10] W. Kelly, V. Maslov, W. Pugh, E. Rosser, T. Shpeisman, and D. Wonnacott, The Omega Calculator and Library, version 1.1.0, 1996.

（平成 15 年 6 月 3 日受付，9 月 11 日再受付）

竹中　　崇　（正員）

平 7 阪大・基礎工・情報中退．平 9 同大大学院前期課程了．平 12 同大学院後期課程了．同年，日本電気（株）入社．現在，同社マルチメディア研究所在籍．ハードウェアの設計，検証などに関する研究に従事．博士（工学）．

岡野　浩三　（正員）

平 2 阪大・基礎工・情報卒．平 5 同大大学院博士後期課程中退．同年同大助手，平 14 ケント大客員研究員．平 15 バーミンガム大客員講師．現在，阪大情報科学研究科助教授．工博．代数的手法によるプログラム開発，分散システムなどの研究に従事．情報処理学会，IEEE-CS 各会員．

東野　輝夫　（正員）

昭 54 阪大・基礎工・情報卒．昭 59 同大大学院博士課程了．同年同大助手．平 2, 6 モントリオール大学客員研究員．平 3 阪大・基礎工・情報工学科助教授．平 11 基礎工・情報科学科教授．平 14 同大情報科学研究科教授．工博．分散システム，通信プロトコル等の研究に従事．情報処理学会，IEEE, ACM 各会員．

谷口　健一　（正員：フェロー）

昭 40 阪大・工・電子卒．昭 45 同大大学院博士課程了．同年同大・基礎工・助手，現在，同大情報科学研究科教授．工博．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・検証の代数的手法及び支援システム，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・検証法などに関する研究に従事．