ICOT Technical Memorandom: TM-1161

TM-1161

Primitive Prolog の正データからの多項式時間推論可能性

石坂　裕毅、有村　博紀、
篠原　武（九州工業大学）

February, 1992

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03)3456-3191～5
Telex ICOT J32964

Institute for New Generation Computer Technology

Primitive Prolog の正データからの多項式時間推論可能性

Polynomial Time Inferability of Primitive Prolog from Positive Data

石坂裕毅 †　　有村博紀 ‡　　篠原武 ‡

Hiroki Ishizaka　　Hiroki Arimura　　Takeshi Shinohara

† 新世代コンピュータ技術開発機構 (ICOT), ‡ 九州工業大学 (Kyushu Institute of Technology)

Abstract: We consider the polynomial time inferability of primitive Prolog from positive data. Primitive Prolog is a proper subclass of linear Prolog that is known to be inferable from only positive data. In this paper, we discuss on the polynomial time inferability of the subclass. We give a consistent and conservative polynomial time inference algorithm that, when it is given the base case of the target program as a hint, identifies the subclass in the limit.

1 はじめに

Shapiro によるモデル推論システム [12, 13] の発表以来，論理プログラムを表現言語とする学習モデルについて多くの研究がなされている．特に，近年では，帰納的論理プログラミング (inductive logic programming) [9] という枠組で盛んに研究が行なわれている．最近の研究の特徴は，モデル推論システムに見られるような枚挙法とは対照的な構成的手法による学習アルゴリズムの開発に注目が集まっているという点にある [9, 6]．そのような論理プログラムの構成的学習アルゴリズムにおいては，最小汎化の概念が (least generalization) [11] が重要な役割を果たす．

有村は木パタン言語の和集合族の推論 [2, 3] の中で，最小汎化を一般化した概念を与えている．最小汎化は，与えられた木パタン (アトムまたは項) の集合 S を覆う 1 個の最小な木パタンであるが，木パタン言語の和集合族の推論では，S を複数の木パタンで極小に覆うことが問題となる．本稿では S を極小に覆う複数の木パタンのことを極小多重汎化と呼ぶことにする．[4, 9] 等に見られる学習アルゴリズムにおいては，与えられた正データの最小汎化を基にプログラム節の頭部を推測する．しかし，一般にプログラムは複数の節からなるから，最小汎化によってそれぞれの節の頭部を推測するためには，与えられた正データを適当に分割し，分割されたそれぞれの正データの部分集合に対して最小汎化をとる必要がある．この「分割してそれぞれの最小汎化をとる」という操作がまさしく極小多重汎化を求めることに対応する．したがって，極小多重汎化は帰納的論理プログラミングにおける極めて基本的かつ重要な概念といえる．本研究の 1 つの目標は極小多重汎化を利用する効率的な学習アルゴリズムの構築を目指すことにある．

一方，正データからの推論可能性については，極く最近，篠原によってその能力が極めて高いという事実が示された [15]．そこでは多くの興味深い概念族が正データだけから推論可能であることが示されている．線形 Prolog (linear Prolog) と呼ばれる論理プログラムもその中の一つである．しかし，そこで示された結果には計算のコストに関する評価が含まれない．そこで，今回は原始 Prolog (primitive Prolog) と呼ばれる線形 Prolog の真部分族を対象に，正データからの多項式時間推論可能性について考察を行なう．

多項式時間推論可能性の定義に関しては Angluin[1] と Pitt[10] によるものが有名であるが，本稿では前者の定義に従う．なお，正データからの多項式時間推論可能性に関する結果としては，いくつかのパタン言語の族が Angluin の意味で推論可能であることが示されている [2, 3, 14] 他，正則言語の真部分族の一部や単純決定性言語の真部分族の一部が，Pitt の意味で推論可能であることなどが示されている [16, 17]．

次節で，以下の議論に必要な基本的な用語の定義を行なう．3 節では極小多重汎化について簡単に紹介する．4 節では，最小汎化や極小多重汎化を用いて頭部を推測するような推論手法において重要な概念となる最大特殊化プログラムについて議論する．5 節で最大特殊化プログラムの性質を利用した本体の探索法について述べる．最後に，原始

Prolog の多項式時間推論可能性について考察を行なう．

2 準備

以下では，有限個の述語記号と有限個の関数記号 (定数記号は 0 引数関数記号と見なす) および可算個の変数記号からなる一階言語 $\mathcal{L}$ を仮定し，アトム，項，確定節 (単に節と呼ぶ)，論理プログラム等はすべて $\mathcal{L}$ 上で定義されるものとする．また，$\mathcal{L}$ の関数記号の集合を Σ で表す．アトム (atom) または項 (term) のことを語 (word) と呼ぶ．変数でない語 (項) を**非変数語 (項)** と呼ぶ．語 t に出現する変数の集合を $var(t)$ で表す．相異なる変数 $x_1,\ldots,x_m$ を含む非変数語を $t[x_1,\ldots,x_m]$ で表す．すなわち，$\{x_1,\ldots,x_m\} \subseteq var(t[x_1,\ldots,x_m])$ である．$t[x_1,\ldots,x_m]$ 中の変数 x_i に項 R_i を代入して得られる語を $t[R_1,\ldots,R_m]$ で表す．また，語 t のすべての基礎代入例の集合を $L(t)$ で表す．語 t に対し $t\theta$ が t の基礎代入例になるような，t 中の変数のみに作用する代入 θ を t の**基礎化代入**と呼ぶ．語 t に現れる記号の総数を $size(t)$ で表し，語の集合 S に対し，$size(S) = \sum_{t\in S} size(t)$ と定義する．集合 S の要素の個数を $|S|$ で表す．

非負整数の有限列 $\langle i_1, i_2, \ldots, i_n\rangle$ を索引と呼ぶ．語 W の部分語 t の W における索引 I を以下のように帰納的に定義する．

- $W = t$ のとき，$I = \langle\,\rangle$．
- $W = \varphi(t_1,\ldots,t_n)$ で t_i における t の索引が $\langle j_1,\ldots,j_m\rangle$ のとき，$I = \langle i, j_1,\ldots,j_m\rangle$．

W における索引が I である部分語を $W(I)$ で表す．

以下の制約を満たす Prolog プログラムを**原始 Prolog** (primitive Prolog) と呼ぶ．

1. 出現する述語記号は 1 個．
2. 高々 2 個の節からなる．
3. 各々の節の頭部は共通インスタンスをもたない．
4. 本体中のアトムの引数には頭部に出現する変数しか現れない．

すなわち，

$$p(t[x_1,\ldots,x_m]) \leftarrow p(x_1),\ldots,p(x_m).$$
$$p(s).$$

の形をした Prolog プログラムである．ただし，$L(t[x_1,\ldots,x_m]) \cap L(s) = \phi$ (空集合) である．

論理プログラム P の最小 Herbrand モデルを M_P または $M(P)$ で表す．α を基礎アトム，M を基礎アトムの集合とし，C を $A \leftarrow B_1,\ldots,B_n$ なる節とする．このとき，C が M において α を**被覆する**とは，以下の条件を満たす代入 θ が存在することをいう．

$$\alpha = A\theta \text{ かつ } B_i\theta \in M \ (1 \le i \le n)$$

C が M において被覆するすべての基礎アトムの集合を $C(M)$ で表す．また，節 C の頭部を $head(C)$ で表す．

3 極小多重汎化

t, s を語とする．$t\theta = s$ なる代入 θ が存在するとき，t は s の汎化であるといい，$t \succeq s$ で表す．$t \succeq s$ であるが，$s \succeq t$ でないことを $t \succ s$ で表す．また，$t\theta = s$ なる代入 θ が存在しないことを $t \not\succeq s$ で表す．一方，$t \succeq s$ かつ $s \succeq t$ であるとき，t と s は**変種** (variant) であるといい，$t \equiv s$ で表す．S を語の集合とするとき，任意の $\alpha \in S$ に対し，$t \succeq \alpha$ なる語 t を S の汎化と呼ぶ．t を S の汎化とする．S の任意の汎化 t' に対し $t' \succeq t$ であるとき，t を S の**最小汎化** (least generalization) と呼び，$lg(S)$ で表す．任意の語の集合 S に対し，$lg(S)$ は $\equiv$ を法として一意である [11]．

k を非負整数とする．語の集合 S に対し，

$$S \subseteq L(t_1) \cup L(t_2) \cup \cdots \cup L(t_m) \ (1 \le m \le k)$$

かつ，任意の

$$S \subseteq L(s_1) \cup L(s_2) \cup \cdots \cup L(s_n) \ (1 \le n \le k)$$

なる $s_1,\ldots,s_n$ に対し、

$$L(s_1)\cup L(s_2)\cup\cdots\cup L(s_n)\not\subseteq L(t_1)\cup L(t_2)\cup\cdots\cup L(t_m)$$

が成り立つとき、語の集合 $\{t_1,\ldots,t_m\}$ を S の k **極小多重汎化** (k-minimally multiple generalization) と呼ぶ.

定理 1 [3] ある固定された非負整数 k に対し, $|\Sigma|\geq k+1$ と仮定する. 任意の語の有限集合 S に対し, S の k 極小多重汎化は $size(S)$ の多項式時間で計算可能である.

一般に S の極小多重汎化は一意に存在するとは限らない. 有村のアルゴリズム (以下では, k-mmg アルゴリズムと呼ぶ) は、極小多重汎化が複数存在する場合の網羅性に関しては保証されていない. しかし, その中の少なくとも一つを $size(S)$ の多項式時間内に出力すること、および出力したものはすべて S の極小多重汎化であることが保証されている.

k と S を入力としたときの, k-mmg アルゴリズムの出力の集合を k-$mmg(S)$ で表す. $P=\{C_0,C_1\}$ を原始 Prolog とし, $h_0=lg(C_0(M_P))$, $h_1=lg(C_1(M_P))$ とする. 2-mmg の計算方法 [2] から次の補題が導ける.

補題 2 $S\subseteq S_\Delta$ なる任意の有限集合 $S_\Delta\subseteq M_P$ に対し, $\{h_0,h_1\}\in$ 2-$mmg(S_\Delta)$ となる有限集合 $S\subseteq M_P$ が存在する.

次節で述べるように, $lg(C_0(M_P))$, $lg(C_1(M_P))$ は P の最大特殊化プログラムの頭部を構成する. したがって, 上の補題は, 原始 Prolog の推論アルゴリズムにおける 2-mmg アルゴリズムの適用可能性を示唆している.

4 原始 Prolog の最大特殊化プログラム

本節では, 最大特殊化と呼ばれる論理プログラムの一種の標準形について定義し, 原始 Prolog の最大特殊化プログラムに関するいくつかの性質を明らかにする. $P=\{C_0,C_1\}$ を原始 Prolog とすると, $M_P=C_0(M_P)\cup C_1(M_P)$ である. 後述の補題 3, 補題 5 で示されるように, $lg(C_0(M_P))$, $lg(C_1(M_P))$ は P の最大特殊化プログラムの 2 つの節の頭部となる. したがって, 最小汎化を用いて節の頭部を推測するような推論アルゴリズムを考える際には最大特殊化の概念が重要となるのである.

C を節, θ を代入とし, P を論理プログラムとする. $C\theta(M_P)\supseteq C(M_P)$ が成り立つとき, $C\theta$ は P に関する C の**特殊化** (more specific version) であるといい, $C\succeq_P C\theta$ で表す.

論理プログラム $P=\{C_1,\ldots,C_n\}$ に対し, $C_i\succeq_P C'_i$ $(1\leq i\leq n)$ なる節 C'_i からなる論理プログラム $P'=\{C'_1,\ldots,C'_n\}$ を P の**特殊化プログラム** (more specific version) と呼び, $P\succeq_P P'$ で表す. P の特殊化プログラムの集合を $SV(P)$ で表す. $P\succeq_P P'$ かつ任意の $P\succeq_P P''$ に対して $P''\succeq_{P''} P'$ であるとき, P' を P の**最大特殊化プログラム** (most specific version) と呼び, $msv(P)$ で表す.

ただし, この定義は [8] による定義とは等価でない. たとえば,

$$C_1=p(x)\leftarrow q(x,y).$$
$$C_2=q(f(x),y)\leftarrow q(x,y).$$
$$C_3=q(f(a),y).$$

からなるプログラム P に対し, C_1 を $C'_1=p(f(x))\leftarrow q(f(x),a)$ で置き換えたプログラム P' は, 上の定義によると P の特殊化プログラムになるが, Marriott らの定義では, C'_1 は C_1 の特殊化ではない. しかし, 対象とするプログラムを変数限定的 (variable bounded)[1] なものに制限した場合には, 等価な定義となる.

[8] により, 任意の論理プログラムは ($\equiv$ を法として) 一意な最大特殊化プログラムを持つことが保証されている. 原始 Prolog は変数限定的であるから, 任意の原始 Prolog は, 上の定義の下で ($\equiv$ を法として) 一意な最大特殊化プログラムを持つことが保証される. より, 具体的には以下の補題で特徴付けられるような最大特殊化プログラムをもつ.

$P=\{C_0,C_1\}$ を原始 Prolog とする. ただし,

$$C_0=p(s).$$
$$C_1=p(t[x_1,\ldots,x_m])\leftarrow p(x_1),\ldots,p(x_m).$$

[1] 本体に現れる変数がすべて頭部に現れるような節からなるプログラム.

補題 3 $|\Sigma| \geq 2$ とする．任意の $P' \in SV(P)$ は，$P' = \{C_0, C_1\theta\}$ の形で与えられる．ただし，

$$\theta = \{x_1/R_1, \ldots, x_m/R_m\}.$$

[証明]　x を C_1 の頭部のみに出現する変数とする．$|\Sigma| \geq 2$ であるから，任意の非変数項 τ と $x/\tau \in \sigma$ なる任意の代入 σ に対して，$C_1\sigma(M_P) \not\supseteq C_1(M_P)$ となる．したがって，代入を施せるのは $\{x_1, \ldots, x_m\}$ 中の変数だけである．

[証明終わり]

$\theta = \{x_1/R_1, \ldots, x_m/R_m\}$ とし，$msv(P) = \{C_0, C_1\theta\}$ とする．すなわち，

$$C_0 = p(s).$$
$$C_1\theta = p(t[R_1, \ldots, R_m]) \leftarrow p(R_1), \ldots, p(R_m).$$

補題 4 $lg(\{s, t[R_1, \ldots, R_m]\}) \succ s$ かつ $lg(\{s, t[R_1, \ldots, R_m]\}) \succ t[R_1, \ldots, R_m]$.

[証明]　$lg(\{s, t[R_1, \ldots, R_m]\}) \equiv s$ と仮定すると，$s \succeq t[R_1, \ldots, R_m]$ となり，これは原始 Prolog の条件 $L(s) \cap L(t[x_1, \ldots, x_m]) = \phi$ に矛盾する．$lg(\{s, t[R_1, \ldots, R_m]\}) \succ t[R_1, \ldots, R_m]$ についても同様．　[証明終わり]

補題 5 $|\Sigma| \geq 2$ のとき，$p(t[R_1, \ldots, R_m]) \equiv lg(C_1(M_P))$.

[証明]　[4] の定理 2 と同様の議論により，$head(C_1)\sigma \equiv lg(C_1(M_P))$ なる代入 σ に対し，$C_1\sigma(M_P) = C_1(M_P)$. したがって，$\{C_0, C_1\sigma\} \in SV(P)$.

一方，任意の $P' = \{C_0, C_1''\} \in SV(P)$ に対し，$C_1''(M_P) \supseteq C_1(M_P)$ であるから，$head(C_1'') \succeq lg(C_1(M_P))$. したがって，ある代入 δ が存在して，$head(C_1''\delta) = head(C_1\sigma)$. C_1 は変数限定的であるから，$C_1''\delta \equiv C_1\sigma$. [8] の命題 3.1 より，$M_P = M_{P'}$ が成り立つから，

$$C_1''(M_{P'}) = C_1''(M_P) \supseteq C_1(M_P) = C_1\sigma(M_P) = C_1\sigma(M_{P'}).$$

すなわち，$C_1'' \succeq_{P'} C_1\sigma$ であり，$P' \succeq_{P'} \{C_0, C_1\sigma\}$. よって，$\{C_0, C_1\sigma\} = msv(P)$ となり，補題の主張が成り立つ．

[証明終わり]

補題 6 任意の $1 \leq i \leq m$ に対し，$p(R_i) \equiv lg(M_P)$.

[証明]　ある $1 \leq i \leq m$ に対し，$p(R_i) \not\succeq lg(M_P)$ と仮定すると，$p(R_i) \not\succeq p(\beta)$ なる $p(\beta) \in M_P$ が存在する．一方，$p(t[\beta, \ldots, \beta])\gamma \in C_1(M_P)$ なる代入 γ が存在する．ある $1 < i \leq m$ に対して，$p(R_i) \not\succeq p(\beta)$ であるから，

$$head(C_1\theta) = p(t[R_1, \ldots, R_m]) \not\succeq p(t[\beta, \ldots, \beta])\gamma$$

したがって，$p(t[\beta, \ldots, \beta])\gamma \notin C_1\theta(M_P)$. これは，$C_1\theta(M_P) \supseteq C_1(M_P)$ に反する．よって，任意の $1 \leq i \leq m$ に対して，$p(R_i) \succ lg(M_P)$.

ある $1 \leq i \leq m$ に対して，$p(R_i) \succ lg(M_P)$ と仮定する．$lg(M_P) \equiv p(\tau)$ とし，$\sigma = \{x_i/\tau\}$ とおくと，明らかに $C_1\sigma(M_P) \supseteq C_1(M_P)$ が成り立つ．したがって，$P' = \{C_0, C_1\sigma\}$ とおくと，$C_1 \succeq_P C_1\sigma$ であるから，$P' \in SV(P)$ となる．ところが，$R_i \succ \tau$ であるから，$C_1\sigma \not\succeq C_1\theta$. すなわち，$C_1\sigma\gamma = C_1\theta$ なる代入 γ は存在しない．したがって，$P' \not\succeq_{P'} msv(P)$ となり，矛盾する．

[証明終わり]

定理 7 任意の $1 \leq i \leq m$ に対し，$p(R_i) \equiv lg(\{p(s), p(t[R_1, \ldots, R_m])\})$.

[証明]　補題 3 と補題 5 より，

$$lg(\{p(s), p(t[R_1, \ldots, R_m])\}) \equiv lg(\{lg(C_0(M_P)), lg(C_1(M_P))\}).$$

$C_0(M_P) \cup C_1(M_P) = M_P$ であるから，[5] の補題 4 より，

$$lg(\{lg(C_0(M_P)), lg(C_1(M_P))\}) \equiv lg(C_0(M_P) \cup C_1(M_P)) \equiv lg(M_P).$$

補題 6 より，$lg(M_P) \equiv p(R_i)$ が成り立つ．

[証明終わり]

補題 8 $i \neq j$ なる任意の $1 \leq i, j \leq m$ に対し，$var(R_i) \cap var(R_j) = \phi$.

[証明] ある $i \neq j$ に対し，$var(R_i) \cap var(R_j) \neq \phi$ と仮定する．すなわち，索引 I と J が存在して，$p(R_i)(I) = p(R_j)(J) = x$ である．ただし，x は変数．$p(R_i) \equiv lg(M_P)$ であるから，任意の $p(\alpha) \in M_P$ に対し，$p(\alpha)(I)$ はある基礎項になっている．

ここで，a, b を 2 つの異なる基礎項とし，$p(\alpha)(I) = a$ なる $p(\alpha) \in M_P$ と $p(\beta)(J) = b$ なる $p(\beta) \in M_P$ が存在すると仮定する．節

$$C_1 = p(t[x_1, \ldots, x_m]) \leftarrow p(x_1), \ldots, p(x_m)$$

において $x_1, \ldots, x_m$ は相異なる変数であるから，$p(t') = p(t[x_1, \ldots, x_m])\{x_i/\alpha, x_j/\beta\}$ とおくと，$p(t')\gamma \subset C_1(M_P)$ なる $p(t')$ の基礎化 γ が存在する．一方，$C_1\theta$ においては，$p(R_i)(I) = p(R_j)(J) = x$ であるから，$p(R_i)\sigma = p(\alpha)$ かつ $p(R_j)\sigma = p(\beta)$ なる代入 σ は存在しない．すなわち，$p(t') \notin C_1\theta(M_P)$．これは $msv(P) = \{C_0, C_1\theta\}$ に矛盾する．したがって，上の条件を満たす α, β は存在しない．すなわち，任意の $p(\alpha), p(\beta) \in M_P$ に対し，$p(\alpha)(I) = p(\beta)(J)$ である．したがって，ある基礎項 a が存在して，任意の $p(\alpha) \in M_P$ に対して，$p(\alpha)(I) = a$ となる．なぜなら，相異なる基礎項 a, b と $p(\alpha)(I) = a$, $p(\alpha')(I) = b$ なる α, $\alpha' \in M_P$ が存在するならば，$p(\alpha)(I) = p(\beta)(J) = a$ なる $p(\beta) \in M_P$ に対し，$p(\alpha')(I) \neq p(\beta)(J)$ となり上の結論に反する．$p(R_i) \equiv lg(M_P)$ であるから，結局，$p(R_i)(I) = a$ となり最初の仮定に矛盾する． [証明終わり]

同様の議論によって次の補題も成り立つ．

補題 9 任意の $1 \leq i \leq m$ に対し，$var(R_i) \cap (var(t[x_1, \ldots, x_m]) - \{x_1, \ldots, x_m\}) = \phi$.

補題 10 $P' = \{C_0, C_1''\}$ とする．ただし，

$$C_1'' = p(t[R_1, \ldots, R_m]) \leftarrow p(R_{i_1}), \ldots, p(R_{i_k}).\ \text{かつ}\ \{R_{i_1}, \ldots, R_{i_k}\} \subseteq \{R_1, \ldots, R_m\}$$

である．このとき，$M_P \subset M_{P'}$ が成り立つ．

[証明] 任意の基礎アトムの集合 M に対し，$C_1\theta(M) \subseteq C_1''(M)$ であるから，$M_P = M(msv(P)) \subseteq M_{P'}$.

[証明終わり]

補題 11 $P' = \{C_0, C_1''\}$ とする．ただし，$C_1'' = p(t[R_1, \ldots, R_m]) \leftarrow p(R')$ であり，R' は $R' \equiv R_i$ $(1 \leq i \leq m)$ かつ $R' \notin \{R_1, \ldots, R_m\}$ なる $t[R_1, \ldots, R_m]$ の部分項である．このとき，$M_P - M_{P'} \neq \phi$ が成り立つ．

[証明] 補題 8，補題 9 および R' の条件から，$var(R') \cap var(R_i) = \phi$ $(1 \leq i \leq m)$ が成り立つ．したがって，R' の任意の基礎化代入 σ に対し，

$$p(t[R_1, \ldots, R_m])\sigma\gamma \in M_P$$

なる基礎化代入 γ が存在する．

補題の証明は

$$t[R_1, \ldots, R_m] \not\succeq R'\sigma \ \text{かつ}\ s \not\succeq R'\sigma \qquad (*)$$

なる R' の基礎化代入 σ が存在することを示せば十分．なぜなら，$(*)$ より $p(R')\sigma \notin M_{P'}$ が成り立つから，任意の基礎化代入 γ に対して，$p(t[R_1, \ldots, R_m])\sigma\gamma \notin M_{P'}$ となる．上の事実と合わせると，

$$p(t[R_1, \ldots, R_m])\sigma\gamma \in M_P \ \text{かつ}\ p(t[R_1, \ldots, R_m])\sigma\gamma \notin M_{P'}$$

なる γ が存在する．

$R' \equiv R_i$ であるから，補題 4 より

$$R' \succ s \ \text{かつ}\ R' \succ t[R_1, \ldots, R_m]$$

が成り立つ．以下，$t[R_1, \ldots, R_m]$ を単に t と略記する．

$R' \succ s$ より，次の 2 つの場合が考えられる．

(1) ある索引 I が存在して $R'(I) = x$ かつ $s(I) = u$．ただし，x は変数，u は非変数項．

(2) ある索引 I_1, I_2 が存在して $R'(I_1) = x$, $R'(I_2) = y$ かつ $s(I_1) = z$, $s(I_2) = z$．ただし，x, y, z は異なる変数．

$R' \equiv lg(s,t)$ であるから，さらに，これらは以下のように場合分けできる．

(1-1) ある索引 I が存在して $R'(I) = x$，$s(I) = u$ かつ $t(I) = v$．ただし，x は変数，u，v は主関数子が異なる非変数項．

(1-2) ある索引 I が存在して $R'(I) = x$，$s(I) = u$ かつ $t(I) = y$．ただし，x，y は変数，u は非変数項．

(2-1) ある索引 I_1，I_2 が存在して $R'(I_1) = x$，$R'(I_2) = y$，$s(I_1) = z$，$s(I_2) = z$ かつ $t(I_1) = u$，$t(I_2) = v$．ただし，x, y, z は異なる変数，u，v は相異なる項であり，少なくとも一方は非変数項．

(2-2) ある索引 I_1，I_2 が存在して $R'(I_1) = x$，$R'(I_2) = y$，$s(I_1) = z$，$s(I_2) = z$ かつ $t(I_1) = x_1$，$t(I_2) = y_1$．ただし，x, x_1, y, y_1, z は異なる変数．

(1-1) の場合：

$|\Sigma| \geq 3$ の仮定から，主関数子が u，v の主関数子と異なる基礎項 w が存在する．したがって，x に w を代入するような任意の基礎化代入 σ に対し $(*)$ が成り立つ．

(1-2) の場合：

$R' \succ t$ であるから，s の場合と同様に，

(1') I と異なる索引 I' が存在して $R'(I') = x'$ かつ $t(I') = u'$．ただし，x' は変数，u' は非変数項．

(2') ある索引 I_1'，I_2' が存在して $R'(I_1') = x'$，$R'(I_2') = y'$ かつ $t(I_1') = z'$，$t(I_2') = z'$．ただし，x', y', z' は異なる変数．

なる場合が存在する．(1') の場合，$x \neq x'$ が成り立つ[2]から，x に主関数子が u の主関数子と異なる基礎項 w を，x' に主関数子が u' の主関数子と異なる基礎項 w' を代入するような任意の基礎化代入 σ に対し $(*)$ が成り立つ．(2') の場合，主関数子が u の主関数子と異なる基礎項を w_1，w_1 と異なる基礎項を w_2 とし，x に w_1 を，x' には w_1，w_2 のどちらか一方を y' には他方を[3]代入するような基礎化代入 σ に対し $(*)$ が成り立つ．

(2-1) の場合：

u を非変数項としても一般性を失わない．主関数子が u の主関数子と異なる基礎項を w_1，w_1 と異なる基礎項を w_2 とすると，x に w_1 を y に w_2 を代入するような任意の基礎化代入 σ に対し $(*)$ が成り立つ．

(2-2) の場合：

(1-2) の場合と同様に (1')，(2') の場合がある．(1') の場合は，(1-2) における (2') の場合と同じ．(2') の場合は，$x \neq y$ かつ $x' \neq y'$ であるから，2 個の異なる基礎項 w_1，w_2 をとり，それぞれ x，x' に w_1，w_2 のどちらか一方を y，y' に他方を代入するような基礎化代入を考えれば良い． [証明終わり]

補題 12 $P' = \{C_0', head(C_1\theta)\}$ とする．ただし，$C_0' = p(s) \leftarrow p(R')$ であり，R' は $R' \equiv R_i$ $(1 \leq i \leq m)$ なる s の部分項である．このとき，$M_P - M_{P'} \neq \phi$ が成り立つ．

[証明] $L(p(s)) \subseteq M_P$ であり R' は s の部分項であるから，R' の任意の基礎化代入 σ に対し，$p(s)\sigma\gamma \in M_P$ なる $p(s)\sigma$ の基礎化代入 γ が存在する．したがって，補題 11 と同様の議論によって証明できる．

[証明終わり]

5 本体の推測

本節では，正データの集合 S と，アトムの順序対 $\langle p(s), p(t)\rangle$ が与えられたとき，以下の条件を満たす論理プログラム $P = \{C_0, C_1\}$ を探索するアルゴリズムについて述べる．ただし，

$$C_0 = p(s).$$
$$C_1 = p(t) \leftarrow p(r_1), \ldots, p(r_k).$$

1. $S \subseteq M_P$.

[2] $t(I)$ は変数であり $t(I')$ は非変数項であるから $x \neq x'$.

[3] x，x'，y' がすべて異なる場合は，この選択は適当で良いが，x と x' が同じ変数の場合には y' に w_2 が，x と y' が同じ変数の場合には x' に w_2 が代入されるように選択する．

2. 各 r_i は $r_i = lg(t,s)$ $(1 \le i \le k)$ かつ $var(r_i) \cap var(r_j) = \phi$ $(i \ne j)$ なる t の真部分項.
3. $x_1, \ldots, x_k$ を t に現れない相異なる変数とする. t における $r_1, \ldots, r_k$ の出現を $x_1, \ldots, x_k$ で置き換えて得られる項を $t'[x_1, \ldots, x_k]$ で表すと, $L(t'[x_1, \ldots, x_k]) \cap L(s) = \phi$ が成り立つ.
4. C_1 の本体は, 以上の条件を満たす極大なものである. すなわち, $r_1, \ldots, r_k$ 以外の t の部分項で条件2を満たす任意の r_{k+1} に対し, $C = p(t) \leftarrow p(r_1), \ldots, p(r_k), p(r_{k+1})$ とおくと, $S \not\subseteq M(\{C_0, C\})$ または C は条件3を満たさない.

以上の条件を満たすプログラムを $P(S, \langle p(s), p(t) \rangle)$ で表すことにする. 一般に, $P(S, \langle p(s), p(t) \rangle)$ は一意ではないが, その中の1つを単純な greedy 探索によって $size(S)$ の多項式時間で見つけることができる.

補題 13 基礎アトムの任意の集合 S と, アトムの順序対 $\langle p(s), p(t) \rangle$ に対し, アルゴリズム1は $size(S \cup \{p(s), p(t)\})$ の多項式時間で $P(S, \langle p(s), p(t) \rangle)$ を出力する.

アルゴリズム 1: Greedy search algorithm for $P(S, \langle p(s), p(t) \rangle)$

Input: A set of positive examples S and a tuple of atoms $\langle p(s), p(t) \rangle$.
Output: $P(S, \langle p(s), p(t) \rangle)$
Procedure:

Let $SUB(t)$ be the set of proper subterms of t, that are variant of $lg(t,s)$.
$Body := \phi$; $Pat := t$;
$H := \{p(t) \leftarrow Body., p(s).\}$;
for each $T \in SUB(t)$ s.t. $var(T) \cap var(Body) = \phi$ **do**
  $H := \{p(t) \leftarrow Body \cup \{p(T)\}., p(s).\}$;
  if $S \not\subseteq M_H$ **then**
    $H := \{p(t) \leftarrow Body., p(s).\}$;
  else Let Pat' be the term which is obtained from Pat by replacing
    the all occurrences of T in Pat with a new variable x;
    if $L(Pat') \cap L(s) \ne \phi$ **then** $H := \{p(t) \leftarrow Body., p(s).\}$;
    else $Body := Body \cup \{p(T)\}$;
      $Pat := Pat'$;
Output H and halt;

[証明] $lg(t,s)$ は $size(\{p(s), p(t)\})$ の多項式時間で計算可能である. $|SUB(t)| \le size(t)$ であり $SUB(t)$ は $size(t)$ の多項式時間で計算可能である. アルゴリズム1の for ループの本体の実行回数は高々 $|SUB(t)| \le size(t)$ であり, $var(T) \cap var(Body) = \phi$ の条件判定も $size(t) \times |SUB(t)| \le size(t)^2$ の多項式時間で行なえる. 各実行における主な計算は2つの **if** 分岐における条件判定である. 外側の **if** 分岐における $S \subseteq M_H$ か否かのチェックは, H の再帰節 $p(t) \leftarrow p(T_1), \ldots, p(T_j)$ における, T_i が t の真部分項であるから, $size(S)$ の多項式時間で終了する. 一方, 内側の **if** 分岐の条件判定は, 語の内部表現として DAG を用い, 2つの DAG の単一化可能性のテストを行なえばよいから, 高々 $max\{size(s), size(t)\}$ の多項式時間で終了する. したがって, アルゴリズム1は $size(S \cup \{p(s), p(t)\})$ の多項式時間で停止する.

アルゴリズムにおける本体の探索法から, 生成される出力が $P(S, \langle p(s), p(t) \rangle)$ の条件を満たすことは明らかである.

[証明終わり]

$P(S, \langle p(s), p(t) \rangle) = \{C_0, C_1\}$, ただし,

$$C_0 = p(s).$$
$$C_1 = p(t) \leftarrow p(r_1), \ldots, p(r_m).$$

をアルゴリズム1の出力とする. 節 C_1 における項 $r_1, \ldots, r_m$ の出現を, C_1 に現れない相異なる変数 $x_1, \ldots, x_m$ で置き換えて得られる節を $C_1' = p(t') \leftarrow p(x_1), \ldots, p(x_m)$. とし, $\{C_0, C_1'\} = pr(P(S, \langle p(s), p(t) \rangle))$ と表す.

補題 14 $pr(P(S,\langle p(s),p(t)\rangle))$ は $P(S,\langle p(s),p(t)\rangle)$ と等価な原始 Prolog である．すなわち，

$$M(pr(P(S,\langle p(s),p(t)\rangle))) = M(P(S,\langle p(s),p(t)\rangle))$$

が成り立つ．さらに，$pr(P(S,\langle p(s),p(t)\rangle))$ は $P(S,\langle p(s),p(t)\rangle)$ から $size(p(t))$ の多項式時間で計算可能である．

[証明]　$size(p(t))$ の多項式時間で計算可能であることは明らか．また，$L(t[x_1,\ldots,x_m]) \cap L(s) = \phi$ が成り立つから原始 Prolog であることも明らか．

簡単のために，$P(S,\langle p(s),p(t)\rangle) = P$ と略記する．$\sigma = \{x_1/r_1,\ldots,x_m/r_m\}$ とおくと，$C_1 = C_1'\sigma$ であるから明らかに $M_P \subseteq M(pr(P))$ が成り立つ．

$M(pr(P)) \subseteq M_P$ を示す．[7] の定理6.5 より，$M_P = T_P \uparrow \omega$ であるから，任意の正整数 n に対して，$T_{pr(P)} \uparrow n \subseteq T_P \uparrow n$ が成り立つことを n に関する帰納法で示す．

$n = 1$ の場合，$T_{pr(P)} \uparrow 1 = L(C_0) = T_P \uparrow 1$ であるから成り立つ．

$n \geq 2$ に対して，$T_{pr(P)} \uparrow n \subseteq T_P \uparrow n$ と仮定する．任意の $\alpha \in T_{pr(P)} \uparrow (n+1)$ に対し，ある代入 σ が存在し，

$$\alpha = p(t')\sigma \text{ かつ } p(x_i)\sigma \in T_{pr(P)} \uparrow n \ (1 \leq i \leq m)$$

である．ここで，$\{x_1,\ldots,x_m\}$ 中の変数のみに作用する代入 δ と，$var(t') - \{x_1,\ldots,x_m\}$ 中の変数のみに作用する代入 γ を用いて，$\sigma = \gamma\delta$ と分割できる．帰納法の仮定より，$p(x_i)\sigma \in T_P \uparrow n \ (1 \leq i \leq m)$．$p(r_i) \equiv lg(\{p(t),p(s)\})$ であるから，任意の $p(x_i)\sigma \in T_P \uparrow n$ に対し，$p(x_i)\sigma = p(r_i)\delta_i$ なる $p(r_i)$ の基礎化代入 δ_i が存在する．$\{x_1,\ldots,x_m\}$ と r_i および r_i 同士は変数を共有しないから，$\delta' = \delta_1 \cdots \delta_m$ とおくと，

$$\alpha = p(t)\gamma\delta' \text{ かつ } p(r_i)\gamma\delta' \in T_P \uparrow n \ (1 \leq i \leq m)$$

となり，$\alpha \in T_P \uparrow (n+1) \ (1 \leq i \leq m)$　　[証明終わり]

6　原始 Prolog の推論可能性

「入力要求 — 計算 — 出力」という過程を繰り返すアルゴリズム $\mathcal{A}$ のことを**推論アルゴリズム**と呼ぶ．入力の列 $e_1,e_2,\ldots$ に対して生成される出力の列を $g_1,g_2,\ldots$ とし，$S_i = \{e_1,\ldots,e_i\}$ とする．推論アルゴリズム $\mathcal{A}$ に対し，ある多項式 f が存在して，任意の時点 i における $\mathcal{A}$ の計算時間 (入力 e_i を受け取ってから g_i を出力するまでの時間) が $f(size(S_i))$ で抑えられるとき，$\mathcal{A}$ を**多項式時間推論アルゴリズム**と呼ぶ．ある自然数 N が存在して，$N \leq n$ なる任意の自然数 n に対して $g_N = g_n$ を満たすとき，$\mathcal{A}$ は入力列 $e_1,e_2,\ldots$ に対して g_N に収束するという．

$\mathcal{P}$ をある論理プログラムの族とし，$\mathcal{M} = \{M_P \mid P \in \mathcal{P}\}$ とする．モデル M の任意の要素 e に対し，ある自然数 i が存在して $e_i = e$ となるような M の要素の列 $e_1,e_2,\ldots$ を M の枚挙という．モデルの任意の枚挙 $e_1,e_2,\ldots$ に対し，推論アルゴリズム $\mathcal{A}$ が生成する出力の列を $P_1,P_2,\ldots$ とし，$S_i = \{e_1,\ldots,e_i\}$ とする．任意の時点 i において，$S_i \subseteq M(P_i)$ が成り立つとき，$\mathcal{A}$ は**無矛盾**であるという．また，任意の時点 i において，$e_i \in M(P_{i-1})$ ならば $P_i = P_{i-1}$ であるとき，$\mathcal{A}$ は**保守的**であるという．$\mathcal{M}$ の任意の要素 M と M の任意の枚挙 $e_1,e_2,\ldots$ に対し，(無矛盾な，保守的な，多項式時間) 推論アルゴリズム $\mathcal{A}$ が $M = M_P$ なる $P \in \mathcal{P}$ に収束するとき，$\mathcal{A}$ は $\mathcal{M}$ を**正データから (無矛盾に，保守的に，多項式時間) 極限同定する**という．$\mathcal{M}$ を正データから極限同定する (無矛盾な，保守的な，多項式時間) 推論アルゴリズムが存在するとき，$\mathcal{M}$ は**正データから (無矛盾に，保守的に，多項式時間) 推論可能**であるという．

以下では，$P = \{C_0, C_1\}$ を任意の原始 Prolog とする．ただし，

$$C_0 = p(s).$$
$$C_1 = p(t[x_1,\ldots,x_m]) \leftarrow p(x_1),\ldots,p(x_m).$$

さらに，$msv(P) = \{C_0, C_1\theta\}$ とする．ただし，$\theta = \{x_1/R_1,\ldots,x_m/R_m\}$ である．簡単のために，$t[x_1,\ldots,x_m]$ を t と略記し，$t[R_1,\ldots,R_m]$ を t_R と略記する．また，$e_1,e_2,\ldots$ を M_P の任意の枚挙とし，$S_i = \{e_1,\ldots,e_i\}$ とする．

定理 15 ある自然数 N が存在して，$N \leq n$ なる任意の自然数 n に対して，S_n と $msv(P)$ の頭部の対 $\langle p(s),p(t_R)\rangle$ とを入力としたときのアルゴリズム 1 の出力は $msv(P)$ になる．

[証明]　$T_1, \ldots, T_k$ を $R_1, \ldots, R_m$ 以外の $p(T_j) \equiv lg(\{p(s), p(t_R)\}$ なる t_R のすべての部分項とする．補題11より任意の T_i に対し，

$$C_{T_i} = p(t_R) \leftarrow p(T_i).$$

とおくと，$P_{T_i} = \{C_0, C_{T_i}\}$ には正反例，すなわち $c_i \notin M(P_{T_i})$ なる $c_i \in M_P$ が存在する．よって，$\{c_1, \ldots, c_k\} \subseteq S_n$ なる任意の S_n に対し，アルゴリズム1の出力は，$P = \{C_0, C'_1\}$，ただし，

$$C'_1 = p(t_R) \leftarrow p(R_{i_1}), \ldots, p(R_{i_j}).$$

$\{R_{i_1}, \ldots, R_{i_j}\} \subseteq \{R_1, \ldots, R_m\}$ なるプログラムになる．一方，補題10と $P(S_n, \langle p(s), p(t_R)\rangle)$ の極大性(条件4)から $\{R_{i_1}, \ldots, R_{i_j}\} \subset \{R_1, \ldots, R_m\}$ となることはない．[証明終わり]

定理15により，目標のプログラムの base case $p(s)$ が推論アルゴリズムに陽に与えられる場合には，原始 Prolog のモデルの族を無矛盾かつ保守的に多項式時間極限同定するアルゴリズムが構成できる．

アルゴリズム2：無矛盾かつ保守的な多項式時間推論アルゴリズム

Given: $p(s)$ that is the base case of a target program P.

Input: $e_1, e_2, \ldots$: any enumeration of M_P.

Output: $P_1, P_2, \ldots$: a sequence of primitive Prologs.

Procedure:

　$S := \{\}$; $H := \{p(s)\}$;
　Read the next example e; $S := S \cup \{e\}$;
　if $e \in M_H$ **then** output H;
　else　$S_- := S - (S \cap L(p(s)))$;
　　Compute $P(S, \langle p(s), lg(S_-)\rangle)$ using Algorithm 1;
　　$H := pr(P(S, \langle p(s), lg(S_-)\rangle))$; output H;

補題 16 アルゴリズム2は無矛盾かつ保守的な多項式時間推論アルゴリズムである．

[証明]　初期仮説 $\{p(s)\}$ は原始 Prolog である．$M_P = C_0(M_P) \cup C_1(M_P) = L(p(s)) \cup C_1(M_P)$ であり $S \subseteq M_P$ かつ $L(p(s)) \cap L(p(t)) = \phi$ であるから，$S_- = S - (S \cap L(p(s))) \subseteq C_1(M_P)$．すなわち，$lg(S_-) \preceq p(t_R) \preceq p(t)$ であり $L(p(s)) \cap L(lg(S_-)) = \phi$ が成り立つ．よって，補題14より初期仮説以外の仮説もすべて原始 Prolog であることが保証される．したがって，任意の時点において $e \in M_H$ の判定は $size(e)$ の多項式時間で終了する．S_-，$lg(S_-)$ は $size(S)$ の多項式時間で計算可能．$size(lg(S_-)) \leq size(S)$ であり，$p(s)$ は base case であるから，任意の $\alpha \in M_P$ に対して $size(p(s)) \leq size(\alpha)$ である．したがって，$size(p(s)) \leq size(S)$ であるから，補題13および補題14より $P(S, \langle p(s), lg(S_-)\rangle)$ および $pr(P(S, \langle p(s), lg(S_-)\rangle))$ の計算も $size(S)$ の多項式で抑えられる．したがって，アルゴリズム2は多項式時間アルゴリズムである．

補題13，補題14より無矛盾性は明らか．$e \in M_H$ の場合は H をそのまま出力するので保守性も明らか．

[証明終わり]

定理 17 モデル M_P の任意の枚挙に対し，アルゴリズム2の出力は $M_H = M_P$ なる原始 Prolog H に収束する．

[証明]　任意の $S \subseteq M_P$ に対し，$S_- = S - (S \cap L(p(s))) \subseteq C_1(M_P)$ であるから，$lg(S_-) \preceq lg(C_1(M_P))$．一方，補題5より，$p(t_R) \equiv lg(C_1(M_P))$ である．[5] における無限集合に対する最小汎化の議論より，ある有限集合 $S^* \subseteq C_1(M_P)$ が存在して $lg(S^*) \equiv p(t_R)$ である．したがって，$S^* \subseteq S_n$ なる任意の自然数 n に対して，$lg(S_-) = p(t_R)$ となる．よって，補題14および定理15より定理の主張は成り立つ．[証明終わり]

く

7　おわりに

正データからの原始 Prolog の多項式時間推論可能性について議論した．残念ながら，今回は，極小多重汎化の概念を有効に利用するアルゴリズムの構築には至らなかった．そのため，base case が予めアルゴリズムに与えられることを仮定するという，極めて弱い結果しか示せなかった．しかし，補題 2が成り立つことを考えると，極小多重汎化利用の可能性は十分にあると思われる．極小多重汎化を利用した節の頭部の推測法について検討し，その結果を，最大特殊化プログラムの性質を利用した今回の結果に有効に反映させていくことが今後の課題である．

参考文献

[1] D. Angluin. Inductive inference of formal languages from positive data. *Information and Control*, 45:117–135, 1980.

[2] H. Arimura, T. Shinohara, and S. Otsuki. Polynomial time inference of unions of tree pattern languages. In S. Arikawa, A. Maruoka, and T. Sato, editors, *Proc. ALT '91*, pp. 105–114. Ohmsha, 1991.

[3] H. Arimura, T. Shinohara, and S. Otsuki. A polynomial time algorithm for finite unions of tree pattern languages. In *Proc. of the 2nd International Workshop on Nonmonotonic and Inductive Logics*, 1991. To appear in LNCS.

[4] H. Ishizaka. Model inference incorporating generalization. *Journal of Information Processing*, 11(3):206–211, 1988.

[5] J-L.Lassez, M. J. Maher, and K. Marriott. Unification revisited. In J. Minker, editor, *Foundations of Deductive Databases and Logic Programming*, pp. 587–625. Morgan Kaufmann, 1988.

[6] X. Ling. Learning and invention of horn clause theories - a constructive method. In Z. W. Ras, editor, *Methodologies for Intelligent Systems, 4*, pp. 323–331. North-Holland, October 1989.

[7] J. W. Lloyd. *Foundations of Logic Programming*. Springer-Verlag, 1984.

[8] K. Marriott, L. Naish, and J.-L. Lassez. Most specific logic programs. In *Logic Programming: Proceedings of the Fifth International Conference and Symposium*, pp. 910–923. MIT Press, 1988.

[9] S. Muggleton. Inductive logic programming. In S. Arikawa, S. Goto, S. Ohsuga, and T. Yokomori, editors, *Proc. ALT '90*, pp. 42–62. Ohmsha, 1990.

[10] L. Pitt. Inductive inference, dfas, and computational comlexity. In K. P. Jantke, editor, *Proc. AII '89, LNAI 397*, pp. 18–44. Springer-Verlag, 1989.

[11] G. D. Plotkin. A note on inductive generalization. In B. Meltzer and D. Michie, editors, *Machine Intelligence 5*, pp. 153–163. Edinburgh University Press, 1970.

[12] E. Y. Shapiro. Inductive inference of theories from facts. Technical Report 192, Yale University Computer Science Dept., 1981. (有川節夫訳：知識の帰納推論，共立出版，1986).

[13] E. Y. Shapiro. *Algorithmic program debugging*. PhD thesis, Yale University Computer Science Dept., 1982. Published by MIT Press, 1983.

[14] T. Shinohara. Polynomial time inference of extended regular pattern languages. *LNCS* 147, pp. 115–127, 1983.

[15] T. Shinohara. Inductive inference of monotonic fomal systems from positive data. In S. Arikawa, S. Goto, S. Ohsuga, and T. Yokomori, editors, *Proc. ALT '90*, pp. 339–351. Ohmsha, 1990.

[16] T. Yokomori. Inductive inference of very simple grammars from positive data. Reseach Report CSIM 90-15, Dept. of Cumput. Sci. and Inform. Math., Univ. of Electro-Communications, 1990.

[17] T. Yokomori. A note on the polynomial-time identification of strictly local languages in the limit. Reseach Report CSIM 90-03, Dept. of Cumput. Sci. and Inform. Math., Univ. of Electro-Communications, 1990.