マルチメディア通信と分散処理 70−18
（1995. 5. 26）

ECFSMモデル通信プロトコルの検証システムにおける不変式の自動生成

樋口 昌宏　玉井 順子　原 圭吾　藤井 護

大阪大学 基礎工学部 情報工学科

e-mail: higuchi@ics.es.osaka-u.ac.jp

筆者らは，プロトコル機械が拡張有限状態機械でモデル化され，通信路が非有界FIFOでモデル化された通信プロトコルの安全性，生存性の検証法を提案している．提案している検証法では，人間の検証者が記述した検証対象であるプロトコルに関する不変式を基に，検証システムがそのプロトコルの安全性，あるいは生存性を検証する．この検証法では，不変式の記述に要する検証者の作業量が大きなものとなってしまうという問題点がある．本稿では，不変式記述作業の軽減を目的に，積和形で記述される不変式の積項のうち，一方のプロトコル機械のメッセージの送信に同期して他方のプロトコル機械のメッセージの受信が行なわれる同期型通信で到達可能な状態と，それらの状態からメッセージの送信のみによって到達可能な状態で成り立つ積項を自動生成する手法を提案する．提案する手法に基づく積項の自動生成システムを試作し，OSIセションプロトコルのデータ転送フェーズから抽出した例プロトコルに適用したところ，不変式の積項472個のうち445個の積項を自動生成することができた．

Automatic Generation of Invariant Formula for Communication Protocols Modeled as ECFSMs

Masahiro HIGUCHI, Junko TAMAI, Keigo HARA, and Mamoru FUJII

Department of Information and Computer Sciences
Osaka University

e-mail: higuchi@ics.es.osaka-u.ac.jp

Previously, we proposed a verification method via invariant for communication protocols modeled as ECFSMs. In the proposed method, a human verifier describes an invariant of a given protocol, and a verification system shows safety or liveness using the invariant. The tedious work on describing invariant formula is a significant shortcoming of the proposed method. This paper deal with a semiautomated derivation of invariant formula for communication protocols modeled as ECFSMs. In the method, formula holds on the states, which is reachable by synchronous communication and sending transitions, are automatically derived. We conducted an experiment on deriving disjuncts of invariant formula of a sample protocol extracted from the OSI session protocol. Among 472 disjuncts of an invariant formula, 445 disjuncts are automatically derived.

1 まえがき

信頼性の高い通信ソフトウェアの実現にはプロトコル設計段階での論理検証が重要である．通常，二者間の通信を規定するプロトコルは順序機械でモデル化される二つのプロトコル機械とそれらを接続する双方向の FIFO でモデル化される通信路からなる系としてモデル化される．実用レベルの通信プロトコルはプロトコル機械が有限制御部の他に整数値などの値を取る変数を持つ拡張有限状態機械 Extended Communicationg Finite-State Machines （以下，ECFSM と呼ぶ）でモデル化される場合が多い．実際，通信システムの標準仕様記述言語である SDL[1] や Estelle[2] は ECFSM モデルに基づいて設計されている．筆者らの研究グループでは通信路の有界性を仮定しない ECFSM モデルのプロトコルに対する安全性，生存性の検証法として，検証者が検証の対象としているプロトコル Π について不変式と思われる論理式 F を記述し，検証システムが F が実際に不変式であることを示すなどにより，Π の安全性，生存性を証明する手法を提案し，それに基づく検証システムが試作している[3] [4]．提案している検証法の問題点として，論理式 F の記述に費やす労力が大きいということがあげられる．実際，OSI セションプロトコルを 9 つのサブプロトコルに分割して安全性の検証を行なう実験では，UNIX ワークステーション上で使用した総 CPU 時間が約 458 秒であったのに対し，検証者による論理式の記述作業に約 120 時間を要した[5]．一方，この検証実験では，メッセージのすれ違いなどの二つプロトコル機械の複雑な相互干渉により到達可能となる状態はほとんどなく，積和形論理式で記述する不変式の積項の大部分については単純な作業の繰り返しにより導出できるであろうという知見が得られた．

本稿では，提案している検証法における検証者の作業を軽減をはかる手法について述べる．具体的には，対象となるプロトコルの到達可能な状態のうち，以下の状態に関する積項を自動生成する．

(i) 一方のプロトコルのメッセージの送信に同期して，他方のプロトコル機械がメッセージの受信を行なう，同期型の通信のみによって到達可能な状態．(ii) 同期通信のみで到達可能な状態から，双方のプロトコル機械のメッセージの送信のみによって到達可能な状態．不変式を得るためには，自動生成した積項に加えて，(i),(ii) の状態以外の到達可能状態に関する積項を人間の検証者が記述することになるが，到達可能な状態のほとんどが (i),(ii) に相当するようなプロトコルについては，本稿で提案する手法により検証者の作業が大きく軽減される．

以下 2. では準備として，プロトコルモデルと，文献 [3] で提案している安全性の検証法について説明し，3. で積項の自動生成手法，4. で例プロトコルに対する検証実験の結果について説明する．

2 準備

2.1 プロトコルモデル

本稿では，プロトコル機械を有限個の非負整数値レジスタを持つ拡張有限状態機械でモデル化し，二つのプロトコル機械を接続する双方向の通信路を長さに制限のない FIFO でモデル化したプロトコルを取り扱う．形式的には以下のように定義する．

定義 1 プロトコル機械を 4 字組 (S, Σ, T, SI) で定義する．

(M1) $S = \langle SF, r \rangle$: プロトコル機械の状態集合を定義する 2 字組．SFは有限制御部の状態の有限集合，rは非負整数値を保持するレジスタの数を表す．プロトコル機械の状態集合は $SF \times \mathcal{N}^r$ となる．但し，$\mathcal{N}$は非負整数の集合を表す．

(M2) $\Sigma = \Sigma_- \cup \Sigma_+$: メッセージ型の有限集合．$\Sigma_-, \Sigma_+$はそれぞれ送信メッセージ型，受信メッセージ型の有限集合を表し，Σ_-とΣ_+は互いに素とする．各メッセージはメッセージ型に加えて非負整数値パラメータを一つ持ち，パラメータ $p \in \mathcal{N}$ を持つ型 $d \in \Sigma$ のメッセージを $\langle d, p \rangle$ と書く．

(M3) T: アクションの有限集合．アクションは 5 字組 (v, d, w, C, R) で定義される．$v \in SF, d \in \Sigma, w \in SF$．$C$は状態遷移前のプロトコル機械のレジスタ値 $p_1, p_2, \ldots, p_r$と送信または受信メッセージ $\langle d, p \rangle$ のパラメータ値 p に関する連立線形不等式であり，遷移条件と呼ぶ．Rは状態遷移前のプロトコル機械のレジスタ値 $p_1, p_2, \ldots, p_r$ と送信または受信メッセージのパラメータ値 p から状態遷移後のプロトコル機械のレジスタ値 $p_1, p_2, \ldots, p_r$を定める $\mathcal{N}^{r+1}$から $\mathcal{N}^r$ への線形演算により定義される部分関数であり，レジスタ更新関数と呼ぶ．本稿では，解析の容易さを考慮してアクションの定義の中で，遷移条件を構成する各不等式を “$x - y \leq c$” という形 (差分制約とよぶ) に限定し，レジスタ更新関数を “$x \leftarrow y + c$” という形に限定する．ここで，x, yはレジスタまたは送受信するメッッセージのパラメータ値である．このような制限をおいても，一連番号やタイマを用いたプロトコルを十分モデル化することができる．Tにより $\langle SF \times \mathcal{N}^r \rangle \times \langle \Sigma \times \mathcal{N} \rangle$ から $SF \times \mathcal{N}^r$ への非決定性状態遷移関数 δ は以下のように定まる．

$$\delta(\langle v, p_1, \ldots, p_r \rangle, \langle d, p \rangle) = \{\langle w, R(p_1, \ldots, p_r, p) \rangle \mid (v, d, w, C, R) \in T \text{かつ } p_1, \ldots, p_r, p \text{ は } C \text{を満たす}\}$$

(M4) $SI \subseteq SF \times \mathcal{N}^r$: 初期状態の集合． □

二つのプロトコル機械 $PM_A = (\langle SF_A, r_A\rangle, \Sigma_A, T_A, SI_A)$, $PM_B = (\langle SF_B, r_B\rangle, \Sigma_B, T_B, SI_B)$ について，$\Sigma_{B-} = \Sigma_{A+}$ (Σ_{BA}と書く)，$\Sigma_{A-} = \Sigma_{B+}$ (Σ_{AB}と書く) であるとき，2字組$\Pi = (PM_A, PM_B)$をプロトコルと呼ぶ．4字組 $gs = \langle s_A, s_B, u_{BA}, u_{AB}\rangle \in \langle SF_A \times \mathcal{N}^{r_A}, SF_B \times \mathcal{N}^{r_B}, \langle \Sigma_{BA} \times \mathcal{N}\rangle^*, \langle \Sigma_{AB} \times \mathcal{N}\rangle^*\rangle$ をプロトコルΠの系の状態と呼ぶ．ここで，s_A，s_Bはそれぞれgsにおけるプロトコル機械PM_A，PM_Bの状態を表し，u_{BA}，u_{AB}はそれぞれgsにおけるPM_BからPM_Aへの通信路，PM_AからPM_Bへの通信路上のメッセージ系列を表している．$s_A \in SI_A$，$s_B \in SI_B$であるとき，状態$\langle s_A, s_B, \varepsilon, \varepsilon\rangle$ (εは空系列を表す) をΠの系の初期状態と呼ぶ．

以下では，混乱のない限り系の状態を単に状態と呼ぶ．

定義 2 プロトコル $\Pi = (PM_A, PM_B)$ の状態，$gs = \langle s_A, s_B, x, y\rangle$ および $gs' = \langle s'_A, s'_B, x', y'\rangle$ に対して，ある $d \in \Sigma_{AB} \cup \Sigma_{BA}$, $p \in \mathcal{N}$ が存在して以下のいずれかが成立するとき，gs から gs' に遷移可能であるといい，$gs \rightarrow gs'$ と書く．

$s'_A \in \delta_A(s_A, \langle d, p\rangle)$, $s'_B = s_B$, $x' = x$, $y' = y \cdot \langle d, p\rangle$

$s'_A \in \delta_A(s_A, \langle d, p\rangle)$, $s'_B = s_B$, $\langle d, p\rangle \cdot x' = x$, $y' = y$

$s'_A = s_A$, $s'_B \in \delta_B(s_B, \langle d, p\rangle)$, $x' = x \cdot \langle d, p\rangle$, $y' = y$

$s'_A = s_A$, $s'_B \in \delta_B(s_B, \langle d, p\rangle)$, $x' = x$, $\langle d, p\rangle \cdot y' = y$

関係 "$\rightarrow$" の反射推移閉包を "$\stackrel{*}{\rightarrow}$" と書く．$gs \stackrel{*}{\rightarrow} gs'$のとき，$gs$ から gs'に到達可能であるという．プロトコルΠの初期状態から到達可能である状態を可達状態と呼ぶ．また，その集合をΠの可達集合と呼び，RS_Πと書く． □

2.2 安全性の検証法

プロトコル Π の任意の可達状態において論理式 F が成立するとき，Fは Π の不変式であるという．論理式 P を満たす Π の状態のみからなる集合を $GS(P)$ と書く．文献[3]で提案している検証法では，

(a) 検証者の記述した論理式 F が不変式，すなわち $GS(F) \supseteq RS_\Pi$ であること，

(b) $GS(F)$ が安全でない状態を含まないこと，

を証明する．(a) と (b) が示されれば，Πは安全であると結論できる．

2.2.1 不変式の記述

検証者はプロトコル$\Pi = (PM_A, PM_B)$ の初期状態から到達可能であると想定している状態の集合を，それぞれの状態における各プロトコル機械の状態，各通信路上のメッセージ系列により，いくつかの互いに素な部分集合に分割する(以下ではその分割数を n とする)．それぞれの状態集合に対して，その集合中のすべての状態で成立する条件を以下の(AF1)-(AF4) の 4 種類の原子式の積項 P_i $(i = 1, 2, \ldots, n)$ として記述し，$F = P_1 \vee P_2 \vee \cdots \vee P_n$とする．

(AF1) $(sf_A(\in SF_A), sf_B(\in SF_B))$: PM_A, PM_Bの有限制御部の値がそれぞれ sf_A，sf_Bであることを表す．

(AF2) 通信路上のメッセージ系列の型系列が満たすべき性質を検証者が定義した述語を用いて記述した式．例えば，AF2 型原子式 "$u_{AB} \in \mathcal{L}(\mathrm{MIP}^+)$" は，$PM_A$から$PM_B$への通信路上の型系列が正規表現 "$\mathrm{MIP}^+$" の表す系列集合の要素である，すなわち 1 個以上の MIP から成る系列であることを表す．ここで，$\mathcal{L}(R)$ は正規表現 Rの表す系列集合である．

(AF3) 通信路上のメッセージ系列のパラメータ系列が満たすべき性質を検証者が定義した述語を用いて記述した式．例えば，$\mathrm{step1}(u_{AB})$ は，PM_Aから PM_Bへの通信路上のパラメータ系列が述語 step1 として定義された性質を満たしていることを表す．ここで，$\mathrm{step1}(\alpha)$ はメッセージ系列α のパラメータ系列が増分 1 の増加列であることを表す検証者の定義した述語である．

(AF4) プロトコル機械のレジスタ値，および通信路上のメッセージ系列の特定位置のメッセージのパラメータ値に関する線形等式または線形不等式．例えば，$VM_A = \mathrm{firstp}(u_{BA}) + 1$ は，PM_Aのレジスタ VM_A の値が PM_Bから PM_Aへの通信路上のメッセージ系列 u_{BA} の先頭メッセージのパラメータ値に 1 を加えたものに等しいことを表す．

2.2.2 安全性の検証手続き

検証者が記述した論理式 $F = P_1 \vee P_2 \vee \cdots \vee P_n$ がプロトコルΠの不変式であることを示すことができ，さらに不変式 Fを満たす状態集合 $GS(F)$ がデッドロック状態および未定義受信状態を含まないことを示すことができれば，Π が安全であると結論できる．

Fが不変式であることは以下のような系の状態遷移系列に関する構造的帰納法により証明する．

[初期段階] Πの各初期状態において，ある $P_i(1 \le i \le n)$ が成立することを示す．

[帰納段階] 各 $P_i(1 \le i \le n)$ について，P_iを満たす任意の状態から遷移可能な任意の状態で論理式 Fが成立することを以下のようにして示す．

(i) P_iを満たすある状態で実行可能なアクション $t \in T_A \cup T_B$をすべて求める．

(ii) (i) で求めたアクション t ごとに，t が実行可能な任意の状態 $gs \in GS(P_i)$ から t による状態遷移により

遷移可能な任意の状態 gs' において，少なくとも一つの積項 P_j が成立する，すなわち P_j 中のすべての原子式が成立することを示す．

帰納段階の (ii) において，P_j 中の AF1 型原子式が成立するかどうかはアクション t より直ちに決定できる．AF2 型原子式の成立は二つの正規集合の包含関係の判定により決定できる．AF4 型原子式の成立は連立線形不等式を解くことにより判定できる．AF3 型原子式の成立については，定義述語の性質を検証者が書き換え規則の形で与え，その書き換え規則と遷移前後のレジスタ間の関係を表す書き換え規則からなる項書換え系上で，一定の書換え戦略の下で AF3 型原子式が論理値 "true" に書きかわるかどうかを判定する．このため，AF3 型原子式のみ必要十分条件ではなく十分条件の判定となっている．

F が不変式であることの上記証明法の帰納段階の証明，および $GS(F)$ がデッドロック状態または未定義受信状態を含まないことの証明を自動化する検証システムが試作されている[3]．

3 部分到達可能性解析による積項の自動生成

一方のプロトコル機械のメッセージの送信に同期して，他方のプロトコル機械がそのメッセージの受信を行なう通信形態を同期型通信と呼ぶ．ここで考える部分到達可能性解析とは，初期状態から同期型通信によって到達可能な状態集合を求め，次にそれらの状態からそれぞれのプロトコル機械のメッセージの送信のみによって到達可能な状態集合を求めることである．同期通信によって到達可能な状態集合を求める場合，プロトコル機械間の通信路を考慮する必要がない．また，ある状態からメッセージの送信のみによって到達可能な状態を求める場合，個々のプロトコル機械とそのプロトコル機械から出ている通信路のみを考慮すればよい．すなわち，部分到達可能性解析を行なう場合，一般の到達可能性解析の場合と比べて考慮すべき条件が少なく，解析が容易となる．

3.1 同期通信により到達可能な状態の積項の生成

同期通信により到達可能な状態をプロトコル機械の各アクションが実行可能かどうかで部分集合に分類し，各部分集合毎に積項を生成することを考える．すなわち生成する積項 p の条件として以下の条件を考える．

[積項条件] p 中の AF 1 型原子式が $\langle u_A, u_B\rangle$ であるとする．同じ型のメッセージの送受信をする各アクション対 $T'_A = \langle u_A, d, v_A, C_A, R_A\rangle$ $T'_B = \langle u_B, d, v_B, C_B, R_B\rangle$ に対して，p を満たすすべての状態がアクション対 T_A, T_B が実行可能（あるパラメータ値について C'_A, C'_B を満たす）であるか，そうでないかのどちらかである． □

プロトコル $\Pi = (PM_A, PM_B)$ において系の初期状態はその満たすべき条件が AF1 型原子式と AF4 型原子式からなる上記の積項条件を満たす一つの積項 p_{init} によって指定されているものとする．このとき，同期通信により到達可能な状態に対する積項の集合 P を以下の手順で生成する．同期通信では通信路にメッセージが滞留しないので，生成される積項は AF1，AF4 型原子式のみからなる．

1. $P \leftarrow p_{init}$;
2. 以下の操作を新しい積項を Pに加えることができなくなるまで続ける

   P中の積項 $p = \langle u_A, u_B\rangle \wedge \Phi$ について，p を満たす状態で実行可能な同じ型のメッセージの送受信をするアクションの対 $T_A = \langle u_A, d, v_A, C_A, R_A\rangle$, $T_B = \langle u_B, d, v_B, C_B, R_B\rangle$ を求め，各 $\langle T_A, T_B\rangle$ について以下の処理を行なう．

   (a) 遷移後の状態に関する積項を求める．

   遷移後の状態で，それぞれの機械の有限制御部の状態は v_A, v_B である．また，遷移後のレジスタ間の関係についての条件は Φ, C_A, C_B, R_A, R_B から遷移前のレジスタ値間の関係と遷移前後のレジスタ値と送受されるメッセージのパラメータ値の関係からなる連立不等式を構成し，構成した連立不等式から遷移前のレジスタ値と送受されるメッセージのパラメータ値を消去することにより，求めることができる．これを Φ' とする．すなわち，遷移後の状態では積項 $p' = \langle v_A, v_B\rangle \wedge \Phi'$ が成り立つ．

   (b) 状態集合の分割．

   p' が積項条件をみたすとは限らないので，$\Phi' = \Phi'_1 \vee \ldots \vee \Phi'_n$ となるような積項条件を満たす $p'_1 = \langle v_A, v_B\rangle \wedge \Phi'_1, \ldots p'_n = \langle v_A, v_B\rangle \wedge \Phi'_n$ を求め，$P = P \cup \{p'_1, \ldots p'_n\}$ とする．

   (c) 重複する積項の除去

   P中の積項 p, q について $GS(p) \subseteq GS(q)$ なら，p を P から除去する．

   (d) 積項の併合

   P中の積項 p, q を比較し，$p \vee q = p'$ となる積項条件を満たす p' が存在するならば p, q を P から除去し，p' を P に加える．

   (e) 積項の拡大

   P中の積項 p_0 から始まる下記の条件をみたす積項の無限系列 $p_0, p_1, \ldots$ と積項条件をみたす p'が検出された場合，p_0 を P から除き，p'を P にいれる．

- 任意の $k \geq 0$ について，p_kを満たす状態から1回の同期通信により p_{k+1}を満たす状態へ遷移可能
- $p' = p_0 \vee p_1 \vee \ldots$

例えば p として $p' \wedge (a = b)$ という積項があり，$k \geq 0$ のとき $p' \wedge (a = b + k)$ を満たす状態から $p' \wedge (a = b + k + 1)$ を満たす状態へ1回の同期通信により遷移できる場合，任意の $k \geq 0$ について，$p' \wedge (a = b + k)$ なる状態へ同期通信により到達できる．このような場合 p を P から除き，$p \wedge (a \geq b)$ という積項を P にいれる．

本稿で扱うプロトコル機械では遷移条件を差分制約に限定し，レジスタ代入式も "$x \leftarrow y + c$" という形のものに限定しているので積項中のAF4型原子式も差分制約に限られる．このため，同期通信により到達可能な状態集合が，あるレジスタの値がいくらでも大きくなりうるなどの理由で無限集合となる場合でも，レジスタ間の差分が有界であれば，生成される積項は有限個となり，生成の手続きは停止する．レジスタの差分が非有界である場合でも，積項の拡大がうまくはたらけば，有限個の積項の生成で停止する場合がある．また，考慮する不等式がすべて差分制約に限られるため，上記の処理で必要となる整数連立不等式の解の存在の判定が，不等式の数を l，変数の数を m として $O(l \cdot m)$ で解くことができる[6]．

3.2 メッセージの送信により到達可能な状態の積項の生成

同期通信により到達可能な状態から送信のみによって到達可能な状態に関する積項を生成するため，まず，個々のプロトコル機械ごとに，プロトコル機械の状態をいくつかの部分集合にわけ，それらの状態集合間の送信動作による遷移関係を明らかにする．次に，やはり個々のプロトコル機械ごとに一連の送信遷移により送出されるメッセージに関する条件の抽出を行なう．そして，それらをもとに，3.1で得られた同期通信により到達可能な状態から，個々のプロトコル機械が送信のみを行なった場合に到達可能な条件を表す積項を構成する．

3.2.1 各プロトコル機械毎の状態遷移グラフの構成

まず，各プロトコル機械 PM_X ごとの送信動作による遷移関係を表す以下のようなラベルつきの多重有向グラフ $RG_X = (V_X, E_X)$ $(X \in \{A, B\})$ を構成する．

- $V_X = \{p_{i,X} \mid p_{i,X}$は3.1で得られた積項 p_iについて，プロトコル機械 PM_Xのみに関する条件を抽出した積項$\}$
- $E_X = \{(p_{i,X}, p_{j,X}) \mid p_{i,X}$を満たす状態から $p_{j,X}$を満たす状態へ送信により遷移可能$\}$

ただし，$p_{i,X}$ を満たす状態から $p_{j,X}$ を満たす状態へ遷移できる複数のアクションを持つ場合はアクションごとに一つの辺をもつ．

各辺はアクションの定義から求めた以下のラベルを持つ．

1. その送信遷移で送出されるメッセージの型．
2. その送信遷移で送出されるメッセージのパラメータ値と送信前の状態のレジスタ値，送信後の状態のレジスタ値の関係を表す連立不等式．

3.2.2 積項の生成

RG_A,RG_B が単純閉路以外の閉路を含まない場合，3.1で求めた同期通信により到達可能な状態についての積項と RG_A, RG_B をもとに積項の生成を行なう．RG_A,RG_Bが単純閉路以外の閉路を含む場合は，検証者がそれらをもとに不変式を記述することになる．

3.1で求めた積項 p_i について，RG_Aで $p_{i,A}$から $p_{j,A}$に RG_Bで $p_{i,B}$から $p_{k,B}$にそれぞれ到達可能であるとき，RG_Aで $p_{i,A}$から $p_{j,A}$に RG_Bで $p_{i,B}$から $p_{k,B}$に至る，閉路を含まない経路ごとに，以下の原子式からなる積項を生成する．

AF1: $p_{j,A}$, $p_{k,B}$で指定された各プロトコル機械の有限制御部の状態を指定．

AF2: それぞれのプロトコル機械ごとに着目している経路とその経路上の頂点を含む単純閉路からなるグラフを $p_{i,A}$ (または $p_{i,B}$) を始状態，$p_{j,A}$ (または $p_{k,B}$) を受理状態とし，辺上のメッセージ型に関するラベルをその辺の表す遷移での入力とする有限オートマトンとしたときの受理言語を正規表現として求め，それらの対を原子式とする．

AF3: それぞれのプロトコル機械ごとに着目している経路上に単純閉路がある場合，閉路上の遷移で送信されるメッセージのパラメータ値と送信前後のレジスタ値の関係に共通の性質がある場合，その閉路上で送信されたメッセージのパラメータに関する述語を生成する．この場合，検出した共通の性質を検証者に提示し，検証者は述語に適当な名前を与えた上，その述語に関する性質を書き換え規則の形で与える

AF4: p_i 中のAF4型原子式，着目している経路上の辺のラベル中の送信されたメッセージのパラメータ値，遷移前後のレジスタ間の関係を表す不等式からなる連立不等式を構成し，送信されたメッセージのパラメータ値，最後の遷移をおこなったあとのレジスタの値を表すものを除く変数を消去して得られた各不等式を原子式とする．

もちろん，以上のようにして生成した積項では，p_i を満たす状態から着目した経路をへて到達可能な状態に関する条件をかならずしも十分抽出できていない場合が考えられる．このような場合は，人間の検証者がいくつかの原子式を積項に追加してやることが必要になる．

4 例プロトコルでの実験結果

OSIセションプロトコルの一部を抽出したプロトコルを例にとりあげ，提案した手法により，その不変式の積項をどの程度自動生成できるかの実験を行なった．

4.1 積項自動生成システム

前述の手法に基づいて不変式の一部の積項を自動生成するシステムが試作した．作成したシステムは二つのプロトコル機械の定義，初期状態に関する積項を入力として，3. で述べた手法により求められた積項からなる積和形の論理式を出力する．ただし，積項の拡大については簡単な十分条件の判定のみを行なっている．また，生成した積項を満たす状態が，デッドロック状態や未定義受信状態を含んでいる場合，これを検出することができる．

得られた論理式 F に対してすでに作成している安全性の検証システム[3]を用いて F が不変式であるための十分条件の判定が行なえる．十分条件が成り立たない場合，検証システムは F に不足している到達可能な状態に関する情報を出力してくるので，検証者がそれを参考にいくつかの積項について条件を加えたり，新たな積項を補足した論理式 F'を構成し，再び検証システムにかけることになる．作成したシステムは C, yacc, lex で記述されており，全体で約 18,000 行で，このうち 12,000 行については，安全性，生存性の検証システムから流用することができた．

4.2 例プロトコル

例プロトコルとして文献 [7] で規定される OSI セションプロトコルのカーネル，全二重，大同期，小同期機能単位のデータ転送フェーズから抽出したプロトコル $\Pi_{scs} = (PM_A, PM_B)$ をとりあげた．プロトコルの抽出にあたり，大同期点および小同期点設定用のトークンを一つのトークンで共用するなどの簡単化を施している．

Π_{scs}の二つのプロトコル機械 PM_A と PM_B は初期状態が異なる，すなわち初期状態でトークンを持っているか持っていないかが異なる点を除いては同型である．プロトコル機械の有限制御部の状態数は 10，レジスタは VM と VA の 2 個，送受信するメッセージ型は 12 種類，アクションの数は 22 である．プロトコル機械のアクションの遷移条件中の線形不等式はすべて差分制約である．

4.3 例プロトコルの検証実験

UNIX ワークステーション (NWS-5000, 64MB) 上で，不変式自動生成システムにΠ_{scs} とその初期状態を指定する積項を入力し，積項の自動生成を行なった．例プロトコルでは作成した RG_A, RG_B はともに単純閉路以外の閉路を含まず，同期通信により到達可能な状態から送信のみによって到達可能な状態についての積項も生成できた．生成に要した時間は約 75 秒で，472 個の積項からなる積和形論理式 F が得られた．得られた積項のうち 27 個の積項については条件がゆるくなっており，実際に到達可能でない状態でも成立するため F 自身は不変式とはならなかった．そこで，検証者がそれらの積項にいくつかの原子式を付け加えることによって不変式 F' を得ることができた．27 個の積項について条件がゆるくなっている原因はバグによるものと考えており，現在デバグ中である．

5 まとめ

本稿では，拡張有限状態機械でモデル化されたプロトコルの安全性の検証のための不変式の半自動生成について述べた．提案手法は，従来検証者が行なっていた不変式の記述作業の多くの部分を自動化するものであり，現時点で，例プロトコルの安全性を示す不変式の積項中その 95%を自動的に生成することができている．今後，システムのデバグをすすめ，より実用規模のプロトコルの検証実験を行なう予定である．

参考文献

[1] CCITT: "Functional Specification and Description Language, Recommendation", Z.100(1989).

[2] ISO: "Information Processing Systems – Open Systems Interconnection – Estelle – A Formal Description Technique Based on an Extended State Transition Model", ISO/DIS 9074 (1987).

[3] Higuchi M. et al. : "A Verification Method via Invariant for Communication Protocols Modeled as Extended Communicating Finite-State Machines", *IEICE Trans. Commun.* E-76B, 11, pp.1363–1372 (1993-11).

[4] 須川他 : "拡張有限状態機械でモデル化された通信プロトコルの生存性の検証法", 信学論 B-I, J78-B-I, 1, pp.17–28(1995-01).

[5] Higuchi M. et al. : "An Experiment on Verfying OSI Session Protocol – Decomposition into Subprotocols –", Proc. 9th Int'l Conf. on Information Networking, pp.231–236 (1994-12).

[6] Cormen T. et al. "Introduction to Algorithms", The MIT Press, pp.539–543 (1990).

[7] ISO: "Basic Connection Oriented Session Protocol Specification", ISO 8327.