人工知能　85－11
(1992. 11. 20)

# 非決定性プロセスの様相論理式からの多項式時間学習

## Shapiroのモデル推論のプロセスへの適用

小池賢一　　岩沼宏治
山梨大学工学部電子情報工学科

あらまし

本研究では、未知のプロセスに強等価なプロセスを多項式時間で学習するアルゴリズムを与える．学習アルゴリズムは，様相論理式を用いて教師に質問を行う。そして、教師の答えにしたがってプロセスの修正を行う、このときShapiroがモデル推論で用いた矛盾点追跡アルゴリズムを変更して用いる．プロセスはラベル付き遷移システムによって定義する．ラベル付き遷移システム上の等価性と，様相論理式との関係は、代数的に記述されたプロセス上の等価性と，様相論理式との関係から導かれる．

和文キーワード　非決定性プロセス，強等価，多項式時間学習，様相論理式，モデル推論，矛盾点追跡アルゴリズム

# Learning Nondeterministic Process from Modal Formulas in Polynomial Time

## An application of Shapiro's Model Inference to process

Kenichi Koike†　　Kouji Iwanuma
Dept. of Electrical Eng. and Computer Science, Faculty of Eng., Yamanashi Univ.
4-3-11 Takeda, Koufu, Yamanashi 400, Japan
† e-mail　koike@kiritsubo.esi.yamanashi.ac.jp

Abstract

In this paper, we present a polynomial time algorithm for learning processes that are strongly equivalent to unknown processes. This algorithm asks queries to teacher using modal formulas. In the next place this algorithm correct a process according to the reply using a modified contradiction backtracing algorithm of Shapiro's Model Inference. The processes are defined as labeled transition system. The relation between equivalentness of labeled transition system and modal formulas is inferred from the relation between processes defined by algebra and modal formulas.

英文 key words Nondeterministic process, strongly equivalent, polynomial time learning, modal formula, model inference

## 1 はじめに

プログラムの作成の支援を目的として，プログラムの自動合成についての研究がなされている．

ここでは特に，自動合成アルゴリズムとプログラム作成の依頼者が対話的に作業を進めていくモデルを想定する．つまり，自動合成アルゴリズムは，仕様に基づいてプログラムを作成し，もし仕様の情報が不十分ならば作成依頼者に質問を行う．一方，依頼者はプログラムに不満ならば，新たに仕様を追加する．これを繰り返して依頼者は最終的に目的のプログラムを得る．

以下では，プログラムを形式化されたプロセスと定め，仕様として様相論理式を用いる．自動合成アルゴリズムは，依頼者の考えているプロセスと等価なプロセスを多項式時間で学習し出力する．

## 2 プロセスの代数的記述

### 2.1 プロセス

初めに，プロセスの代数的記述とプロセス論理 (process logic) の定義を述べ，両者間の関係について既存の結果を示す．

プロセスの動作の基本的な単位をアクション (action) と呼び，その可算無限集合を $Act$ で表す．プロセス変数 (process variable) を $x, y, z$ などで表し，任意のプロセスを意味する，プロセス変数は可算無限個とする．

[定義 2.1] プロセス (process) を，次のように帰納的に定義する．

(1) 0 (inaction) はプロセスである．

(2) プロセス変数 $x$ はプロセスである．

(3) $E$ がプロセスのとき $a.E$( $a \in Act$ によるプレフィクス (prefix)) もプロセスである．

(4) $E_1, E_2$ がプロセスのとき，その和 (summation) $E_1 + E_2$ もプロセスである．

(5) $x_1, \ldots, x_n$ をプロセス変数，$E_1, \ldots, E_n$ をプロセスとすると，再帰 (recursion) $\mu.x_j\{x_i = Ei \mid 1 \leq i \leq n\}$ はプロセスである．ただし，$1 \leq j \leq n$ □

[定義 2.2] プロセスの遷移関係を次の規則によって与える．

(1) $a.p \xrightarrow{a} p$

(2) $p \xrightarrow{a} p' \Longrightarrow p + q \xrightarrow{a} p'$,
$q \xrightarrow{a} q' \Longrightarrow p + q \xrightarrow{a} q'$

(3) $E_j\{(\mu\tilde{x}.\{x_i = E_i \mid 1 \leq i \leq n\})/\tilde{x}\} \xrightarrow{a} E'_j \Longrightarrow$
$\mu x_j.\{x_i = E_i \mid 1 \leq i \leq n\} \xrightarrow{a} E'_j$ □

$x_j.\{x_i = E_i \mid 1 \leq i \leq n\}$ において $x_i$ は束縛変数と呼ばれる．束縛変数でないとき自由変数と呼ばれる．(3) の意味は $E_j$ の自由変数 $x_i$ に $\mu x_i.\{x_i = E_i \mid 1 \leq i \leq n\}$ を同時に代入したプロセス $E_j\{(\mu\tilde{x}.\{x_i = E_i \mid 1 \leq i \leq n\})/\tilde{x}\}$ から $a$ による遷移があれば，$\mu x_j.\{x_i = E_i \mid 1 \leq i \leq n\}$ からも $a$ による遷移があることを表す．

### 2.2 プロセスの等価性

[定義 2.3] プロセス上の関係 $R$ が強双模倣 (strong bisimulation) であるとは，$(p, q) \in R$ ならば任意の $a \in Act$ について

(1) $p \xrightarrow{a} p' \Longrightarrow$ ある $q'$ が存在して $q \xrightarrow{a} q', (p', q') \in R$

(2) $q \xrightarrow{a} q' \Longrightarrow$ ある $p'$ が存在して $p \xrightarrow{a} p', (p', q') \in R$

が成り立つことである． □

[定義 2.4] プロセス $p, q$ が強等価 (strongly equivalent) $p \sim q$ ，であるとは，ある強双模倣 $R$ が存在して $(p, q) \in R$ となることである，すなわち $\sim = \cup\{R \mid R : 強双模倣\}$ □

### 2.3 プロセス論理

プロセスの特性を表すのに論理式を用いる．そして，論理式が成り立つか否かでプロセスを区別する．

[定義 2.5] 論理式を次のように帰納的に定義する．

(1) $T$ は論理式

(2) $A, A_i$ が論理式ならば $\neg A, \bigwedge_{i \in I} A_i$ も論理式，ただし

$I$ はインデックス集合

(3) $A$ が論理式ならば $\langle a \rangle A$ も論理式. ただし $a \in Act$□

[定義 2.6] 論理式の深さを次のように帰納的に定義する.

(1) $depth(T) = 0$

(2) $depth(\neg A) = depth(A) + 1$

(3) $depth(\bigwedge_{i \in I} A_i) = max\{depth(A_i) \mid i \in I\}$

(4) $depth(\langle a \rangle A) = depth(A) + 1$ □

[定義 2.7] 論理式の充足性を次のように定義する.

(1) すべてのプロセス $p$ について $p \models T$

(2) $p \not\models A$ のとき $p \models \neg A$

(3) すべての $A_{i \in I}$ について $p \models A_i$ のとき $p \models \bigwedge_{i \in I} A_i$

(4) あるプロセス $q$ が存在して $p \xrightarrow{a} q$, $q \models A$ のとき $p \models \langle a \rangle A$ □

### 2.4 等価性とプロセス論理

等価性とプロセス論理の関係について次のことがわかっている. [2]

[定理 2.1]$p, q$ をイメージ有限なプロセスとすると, 次が成り立つ

$$p \sim q \iff \text{すべての論理式 } A \text{ について } p \models A \Leftrightarrow q \models A$$ □

## 3 ラベル付き遷移システム

### 3.1 プロセスのラベル付き遷移システムによる形式化

プロセスのアクションをラベルとするラベル付き遷移システムを定義する. 以下では, 学習対象としてラベル付き遷移システムを採用する.

[定義 3.1] ラベル付き遷移システム (labeled transition system) を 4 項から成る $M = (S, \Sigma, \rightarrow, s_0)$ と定義する. ここで,

$S$ : 状態の有限集合
$\Sigma$ : アルファベットの有限集合
$\rightarrow$ : $\{\xrightarrow{a} \mid a \in \Sigma\}$ ただし, $\xrightarrow{a} \subseteq S \times S$
$s_0$ : 初期状態 □

状態間の遷移関係において,$(s, t) \in \xrightarrow{a}$ ならば, 状態 $s$ からアクション $a$ によって状態 $t$ に遷移可能なことを表している. また, 状態 $s$ でアクション $a$ が可能 ($s, a$ に対して $(s, t) \in \xrightarrow{a}$ となる $t$ が存在する) ならば $s \xrightarrow{a}$, 状態 $s$ でアクション $b$ が不可能 (すべての $t$ について $(s, t) \notin \xrightarrow{b}$ ) であることを $s \stackrel{b}{\not\rightarrow}$ と記述する.

ラベル付き遷移システムにおける論理式の充足性をプロセスの場合と同じように定義する.

### 3.2 ラベル付き遷移システムの等価性

[定義 3.2] ラベル付き遷移システムの状態間の関係 $R$ が強双模倣 (strong bisimulation) であるとは,$(s, t) \in R$ ならば任意の $a \in \Sigma$ について

(1) $s \xrightarrow{a} s' \implies$ ある $t'$ が存在して $t \xrightarrow{a} t', (s', t') \in R$

(2) $t \xrightarrow{a} t' \implies$ ある $s'$ が存在して $s \xrightarrow{a} s', (s', t') \in R$

が成り立つことである. □

[定義 3.3] 状態 $s, t$ が強等価 (strongly equivalent)$s \sim t$ であるとは, ある強双模倣 $R$ が存在して $(s, t) \in R$ となることである, すなわち $\sim = \cup\{R \mid R : \text{強双模倣}\}$ □

[定義 3.4] ラベル付き遷移システム $M_1 = (S_1, \Sigma_1, \rightarrow_1, s_{10}), M_2 = (S_2, \Sigma_2, \rightarrow_2, s_{20})$ が強等価 (strongly equivalent)
$M_1 \sim M_2$ であるとは, 各々の初期状態 $s_{10}, s_{20}$ が強等価のときである. □

### 3.3 遷移システムにおける論理式の充足性

[定義 3.5] 遷移システムの各状態における論理式の充足性を次のように定義する.

(1) すべての状態 $s$ について $s \models T$

(2) $s \not\models A$ のとき $s \models \neg A$

(3) すべての $A_{i \in I}$ について $s \models A_i$ のとき $s \models \bigwedge_{i \in I} A_i$

(4) ある状態 $t$ が存在して $s \xrightarrow{a} t, t \models A$ のとき $s \models \langle a \rangle A$ □

ラベル付き遷移システムは, 代数的に記述されたプロセスの部分クラスになっている. 実際, 代数的記述に書き換えることができる.

[定義 3.6] $M=(S,\Sigma,\rightarrow,s_0)$ をラベル付き遷移システムとする、このとき $M$ に対して，代数的に記述されたプロセス $P(M)$ を次のように定義する．プロセス変数の集合を $X$ として，

$$X = S$$
$$Act = \Sigma$$
$$P(M) = \mu s_0.\{s_i = \sum a_j s_k \mid s_i \xrightarrow{a_j} s_k, 1 \le i \le n\}$$

□

[命題 3.1] $M=(S,\Sigma,\rightarrow,s_0)$ をラベル付き遷移システムとすると，すべての論理式 $A$ について $s_0 \models A \iff P(M) \models A$ が成り立つ. □

[証明] 論理式の深さに関する帰納法で証明する. $depth=0$ のとき $A=T$ より $s_0 \models T, P(M) \models A$ よって成り立つ. $depth=k+1$ のとき

(1) $A=\neg A'$ の場合，帰納法の仮定から $s_0 \models A' \iff P(M) \models A'$ が成り立つ，さらに充足性の定義から $s_0 \models A \iff P(M) \models A$

(2) $A=\wedge_{i \in nI} A_i$ の場合,$s_0 \models A_i \iff P(M) \models A_i (i \in I)$ よって $s_0 \models A \iff P(M) \models A$

(3) $A=\langle a \rangle A'$ の場合、$s_0 \not\xrightarrow{a}$ ならば $s_0 \not\models \langle a \rangle A'$ であり $P(M)$ の定義から $s_0 = \sum a_j s_k (a_j \neq a)$ よって $P(M) \not\models \langle a \rangle A'$，一方 $s_0 \xrightarrow{a}$ のときは，状態 $s_n (s_0 \xrightarrow{a} s_n)$ について帰納法の仮定から次が成り立つ $s_n \models A \iff P(M(s_n)) \models A$ (ただし $M(s_n)$ とは $M$ の初期状態を $s_0$ から $s_n$ に変更したラベル付き遷移システム，初期状態以外は全く同じ). また，状態 $s_n$ について $s_0 \xrightarrow{a} s_n$ が成り立つならば $P(M)$ の定義から $P(M) \xrightarrow{a} P(M(s_n))$ よって $s_0 \models A \iff P(M) \models A$ □

[命題 3.2] $M_1=(S_1,\Sigma_1,\rightarrow_1,s_{10}), M_2=(S_2,\Sigma_2,\rightarrow_2,s_{20})$ をラベル付き遷移システムとする，このとき

$$M_1 \sim M_2 \quad \iff \quad P(M_1) \sim P(M_2)$$

が成り立つ. □

[証明]( $\Longrightarrow$ ) について，
$M_1 \sim M_2$ より $s_{10} \sim s_{20}$，よってある強双模倣 $R$ が存在して $(s_{10},s_{20}) \in R$ が成り立つ. $R$ に基づいてプロセス上の関係 $R'$ を次のように定義する

$$R' = \{(P(M_1(s)), P(M_2(t))) \mid s \in S_1, t \in S_2, (s,t) \in R\}$$

このように $R'$ を定めると $(s_{10},s_{20}) \in R$ より $(P(M_1(s_{10})), P(M_2(s_{20}))) \in R'$ つまり $(P(M_1),P(M_2))$ は関係 $R'$ に含まれる. したがって、あとは $R'$ が強双模倣であることを確かめればよい. そこで $(P(M_1(s)),P(M_2(t))) \in R'$ とする，このとき任意の $a \in Act(\Sigma)$ について

(1) $P(M_1(s)) \xrightarrow{a} P(M_1(s))'$ ならば定義 3.6 から $s \xrightarrow{a} s', P(M_1(s')) = P(M_1(s))'$ となる状態 $s'$ が存在する. また $(P(M_1(s)),P(M_2(t))) \in R'$ と $R'$ の定義から $(s,t) \in R$ よってある状態 $t'$ が存在して $t \xrightarrow{a} t', (s',t') \in R$ が成り立つ. $t \xrightarrow{a} t'$ と定義 3.6 から $P(M_2(t)) \xrightarrow{a} P(M_2(t'))$，また $(s',t') \in R$ から $(P(M_1(s')),P(M_2(t'))) \in R'$ したがって強双模倣の条件 (1) を満たす.

(2) $P(M_2(t)) \xrightarrow{a} P(M_2(t))'$ についても (1) と同様.

( $\Longleftarrow$ ) について
$P(M_1) \sim P(M_2)$ よりある強双模倣 $R$ が存在して $(P(M_1),P(M_2)) \in R$ が成り立つ.$R$ に基づいてラベル付き遷移システムの状態間の関係 $R'$ を次のように定義する

$$R' = \{(s,t) \mid s \in S_1, t \in S_2, (P(M_1(s)),P(M_2(t))) \in R\}$$

このように $R'$ を定めると $(P(M_1),P(M_2)) \in R$ より $(s_{10},s_{20}) \in R'$ が言える. したがって，あとは $R'$ が強双模倣であることを確かめればよい. そこで，ある状態 $s \in S_1$ と $t \in S_2$ について $(s,t) \in R'$ とする，このとき任意の $a \in \Sigma$ について

(1) $s \xrightarrow{a} s'$ ならば定義 3.6 から $P(M_1(s)) \xrightarrow{a} P(M_1(s'))$ が成り立つ. また $(s,t) \in R'$ と $R'$ の定義から $(P(M_1(s)),P(M_2(t)))$ は強双模倣 $R$ に含まれる. よって強双模倣の条件 (1) から，あるプロセス $P(M_2(t))'$ が存在して $P(M_2(t)) \xrightarrow{a} P(M_2(t))' \cdots (a)$, $(P(M_1(s')),P(M_2(t))') \in R \cdots (b)$ が成り立つ. ところで定義 3.6 から，$P(M_2(t))'$ について $P(M_2(t))' = P(M_2(t'))$ となるような状態 $t'$ が存在する. したがって $(a)$ は $P(M_2(t)) \xrightarrow{a} P(M_2(t'))$ と書き換えることができ，定義 3.6 から $t \xrightarrow{a} t'$ が成り立つ. $(b)$ は $(P(M_1(s')),P(M_2(t'))) \in R$ と書き換えることができ，$R'$ の定義から $(s',t') \in R'$ が成り立つ. これらより強双模倣の条件 (1) を満

たす.

(2) $t \xrightarrow{a} t'$ についても (1) と同様. □

定理 2.1 と命題 3.1, 命題 3.2 から次の定理が成り立つ.

[定理 3.1] $M_1 = (S_1, \Sigma_1, \rightarrow_1, s_{10})$, $M_2 = (S_2, \Sigma_2, \rightarrow_2, s_{20})$ をラベル付き遷移システムとすると $M_1 \sim M_2$ が成り立つことと, すべての論理式 $A$ について $s_{10} \models A \Longleftrightarrow s_{20} \models A$ が成り立つことは同値である. □

## 4 学習の設定

以下において, 未知のラベル付き遷移システムを $M_u$ , 学習アルゴリズムが教師に提案するラベル付き遷移システムを $M_h$ と略記する.

学習アルゴリズムは, $M_u$ について, それと強等価な遷移システムの学習を行う. 初めに学習者は $M_u$ の状態集合 $S$ と初期状態 $s_0$ さらにラベルの集合 $\Sigma$ を与えられている.

教師は未知の遷移システムに $M_u$ ついて二種類の質問に答える能力を持っている. 一つは, 仮説 $M_h$ と $M_u$ が強等価か否かを聞く S-EQUIV( $M_h$ ) である, 教師は, もし強等価ならば YES を返し, 強等価でなければ $M_u \models A$ と $M_h \not\models A$ が成り立つか, あるいはその逆が成り立つ論理式 $A$ を真偽値と共に返す, 真偽は未知の遷移システムにおける値とする. つまり, $M_u \models A$ のとき $\langle A, 真 \rangle$, $M_u \not\models A$ のとき $\langle A, 偽 \rangle$ を返す. 二つ目は, $M_u$ の状態 $s$ で論理式 $A$ が成り立つか否かを尋ねる SATISFY$(A, s)$ である, 教師は成り立つとき YES, さもなければ NO を返す.

## 5 学習アルゴリズム

(1) 最初の仮説を $M_h = (S, \Sigma, \rightarrow, s_0)$ とする. ただし $\rightarrow = \{\xrightarrow{a} | a \in \Sigma\}$, すべての $s_1, s_2 \in S, a \in \Sigma$ について $(s_1, s_2) \in \xrightarrow{a}$.

(2) S-EQUIV( $M_h$ ) を質問する.

(a) YES のとき $M_h$ を出力して停止.

(b) 反例の論理式 $\langle A, 真 (偽) \rangle$ が返されたとき, 矛盾点追跡アルゴリズムを用いて誤った遷移関係を見つけ出し取り除く. (2) を繰り返す.

## 6 矛盾点追跡アルゴリズム

未知の遷移システムと, アルゴリズムが呼ばれた時点の仮説を $M_u = (S, \Sigma, \rightarrow_u, s_0)$, $M_h = (S, \Sigma, \rightarrow_h, s_0)$ とおく. 論理式 $A$ が未知の遷移システム $M_u$ の状態 $s$ で真 (偽) であり, 同時に $M_h$ の状態 $s$ で偽 (真) のことを $\langle A, s, 真 (偽) \rangle$ と略記する. したがって反例 $\langle A, 真 (偽) \rangle$ が与えられた場合 $\langle A, s_0, 真 (偽) \rangle$ と書ける. 以下の (1)~(6) を繰り返して, 誤った遷移関係を一つ取り除く.

(1) $\langle \langle a \rangle A, s, 真 \rangle$ が成り立つとき, $M_u$ の状態 $s$ において $\langle a \rangle A$ は真であるから, ある状態 $s'$ が存在して $s \xrightarrow{a}_u s', s' \models A$ が成り立つ. この状態 $s'$ には $M_h$ でも遷移可能 $(s \xrightarrow{a}_h s')$ である. このことは $M_h$ の遷移が $M_u$ の遷移を含んでいることから言える. ところで $M_h$ の状態 $s$ において $\langle a \rangle A$ は偽であるから, 任意の状態 $t$ について $s \xrightarrow{a}_h t$ のとき $t \not\models A$ が成り立つ. したがって状態 $s'$ についても $s' \not\models A$ が成り立つ. よって状態 $s'$ における論理式 $A$ の真偽値は $M_u$ における場合と $M_h$ の場合で異なる. そこで, このような状態 $s'$ を探し出す.
$M_u$ において $s$ から遷移可能な状態 $s' : s \xrightarrow{a}_h s'$ について SATISFY$(A, s')$ を質問する.

YES のとき, 状態 $s$ において論理式 $A$ の真偽値が異なるので, $\langle A, s', 真 \rangle$ について (1)~(6) を繰り返す.

NO のとき, YES になるまで別の状態 $s'' : s \xrightarrow{a}_h s''$ について同じ質問を繰り返す.

(2) $\langle \langle a \rangle A, s, 偽 \rangle$ が成り立つとき, $M_h$ の状態 $s$ において $\langle a \rangle A$ は真であるから, ある状態 $s'$ が存在して $s \xrightarrow{a}_h s', s' \models A$ が成り立つ. この状態 $s'$ を論理式 $A$ の評価を行うことによって探し出し, SATISFY$(A, s')$ を質問する.

YES のとき, $M_u$ の状態 $s$ において $\langle a \rangle A$ は偽であるから, 任意の状態 $t$ について $s \xrightarrow{a}_u t$ のとき $t \not\models A$ が成り立つ. つまり論理式 $A$ が真になる状態には遷移できない. ところが $M_u$ において $s \xrightarrow{a}_u s'$ と仮定すると, YES から $s' \models A$ が成り立ち矛盾する. したがって $M_u$ において $s \xrightarrow{a}_u s'$ という遷移は存在しない. よって $M_h$ から取り除く. つまり $\xrightarrow{a}_h = \xrightarrow{a}_h - \{(s, s')\}$.

NO のとき, 状態 $s'$ における論理式 $A$ の真偽値が $M_u$ における場合と $M_h$ の場合で異なるので $\langle A, s', 偽 \rangle$ について (1)~(6) を繰り返す.

(3) $\langle \neg A, s, 真 \rangle$ が成り立つとき, $\langle A, s, 偽 \rangle$ について (1)~(6) を繰り返す.

(4) $\langle \neg A, s, 偽 \rangle$ が成り立つとき,$\langle A, s, 真 \rangle$ について (1)~(6) を繰り返す.

(5) $\langle \bigwedge_{i \in I} A_i, s, 真 \rangle$ が成り立つとき, $M_u$ の状態 $s$ において $\bigwedge_{i \in I} A_i$ は真であるから, 各々 $A_i$ もすべて真. 一方,$M_h$ の状態 $s$ において $\bigwedge_{i \in I} A_i$ は偽であるから, ある $A_j (j \in I)$ が存在して $M_h$ において $s \not\models A_j$ が成り立つ. よって状態 $s$ における論理式 $A_j$ の真偽値が $M_u$ における場合と $M_h$ の場合で異なる. そこで各論理式 $A_i (i \in I)$ を評価して, そのような論理式 $A_j$ を見つけ出し $\langle A_j, s, 真 \rangle$ について (1)~(6) を繰り返す.

(6) $\langle \bigwedge_{i \in I} A_i, s, 偽 \rangle$ が成り立つとき, (5) と同様に, ある論理式 $A_j$ が存在して状態 $s$ における真偽値が $M_u$ における場合と $M_h$ の場合で異なる. そこで論理式 $A_i (i \in I)$ について SATISFY$(A_i, s)$ を質問する

YES のとき,NO になるまで別の論理式 $A_i (i \in I)$ について同じ質問を繰り返す.

NO のとき, $\langle A_i, s, 偽 \rangle$ について (1)~(6) を繰り返す.

## 7 停止性, 正当性と時間計算量

矛盾点追跡アルゴリズムの実行において, 誤った遷移が見つからないまま (1)~(6) を繰り返していくと, 論理式 $T$ に行きつく.$T$ はすべての状態で成り立つ $(\forall s, s \models T)$ ので, いつか必ず (2) の一番目の条件が成り立つ. また,(2) で取り除かれる遷移関係は常に誤った遷移であるから, 正しい遷移はいつまでも含まれたままである.

したがって以上と定理 3.1 から, 学習アルゴリズムは必ず停止して, さらに, そのとき未知の遷移システムと強等価な仮説を出力する.

最初の仮説は, すべての状態からすべての状態へ遷移可能であるから, $|S|^2|\Sigma|$ 個の遷移を持つ. 矛盾点追跡アルゴリズムは, 一度呼ばれると一個の遷移を取り除くので多くても $|S|^2|\Sigma|$ 回実行されるのみである. 一回の実行時間は, 反例の論理式の大きさと仮説の遷移の数に比例する.

よって, 反例の論理式の大きさの最大を $|A|$ とすると, 学習アルゴリズムの時間計算量は $|S|$ と $|\Sigma|$ と $|A|$ の多項式に比例する.

## 8 おわりに

初めに状態の集合を与え, 各状態における論理式の充足性について教師が答えると仮定したとき, プロセスを多項式時間で学習可能であることが示された. 今後の課題として

状態の集合を与えない場合の学習

他の等価性における学習

などが挙げられる.

## 参考文献


[1] Milner,A.J.R.G.: *A calculus of communicating systems*, Lecture Notes in Computer Science, Vol. 92, Springer-Verlag, 1980.

[2] Milner,A.J.R.G.: *Communication and concurrency*, Prentice-Hall, 1989.

[3] 木村成伴, 富樫敦, 野口正一: "様相論理式による基本プロセスの帰納推論アルゴリズム", 情報研報 91-AI-78-5(1991-9).

[4] D.Angluin, Learning regular sets from queries and counter-examples, *Inform.and Comput.* **75**(1987)87-106.

[5] E.Y.Shapiro: "知識の帰納的推論" 共立出版,1986.

[6] 二木厚吉, 富樫敦: "形式仕様とプロセス代数",**bit**, Vol. 23, No. 11, pp. 79-92, 1991.

[7] 富樫敦: "プロセス代数等価性 (前編)",**bit**, Vol. 23, No. 12, pp. 79-92, 1991.

[8] 富樫敦: "プロセス代数等価性 (中編)",**bit**, Vol. 24, No. 1, pp. 84-95, 1991.