



Title	整数入力値を保持するレジスタを持つEFSMに対する記号モデル検査アルゴリズム
Author(s)	竹中, 崇; 岡野, 浩三; 東野, 輝夫 他
Citation	
Version Type	VoR
URL	https://hdl.handle.net/11094/51106
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

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

The University of Osaka

整数入力値を保持するレジスタを持つEFSM に対する記号モデル検査アルゴリズム

Symbolic Model Checking of a Class of Extended Finite State Machines with Integer Input Variables

竹中崇 岡野浩三 東野輝夫 谷口健一
Takashi TAKENAKA Kozo OKANO Teruo HIGASHINO Kenichi TANIGUCHI

大阪大学大学院基礎工学研究科情報数理系専攻
Dept. of Informatics and Mathematical Science,
Graduate School of Engineering Science, Osaka University.

1 はじめに

近年、形式的手法に基づく設計検証に関する研究が注目されている[1]。オートマトンモデルに基づく形式的検証では、回路を有限状態機械でモデル化し、記号モデル検査法[2]を用いて回路が与えられた性質を満たすことを証明する。記号モデル検査法では、BDD[3]を用いて状態の集合を効率的に表現することにより、フリップフロップ数が数百程度の回路の検証が行なえるようになってきているが[4]、レジスタのビット幅が大きくなると、状態数が指数的に増大するため検証は事実上不可能となる。

このため、整数レジスタを持つ拡張有限状態機械(レジスタ付有限状態機械(EFSM))などのモデルに対する効率の良いモデル検査法が望まれている。文献[5]では、遷移条件やレジスタ値の更新関数に整数線形不等式の論理結合(プレスブルガー算術)が利用できるようあるクラスのレジスタ付有限状態機械に対して、到達可能な状態を全て含むような抽象状態の集合を見積り、検証したい性質を満たすことの十分性を判定するようなモデル検査法を提案している。ただし、この手法では回路が検証したい性質を満たすにも関わらず検証に失敗することがありうる。また、文献[6]では、各遷移を保存遷移と更新遷移の二つのタイプに分け、保存遷移では全てのレジスタ値が遷移前の値を保存し、更新遷移では遷移前の状態のレジスタ値に全く依存しないある制約条件を満たすレジスタ値の組に更新されるような拡張有限状態機械モデルに対する状態探索手法を提案している。文献[7]では、実数レジスタ値の変更を差分方程式で指定できるオートマトン(Hybrid automata)に対してモデル検査アルゴリズムを提案している。

本論文では、初期パラメータ値(整数値)と各遷移での入力値(整数値)を保持することのみが許されるレジスタ付有限状態機械で、各遷移の遷移条件として、それらのレジスタの線形不等式の論理結合が記述できるようなモデル(以下、EFSM/presと呼ぶ)に対する記号モデル検査法を提案する。

提案する手法では、整数レジスタ上の線形不等式の論理結合 α と有限制御部の状態 s の組 $[\alpha]@s$ で、有限制御部の状態 s と α を満たすレジスタ値の組からなる抽象状態の集合を表すことにし、“ $[\alpha]@s$ を満たす抽象状態にいつか到達するような(入力値込みの)遷移系列が存在するか”といった時相に関する性質を $\exists\alpha[\alpha]@s$ のようなCTL風の式(以下、CTL/pres式と呼ぶ)で記述する。そして、(初期条件を満たす)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、与えられた性質を満たすことを検証する。

EFSM/presモデル M がCTL/pres式 f を満たすことの記号モデル検査では、CTL/pres式 f の各部分式に対して、内側から順に、その部分式を満たす M の抽象状態の集合を求めていくことにより、 f 全体を満たす抽象状態の集合 Q_f を求め、 Q_f が M の初期抽象状態の集合を含んでいるかどうかを調べる。「部分式 g を満たす抽象状態の集合 Q_g にいつか到達可能

な(入力値込みの)実行系列が存在する」ことを表す抽象状態の集合を求める計算は、 Q_g における各有限制御部の状態から遷移を逆向きにトレースし、その抽象状態の集合に到達可能な抽象状態の集合を順に増やしていくことで行われる(新たな抽象状態が付け加えられなくなるまで繰り返す)。本論文では、対象とするEFSM/presモデルの制約条件により、このトレースが有限回数で必ず終了することを保証している。また、このトレースを終了しても良いか判定を効率よく行うための工夫も行っている。

提案手法の有用性を評価するため、提案アルゴリズムを実装し、ブラックジャックゲームの手札判定回路[8]およびパケット多重化方式プロトコル[9]を例題に、いくつかの性質の検証を行い、有限状態数が100程度、整数レジスタ数が10程度のシステム(有限状態機械モデルではうまく検証が行えないような例)に対して、数秒から数分という実用的な時間で検証が行えることを確認した。

以下では、検証対象とするEFSM/presモデル、検証する性質を表すためのCTL/pres式、記号モデル検査の手順、その停止性および手順の効率化手法、例題による評価、有限状態機械モデルをベースにしたモデル検査法との比較および今後の課題について順に述べる。

2 対象とするレジスタ付有限状態機械

EFSM/presモデルは、初期パラメータ値と入力値を保持するための有限個の整数レジスタを持ち、それぞれ初期パラメータレジスタ、入力レジスタと呼ぶ。各遷移 $t_{ij} = (s_i, C_{ij}, I_{ij}, s_j)$ には、遷移の始状態 s_i と終状態 s_j 、遷移条件 C_{ij} 、この遷移で入力値がセットされるレジスタの集合 I_{ij} が指定される。遷移条件 C_{ij} は、初期パラメータレジスタ及び入力レジスタの値を表す整数変数の線形不等式の論理結合で指定する。遷移条件に出現する入力レジスタは、初期状態からどのように遷移を実行しても必ずその値が確定しているか、あるいは、その遷移で値がセットされるレジスタでなければならない。その遷移で値がセットされる入力レジスタが遷移条件に含まれる場合、その遷移条件を満たす場合のみ入力値が許されると考える。遷移条件 C_{ij} が成立し、遷移 t_{ij} が実行されると、各入力レジスタ $i_1, i_2, \dots, i_k \in I_{ij}$ にこの遷移での入力値 $v_1, v_2, \dots, v_k$ がセットされる。入力レジスタの値は、新たな値がセットされるまでその値を保持する。このモデルでの有限制御部の状態と(値が定まっている)全レジスタ値の直積を抽象状態と呼ぶ。このうち、有限制御部の状態が初期状態 s_0 で、初期パラメータレジスタが条件 α_{s_0} を満たすような抽象状態の集合を Q_{init} と表記する。EFSM/presモデルでは、初期パラメータレジスタが取りうる値に関する条件をこのような初期パラメータ値に関する条件式 Q_{init} で指定し、この条件を満たす初期パラメータ値の組に対してEFSMの動作が定義されていると考える。

一般に、本論文で取り扱うEFSM/presモデルはチューリン

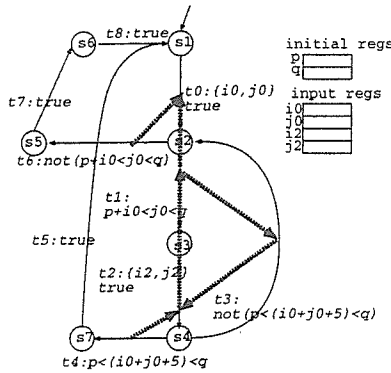


図 1: EFSM/pres の例 M

グ機械を模倣できるため（詳細は後述）、何らかの制約を与えない限り検証の自動化が行えない。本論文では、EFSM 上の各閉路において、その閉路の実行回数がそれ以降の遷移の実行可能性に影響をおよぼさないための制約を与え、記号モデル検査法の手法を用いて、整数パラメタを含むシステムのモデル検査を自動化する方法を考案する。

EFSM/pres モデル M が与えられたとき、 M の動作を表す有向グラフ上において、各入力レジスタ i の値がセットされる遷移からその値が（再びセットされず）最後に参照されるまでの遷移系列 $s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} \dots \xrightarrow{t_{n-1}} s_n$ をそのレジスタの生存遷移系列と呼び、その遷移系列中でレジスタ i の値が生存していると呼ぶことにする。連続する二つの遷移 t_i, t_{i+1} について、 t_{i+1} 以降の動作が t_i 、あるいは t_i 以前の遷移での入力値に依存するように、これらの遷移 t_i, t_{i+1} がともにあるレジスタの生存遷移系列に含まれるとき、 t_{i+1} が t_i に依存すると呼ぶ。EFSM/pres モデル上ですべてのレジスタに対してこのような遷移間の依存関係を求め、依存する遷移間を有向枝として結んで得られる有向グラフを依存遷移グラフと呼ぶ。依存遷移グラフの各閉路に対して、この閉路に閉路外の状態から入射する遷移がある場合、その状態をこの閉路の合流状態と呼び、各合流状態で生存するどの入力レジスタの値もこの閉路内の遷移でセットされない（この閉路内で値が変わらない）とき、その閉路は正則な閉路であるという。本論文では、取り扱う EFSM/pres モデルに以下の制約を課す。

- EFSM/pres モデルの依存遷移グラフに閉路が存在すれば、それらはすべて正則な閉路でなければならない。

依存遷移グラフ上の正則な閉路では、その閉路外でセットされその閉路内で生存しているようなレジスタ値は、閉路中の遷移を何度繰り返して実行しても変化しないし、それ以降の遷移が実行可能となるかどうかの条件判定にも影響を与えない。従って、依存遷移グラフの閉路がすべて正則な閉路であれば、ある遷移 t が実行可能となるかどうかなどの条件判定は依存遷移グラフから正則な閉路を取り除いたグラフ上で順に依存する（有限長の）遷移の実行にのみ影響を受ける。反対に、依存遷移グラフ上に正則な閉路以外の閉路が存在するとき、入力レジスタ i に対して、補助変数となる他の入力変数を用いることにより、 i の値を 1 ずつ増減させるような代入 ($i+1$ や $i-1$ の代入) やゼロ判定を行う記述が可能となる。そのことは、2 カウンタオートマトンを模倣できることを意味し、一般に到達可能性などの判定問題が決定不能となる。

EFSM/pres モデルではレジスタ間の演算や代入が許されていないため、一見すると対象とする回路の種類が大幅に限られるように見える。しかし、レジスタ間の代入があっても、演算が整数の線形演算に限られ、代入の依存関係に閉路が存在しないような EFSM は、遷移条件の代入されるレジスタのすべての出現を入力レジスタの表現式に書き換えることで、EFSM/pres

モデルに変形可能である。また、代入の依存関係に閉路が存在しても、その代入の回数に上限がある場合は同様に EFSM/pres モデルに変形可能である。したがって、本手法は多くの回路に適用できると考えられる。

[例 1] (EFSM/pres モデルの例)

図 1 に EFSM/pres モデルの例 M を示す。初期パラメタレジスタの集合は $\{p, q\}$ 、入力レジスタの集合は $\{i_0, j_0, i_2, j_2\}$ である。有限制御部の初期状態は s_1 である。各遷移には、遷移名及びその遷移で値がセットされる入力レジスタの集合 I_{ij} 、遷移条件 C_{ij} を示している。 I_{ij} が空集合であるときは記述を省略している。

図 1 の EFSM/pres モデル M の依存遷移グラフを同じ図 1 上に太破線で示している。この依存遷移グラフの閉路は遷移 t_1, t_2, t_3 からなる閉路のみである。この閉路の合流状態は s_2 のみである。 s_2 で生存する入力レジスタ i_0, j_0 の値をこの閉路 t_1, t_2, t_3 上の遷移においてセットしていないので、この閉路は正則な閉路である。よって、この EFSM/pres モデル M は本論文の制約を満たしている。 $\square$

3 検証する性質

以下では、整数レジスタ上の線形不等式の論理結合 α と有限制御部の状態 s の組 $[s]@s$ を State formula と呼ぶ。State formula は、有限制御部の状態が s でレジスタ値が α を満たすような抽象状態の集合を表している。

[定義 1] (State Formula)

State Formula SF は次の文法で定義される。

$$\begin{aligned} \text{SF} &::= [sp(s)]@s \\ sp(s) &::= g(s) \mid (sp(s)) \mid sp(s) \wedge sp(s) \\ &\quad \mid \neg sp(s) \mid \text{true} \mid \text{false} \\ g(s) &::= t(s) < t(s) \\ t(s) &::= (t(s)) \mid t(s) + t(s) \mid -t(s) \\ &\quad \mid x \in DVar(s) \mid 1 \mid 2 \mid \dots \end{aligned}$$

ここで、 $DVar(s)$ は有限制御部の状態 s で値が確定している入力レジスタおよび初期パラメタレジスタの集合であり、 $sp(s)$ はこのようなレジスタ上の整数線形不等式の論理結合を表している。 $\square$

[定義 2] (State Formula のセマンティクス)

State Formula のセマンティックスを以下のように定める。

$$[sp(s)]@s = \{(s, r_0, r_1, \dots) \mid sp(s)\} \quad \square$$

上述のように State Formula “ $[sp(s)]@s$ ” は、有限状態が s で各レジスタの値が $sp(s)$ を満たすような抽象状態の集合を表す。

また、State Formula の集合和、集合積に関して以下の性質が成り立つ。

$$\begin{aligned} [\alpha_{s_i}]@s_i \cup [\beta_{s_i}]@s_i &= [\alpha_{s_i} \vee \beta_{s_i}]@s_i \\ [\alpha_{s_i}]@s_i \cap [\alpha_{s_j}]@s_j &= \begin{cases} [\alpha_{s_i} \wedge \alpha_{s_j}]@s_i & : (s_i = s_j) \\ \emptyset & : (s_i \neq s_j) \end{cases} \end{aligned}$$

本論文では、このような抽象状態の集合に“いつか到達する”ような（入力値込みの）実行系列が存在するか”といった時相に関する性質を検証すべき性質として、以下で定義する CTL 風の式 (CTL/pres 式と呼ぶ) を用いて記述する。

[定義 3] (CTL/pres 式の文法)

CTL/pres 式 f は次の文法で定義する。

$$\begin{aligned} f &::= \text{SF} \mid \neg f \mid f_1 \cap f_2 \mid f_1 \cup f_2 \mid \\ &\quad \exists \bigcirc f \mid \exists \bigcirc f \mid f_1 \exists \mathcal{U} f_2 \quad \square \end{aligned}$$

以下では、表記上の簡便のため次のような記法を用いる。

$$\begin{aligned}\forall \bigcirc f &\equiv \neg \exists \bigcirc \neg f \\ \exists \bigcirc f &\equiv \text{true} \exists \bigcirc f \\ \forall \square f &\equiv \neg(\text{true} \exists \bigcirc \neg f) \\ \forall \diamond f &\equiv \neg \exists \bigcirc \neg f \\ f \forall \bigcirc g &\equiv \neg(g \exists \bigcirc (\neg f \wedge \neg g)) \wedge \neg \exists \bigcirc \neg g\end{aligned}$$

CTL/pres 式 f のセマンティクスは EFSM/pres モデル M の実行系列 $(q_0, q_1, \dots)$ 上で定義される。実行系列とは M 上のある抽象状態から適当な入力系列を与えて得られる抽象状態の系列である。実行系列の任意の連続する 2 つの抽象状態 (q_i, q_{i+1}) (有限制御部の状態をそれぞれ s_i, s_{i+1} とする) に対して、 $(s_i, C_{i,i+1}, I_{i,i+1}, s_{i+1}) \in T$ が存在し、この遷移で入力値がセットされるレジスタが存在するならば q_i において C_{ij} を満たすような入力値が存在しなければならず、そのようなレジスタがない場合には q_i において C_{ij} が満たされていなければならない。

[定義 4] (CTL/pres 式のセマンティクス)

EFSM/pres モデル M と CTL/pres 式 f が与えられたとき、 f の値を以下のように定義する。ここで q および q_i は M の抽象状態を表す。

$$\begin{aligned}q \models \text{SF} &\text{ iff. } q \in \text{SF} \\ q \models \neg f &\text{ iff. } q \not\models f \\ q \models f \cap g &\text{ iff. } q \models f \text{ かつ } q \models g \\ q \models f \cup g &\text{ iff. } q \models f \text{ または } q \models g \\ q_0 \models \exists \bigcirc f &\text{ iff. } q_0 \text{ から始まるある実行系列 } (q_0, q_1, \dots) \\ &\text{ が存在し, } q_1 \models f \text{ が成り立つ。} \\ q_0 \models \exists \square f &\text{ iff. } q_0 \text{ から始まるある実行系列 } (q_0, q_1, \dots) \\ &\text{ の全ての } q_j \text{ に対して } q_j \models f \text{ が成り立つ。} \\ q_0 \models f \exists \bigcirc g &\text{ iff. } q_0 \text{ から始まるある実行系列 } (q_0, q_1, \dots) \\ &\text{ に対してある } i \text{ が存在して } q_i \models g \text{ かつ } \\ &\text{ かつ } j < i \text{ なる全ての } j \text{ に対して } q_j \models f \\ &\text{ が成り立つ。}\end{aligned}$$

CTL/pres 式 f をみたす EFSM/pres モデル M の抽象状態の集合を $Q_f = \{q \mid q \models f\}$ と表記する。また、 $Q_{\text{init}} \subseteq Q_f$ であるとき $M \models f$ であると表記し、 M が f を満たすと言う。□

[例 2] (CTL/pres 式の例)

例 1 の EFSM/pres モデル M に対する CTL/pres 式の例を以下に示す。

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

この CTL/pres 式は、「初期状態より始まる遷移系列に、有限制御部の状態が s_7 で、かつ、レジスタ値が条件 $(0 < j_0 + 3) \wedge (i_0 < i_2) \wedge (i_2 < p + j_0)$ を満たすような抽象状態にいつか到達するような実行系列が存在する」ことを表している。□

4 記号モデル検査アルゴリズム

本節では、EFSM/pres モデル M とこれに対する CTL/pres 式 f が与えられたとき、 M が f を満たすことを検証する記号モデル検査について述べる。

記号モデル検査では、CTL/pres 式 f の各部分式が成立する M の抽象状態の集合をそれぞれ求めることにより、帰納的に f 全体が成立する抽象状態の集合 Q_f を求める。求めた抽象状態の集合 Q_f が M の初期抽象状態の集合 Q_{init} を含んでいれば、 $M \models f$ が成立する。

4.1 CTL/pres 式 $\exists \bigcirc f$ をみたす抽象状態の集合

CTL/pres 式 f を満たす抽象状態の集合 Q_f が求まっているとき、CTL/pres 式 $\exists \bigcirc f$ を満たす抽象状態の集合 $Q_{\exists \bigcirc f}$ を求めることを考える。

ここで Q_f を $Q_f = \bigcup_{s_j \in S} [\alpha_{s_j}] @ s_j$ と表す。このとき Q_f 中の抽象状態に一遷移で到達可能な抽象状態は、各状態 s_j への遷移 $t_{ij} = (s_i, I_{ij}, C_{ij}, s_j)$ の実行後に各レジスタ値が α_{s_j} を満たすような抽象状態である。本論文で対象としている EFSM/pres モデルでは、各レジスタの値はその遷移によって値がセットされる入力レジスタ以外は変化しない。したがって、遷移後に α_{s_j} を満たすためには、この遷移でセットされる入力がない場合は遷移前に論理式 $(C_{ij} \wedge \alpha_{s_j})$ が成り立たなければならない。この遷移でセットされる入力がある場合には論理式 $(C_{ij} \wedge \alpha_{s_j})$ が成り立つような入力値が存在しなければならない。したがって、CTL/pres 式 $\exists \bigcirc f$ を満たす抽象状態の集合 $Q_{\exists \bigcirc f}$ は以下のようになる。

$$Q_{\exists \bigcirc f} = \bigcup_{s_j \in S} \bigcup_{\substack{s_i \in S \\ (s_i, C_{ij}, I_{ij}, s_j) \in T}} ([\exists i \in I_{ij}. (C_{ij} \wedge \alpha_{s_j})] @ s_i)$$

ただし、 $\exists i \in I_{ij}. (C_{ij} \wedge \alpha_{s_j})$ は

$$\begin{aligned}(C_{ij} \wedge \alpha_{s_j}) &: I_{ij} = \emptyset \text{ のとき} \\ \exists i_1, i_2, \dots, i_k. (C_{ij} \wedge \alpha_{s_j}) &: I_{ij} = \{i_1, i_2, \dots, i_k\} \text{ のとき}\end{aligned}$$

の略記である。

4.2 CTL/pres 式 $f \exists \bigcirc g, \exists \square f$ を満たす抽象状態の集合

次に、CTL/pres 式 f, g を満たす抽象状態の集合 Q_f, Q_g が求まっているときに、CTL/pres 式 $f \exists \bigcirc g, \exists \square f$ を満たす抽象状態の集合 $Q_{f \exists \bigcirc g}, Q_{\exists \square f}$ を求めることを考える。

4.2.1 抽象状態の集合 $Q_{f \exists \bigcirc g}$ を求める計算の基本手順

いまここに CTL/pres 式 f, g を満たす抽象状態の集合 Q_f, Q_g が以下のように求められているとする。

$$\begin{aligned}Q_f &= \bigcup_{s_k \in S} [\gamma_{s_k}] @ s_k \\ Q_g &= \bigcup_{s_j \in S} [\alpha_{s_j}] @ s_j\end{aligned}$$

$Q_{f \exists \bigcirc g}$ は、以下のように、 Q_g 中の抽象状態に到達可能、且つ、到達するまでに経由するすべての抽象状態が Q_f に含まれるような抽象状態を、新たな抽象状態が求められなくなるまで順にすべて求め、追加していくことで得られる。

$$\begin{aligned}Q_0 &= Q_g \\ Q_{n+1} &= Q_n \cup (Q_f \cap \exists \bigcirc Q_n) \\ Q_{f \exists \bigcirc g} &= Q_\infty\end{aligned}$$

ただし、 Q_n は Q_g 中の抽象状態に n 回以下の遷移の実行で到達可能、且つ、到達するまでに経由するすべての抽象状態が Q_f に含まれるような抽象状態の集合である。

ここで、「 Q_g 中の抽象状態にちょうど n 回の遷移の実行で到達可能、且つ、到達するまでに経由するすべての抽象状態が Q_f に含まれるような抽象状態」の集合 Q_n (Q_{n-1} から Q_n を求める際に新たに求められるような抽象状態) は、深さ優先探索的に、 Q_g の部分集合 $[\alpha_{s_j}] @ s_j$ それぞれに対して、各有限状態 s_j から EFSM/pres モデル M 上の遷移を一遷移ずつ n 回逆にトレースすることによって得られる。たとえば、ある有限状態 s_j から、 n 個の遷移からなるある遷移系列 $\sigma_{ij} = s_i \xrightarrow{t_1} \dots \xrightarrow{t_{n-1}} s_i \xrightarrow{t_n} s_j$ を一遷移ずつ逆にトレースして得られる抽象状態の集合は $[\beta_{\sigma_{ij}}] @ s_i$ となる。ただし、 $\beta_{\sigma_{ij}}$ は各遷移の遷移条件および途中の状態 s_m において (経由するすべての抽象状態が Q_f に含まれるために) みたすべき条件 γ_{s_m} と存在限定子が順にネストした条件

$$\begin{aligned}\beta_{\sigma_{ij}} &= (\gamma_{s_i} \wedge [\exists i \in I_n. (C_n \wedge \dots \\ &\dots \wedge (\gamma_{s_1} \wedge [\exists i \in I_1. (C_1 \wedge \alpha_{s_j})]) \dots])\end{aligned}$$

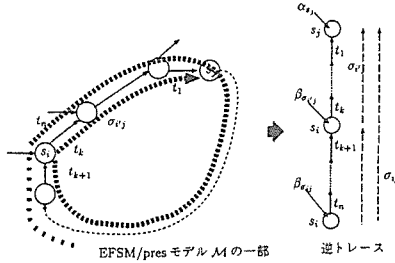


図 2: 遷移系列 σ_{ij} の逆トレース中に 2 度訪問するような有限状態 s_i が存在する場合

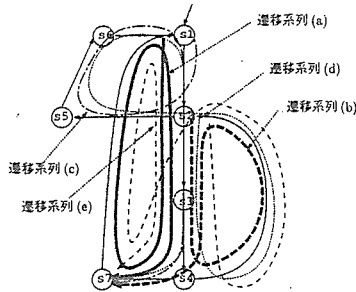


図 3: EFSM/pres モデル M 上の逆トレース

である。したがって、 Q'_n は各有限状態 s_j に至る長さ n の遷移系列すべてについて、このように一遷移づつ逆にトレースすることで得られる。また、 $Q_f \cup Q_g$ は、このような逆トレースをそれぞれ新たな状態を求める必要がなくなるまで続けることで得られる。

逆トレースをさらに続けて新たな状態を求める必要があるかの判定は、1) その逆トレースで経由した遷移系列が実行不能か否か、および、2) その逆トレースを続けて新たに求められる抽象状態の集合がそれまでの計算で既に求められている抽象状態の集合に含まれているか否かの判定で行う。前者は、条件 $\beta_{\sigma_{ij}}$ が充足可能か否かの判定で行う。後者は、遷移系列 σ_{ij} にそった逆トレースによって新たに求められた抽象状態の集合 $[\beta_{\sigma_{ij}}]@s_i$ が、遷移系列 σ_{ij} の後半の部分系列 $\sigma'_{ij} = s_{i'} \xrightarrow{t_{k+1}} \dots \xrightarrow{t_n} s_j$ にそったより短い逆トレースですでに求められている状態集合 $[\beta_{\sigma'_{ij}}]@s_{i'}$ と同じか、または、含まれているか否かの判定で行う。 $[\beta_{\sigma_{ij}}]@s_i \subseteq [\beta_{\sigma'_{ij}}]@s_{i'}$ となるためには、1) s_i と $s_{i'}$ が同じ状態であり（これは、遷移系列 σ_{ij} にそった逆トレース中に同じ状態 s_i を二度訪問していることに相当する。図 2 参照）、且つ、2) $\beta_{\sigma_{ij}}$ が $\beta_{\sigma'_{ij}}$ より同じかきつい条件となる、すなわち $\beta_{\sigma_{ij}} \rightarrow \beta_{\sigma'_{ij}}$ が全ての自由変数の任意の解釈のもとで恒真である、の 2 条件を満たせばよい。このとき、これ以上逆にトレースを続けて得られる抽象状態は、この逆トレースによってすでに求められた抽象状態か、これより短い遷移系列を経由する他の逆トレースにより求められる抽象状態となる。したがって、これ以上の逆トレースは行わず、逆トレースが有限長となる。

本論文で対象とする EFSM/pres モデルは、その遷移条件が整数変数上の線形不等式の論理結合で記述されているため、条件 $\beta_{\sigma_{ij}}$ などは整数変数上の線形不等式の（限定子を含む）論理結合（プレスブルガー文 [10]）となる。そこで、これらの条件の充足可能性や包含関係の判定は、プレスブルガー文の真偽判定問題に帰着し行う。

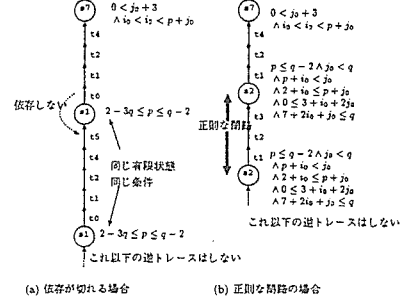


図 4: 逆トレースによる抽象状態の集合の計算

ここで注意すべきことは、逆トレース中に二度訪問するどのような状態 s_i に対しても $\beta_{\sigma_{ij}}$ が $\beta_{\sigma'_{ij}}$ よりきつい条件が同じ条件となるとは限らないということである。前述のように $\beta_{\sigma_{ij}}$ 、 $\beta_{\sigma'_{ij}}$ はそれぞれ遷移条件と限定子がネストした条件となっているため、 $\beta_{\sigma_{ij}}$ は $\beta_{\sigma'_{ij}}$ を用いて

$$\beta_{\sigma_{ij}} = (\gamma_{s_i} \wedge [\exists i \in I_n. (C_n \wedge \dots \wedge (\gamma_{s_{k+1}} \wedge [\exists i \in I_{k+1}. (C_{k+1} \wedge \beta_{\sigma'_{ij}}))] \dots)])$$

と表される。このため、 $\beta_{\sigma_{ij}}$ では $\beta_{\sigma'_{ij}}$ の自由変数（前半の部分系列の遷移 $t_1, \dots, t_k$ の遷移条件に出現する入力レジスタのうちこの値をセットせずに参照しているもの）をその外の存在限定子で束縛することがあり、このとき $\beta_{\sigma_{ij}}$ は $\beta_{\sigma'_{ij}}$ より緩い条件となりうるからである。このようなとき、 $[\beta_{\sigma_{ij}}]@s_i$ 中の抽象状態であるが $[\beta_{\sigma'_{ij}}]@s_{i'}$ 中の抽象状態でないような抽象状態は、直感的にはその入力レジスタ値のままでは遷移系列 σ'_{ij} のみを経由して $[\alpha_{s_j}]@s_j$ に到達可能ではないが、遷移系列 σ'_{ij} の前半の部分系列を経由して入力レジスタ値を新たにセットしなおせば、そのあとに遷移系列 σ'_{ij} を経由して $[\alpha_{s_j}]@s_j$ に到達可能となるような抽象状態である。

4.2.2 抽象状態の集合 $Q_{\exists \text{of}}$ を求める計算の基本手順
いまここに CTL/pres 式 f を満たす抽象状態の集合 Q_f が以下のように求められているとする。

$$Q_f = \bigcup_{s_i \in S} [\delta_{s_i}]@s_i$$

$Q_{\exists \text{of}}$ は、 $Q_f \cup Q_g$ を求めたときは反対に、各有限状態 s_i から EFSM/pres モデル M 上の遷移を一遷移づつ順にトレースし、経由する抽象状態が全て Q_f に含まれる実行系列が存在するような Q_f の部分集合となる抽象状態の集合を求めることにより得られる。

抽象状態の集合 Q_f 中のある抽象状態の集合 $[\delta_{s_i}]@s_i$ について、有限状態 s_i からある遷移系列 $\sigma_{ij} = s_i \xrightarrow{t_1} \dots \xrightarrow{t_n} s_j$ にそって順にトレースすることにより、この遷移系列にそった実行系列中に経由する抽象状態がすべて Q_f に含まれる抽象状態の集合 $[\eta_{\sigma_{ij}}]@s_i$ ($[\delta_{s_i}]@s_i$ 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

$$\eta_{\sigma_{ij}} = \delta_{s_i} \wedge [\exists i \in I_1. (C_1 \wedge \delta_1 \wedge \dots \wedge [\exists i \in I_n. (C_n \wedge \delta_n)] \dots)]$$

である。抽象状態の集合 $Q_{\exists \text{of}}$ は、このような各有限状態からのすべてのトレースを抽象状態の集合がそれ以上小さくなくなるまで続けてそれぞれ得られた抽象状態の集合和となる ($Q_f \cup Q_g$ とは異なり、トレース途中で得られた抽象状態を逐次追加することはしない)。

トレースをさらに続ける必要があるか否かの判定は、1) そのトレースで経由した遷移系列が実行可能か否か、および、2) 今後トレースを続けて新たに求められる抽象状態がそれまでのトレースで既に求められている抽象状態の集合より小さくなる可能性があるか否かの判定で行う。前者は、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。後者は、逆トレースを行った遷移系列 σ_{ij} に対してその前半の部分系列 $\sigma_{ij} = s_i \xrightarrow{t_k} \dots \xrightarrow{t_k} s_{j'}$ を考えたときに、1) s_j と $s_{j'}$ が同じ状態であり（これは、遷移系列 σ_{ij} にそったトレース中に同じ状態 s_j を二度訪問していることに相当する）、且つ、2) $\eta_{\sigma_{ij}}$ が $\eta_{\sigma_{ij'}}$ と同じ条件であるか、すなわち $\eta_{\sigma_{ij}} \leftrightarrow \eta_{\sigma_{ij'}}$ が全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} 実行後にさらに遷移系列 $s_{j'} \xrightarrow{t_{k+1}} \dots \xrightarrow{t_n} s_j$ 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+} をいくらか続けてトレースを行っても、得られる状態集合は $[\eta_{\sigma_{ij}}]@s_i$ より小さくならず、それ以上のトレースを行わない。このとき、 $[\eta_{\sigma_{ij}}]@s_i$ 中の任意の抽象状態から、無限長の遷移系列 σ_{ij+} の実行したとき途中で経由する抽象状態はすべて Q_f に含まれるので、 $[\eta_{\sigma_{ij}}]@s_i \in Q_{\exists \cup g}$ 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、トレース中に同じ状態に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 $Q_f \cup g$ を求めるときと同様である。

4.2.3 抽象状態の集合を求める計算の有限性

ここでは、抽象状態の集合 $Q_f \cup g$ を求める計算が有限ステップで終了することを示す。

まず、与えられた EFSM/pres モデルの有限制御部の構造から逆トレースを打ち切るか否かを判定できることを示す。本論文で提案しているクラスの EFSM/pres では、 s_j から遷移系列 σ_{ij} にそって逆にトレースしたときに、同じ状態 s_i を二度訪問し、且つ、以下の2つの条件のいずれかが成り立つとき、 $[\beta_{\sigma_{ij}}]@s_i \supseteq [\beta_{\sigma_{ij'}}]@s_i$ となり、逆トレースを打ち切ることができる。

(1) 遷移系列 σ_{ij} の途中の状態 s_i の前後の遷移 t_{k+1}, t_k において t_{k+1} に t_k が依存していないとき (s_j で依存が切れると呼ぶ)。たとえば、図3の s_1 から s_7 に至る遷移系列 (a) (図4(a)参照)。

(2) 遷移系列 σ_{ij} の前半の部分系列が依存遷移グラフ上の正則な閉路であり、かつ、 s_i がその合流状態である場合。たとえば、図3の s_2 から s_7 に至る遷移系列 (b) (図4(b)参照)。これは、(依存関係や正則な閉路の定義より) $\beta_{\sigma_{ij}}$ の自由変数の値を前半の部分系列でセットしない ($I_n, \dots, I_{k+1}$ に出現しない) ので、条件 $\beta_{\sigma_{ij}}$ が条件 $\beta_{\sigma_{ij'}}$ を用いて

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

と表すことができるためである。

次に、上記の性質を利用して、 $Q_f \cup g$ を求める計算では無限長の逆トレースを考えなくても良いことを示す。本論文の EFSM/pres のクラスでは、依存遷移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、正則な閉路をふくまないような順に依存しあ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 $[\alpha_{s_i}]@s_j$ に対して状態 s_j からの逆トレースを考えると、次の理由より無限長のトレースは存在しない。

ここで、仮に無限長の逆トレースが存在するとする。正則な閉路となる遷移の系列や、正則な閉路をふくまないような順に依存しあう遷移系列はともにその長さが有限であるので、こ

の無限長の逆トレース中には前後の遷移が依存しないような状態や正則な閉路中の合流状態であるような状態が無限に存在する。EFSM/pres には、その前後の遷移が依存しないような遷移の組合わ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、無限の逆トレース中に存在する、前後の遷移が依存しないような状態や正則な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度以上出現する状態となる。逆トレース中にこのような制御部状態を二度訪問したときは、そのトレースを終了することができるため、この場合にはこの逆トレースは有限長となり、これは仮定に反する。

したがって、EFSM/pres モデルでは抽象状態の集合 Q_g 中の各抽象状態の集合 $[\alpha_{s_j}]@s_j$ に対して、状態 s_j からの逆トレースを考えると、全ての逆トレースが有限長になるため部分式 $f \exists \cup g$ を満たす状態集合を求める計算が有限時間で終了する。

抽象状態の集合 $Q_{\exists \cup g}$ を求める計算が有限ステップで終了することも、 $Q_f \cup g$ を求める計算の時と同様の議論で示すことができる。

4.2.4 抽象状態の集合を求める計算の効率化

前述の(4.2.1節で述べた)手順では、抽象状態の集合 $[\alpha_{s_j}]@s_j$ に対して状態 s_j からの逆トレースを考えると、その逆トレースを終了するか否かの判定は、プレスブルガー文の真偽判定問題に帰着して行っている。しかし、プレスブルガー文の真偽判定問題はNP困難であるため、以前に訪問したことのある有限状態に再び訪問するたびにこの判定を毎回行うことは非効率である。

ところが4.2.3節で述べたように、遷移系列 σ_{ij} にそった逆トレース中に二度訪問する状態 s_i の前後の遷移 t_{k+1}, t_k が依存しないか、あるいは前半の遷移系列が正則な閉路である場合、遷移系列 σ_{ij} すべてを逆トレースして得られる抽象状態の集合は、後半の部分系列 $\sigma_{ij'}$ を逆トレースして得られる抽象状態の集合と同じに含まれるので、この逆トレースを終了しても良い。そこで、逆トレース中に以前に訪問済の状態 s_i を再び訪問したときに、この状態で逆トレースを終了するか否かは、これらの評価をもとに決定する。これにより、記号モデル検査に必要な計算時間が削減できる。

ただし、CTL/pres 式 f の各部分式が成立する M の抽象状態の集合を順に求めて得られた抽象状態の集合が M の初期抽象状態の集合 Q_{init} を含んでいるかの判定は、プレスブルガー文真偽判定問題に帰着して行わなければならない。

4.3 モデル検査の例

例1のEFSM/presモデル M のもとで例2のCTL/pres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、 $\exists \cup [(0 < j_0 + 3) \wedge (i_0 < i_2) \wedge (i_2 < p + j_0)]@s_7$ を満たす抽象状態の集合を求めることを考える。考慮すべき状態 s_7 からの逆トレースは、4.2.4節の条件より、図3の遷移系列(a), (b), (c), (d), (e)を経由するような逆トレースである。

たとえば、状態 s_2 では、遷移系列(b)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)の全てを経由する条件共に、 $\beta_0 = (p \leq q - 2) \wedge (j_0 < q) \wedge (p + i_0 < j_0) \wedge (2 + i_0 \leq p + j_0) \wedge (0 \leq 3 + i_0 + 2j_0) \wedge (7 + 2i_0 + j_0 \leq q)$ であり、同じ条件になっていることがわかる。これら全ての遷移系列について逆トレースを行うことにより、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 Q_f に初期抽象状態の集合が含まれ、 $M \models f$ 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なう回路である。

表 1: 検証実験の結果

	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 450MHz, 512MB Memory (単位 秒)

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 が出力されたときに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 EFSM/pres モデル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、初期パラメタレジスタは持たない。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 $\forall \alpha (\exists \beta (\text{true} @ s_0))$ のような CTL/pres 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、且つ、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/pres 式で表される。

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

ただし、 S_{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、且つ、Stand が真である状態に到達することがない。

(性質 4) カードが 5 枚入力され、且つ、それらが全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はない。

また、カードの最大値、および有効なスコアの範囲を初期パラメタとして扱い、(Q_{init} 中の) 任意の初期パラメタ値 (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うことができない。

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[11] を利用した。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 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タとして与えられる。有限制御部の状態数は 32, 初期パラメタの数は 8 個、入力レジスタの数は 6 個であった。このような例の、(Q_{init} 中の) 任意の初期パラメタ値 (整数値) に対して定義されるすべての EFSM/pres が、たとえば「初期状態から全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 あとがき

本論文では、初期パラメ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持ち、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。

各レジスタがとりうる値の範囲が決められていれば、有限状態機械モデルをベースにしたモデル検査法を行うことが可能となるが、このような場合との比較について述べる。一般に有限状態機械上での記号モデル検査で現実的に検証が可能か否かは、現実的なサイズの BDD でモデルを表現することが可能か否かが問題となる。モデルの大きさはレジスタのビット幅に大きく依存する。一方、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、レジスタがとりうる値にはさほど依存しない。このように、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、レジスタがとりうる値の範囲が広い (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、各レジスタがとりうる値の範囲を初期パラメタ (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。

本手法では、有限制御部の状態数が多くなるとトレースすべき遷移系列の数や、プレスブルガー文真偽判定手続きで判定すべき式の式長が増加し、計算時間が多くなる傾向が見られた。そこで、今後の課題は有限制御部の状態数が多くなっても効率的に検証を行うことができるように拡張することなどが考えられる。

参考文献

- [1] Clarke, E. M. and Kurshan, R. P.: Computer-aided verification, *IEEE Spectrum*, Vol. 33, No. 6, pp. 61-67 (1996).
- [2] Clarke, E. M., Grumberg, O. and Peled, D. A.: *Model Checking*, The MIT Press (1999).
- [3] Minato, S.: *Binary Decision Diagrams and Applications for VLSI CAD*, Kluwer Academic Publishers (1996).
- [4] Burch, J. R., Clarke, E. M., McMillan, K. L., Dill, D. L. and Hwang, L. J.: Symbolic Model Checking: 10²⁰ States and Beyond, *Information and Computing*, Vol. 98, No. 2, pp. 142-170 (1992).
- [5] Bultan, T., Gerber, R. and Pugh, W.: Symbolic model checking of infinite state systems using Presburger arithmetic, in *Proc. of 9th Int'l Conf. on CAV*, Vol. 1254 of *LNCS*, pp. 400-411, Springer-Verlag (1997).
- [6] Chan, W., Anderson, R., Beame, P. and Notkin, D.: Combining Constraint Solving and Symbolic Model Checking for a Class of Systems with Non-linear Constraints, in *Proc. of 9th Int'l Conf. on CAV*, Vol. 1254 of *LNCS*, pp. 316-327, Springer-Verlag (1997).
- [7] Alur, R., A. Henzinger, T. and Ho, P.-H.: Automatic Symbolic Verification of Embedded Systems, in *Proc. of 14th TRSS*, pp. 2-11 (1993).
- [8] Kropf, T.: Benchmark-Circuits for Hardware-Verification, in *Proc. of 2nd Int'l Conf. TPCD*, Vol. 901 of *LNCS*, pp. 1-12, Springer-Verlag (1994).
- [9] 大久保栄: マルチメディアの基礎技術 8 同期多重分離技術, *bit*, Vol. 29, No. 8, pp. 97-105 (1997).
- [10] 東野輝夫, 北道淳司, 谷口健一: 整数上の線形制約の処理と応用, *コンピュータソフトウェア*, Vol. 9, No. 6, pp. 31-39 (1992).
- [11] Kelly, W., Maslov, V., Pugh, W., Rosser, E., Shpeisman, T. and Wonnacott, D.: *The Omega Calculator and Library, version 1.1.0* (1996).