特集論文

Technical Papers

極小多重汎化を用いた正事実からの論理プログラムの帰納的学習

Inductive Learning of Logic Programs From Positive Facts Using Minimal Multiple Generalization

石坂　裕毅*1　有村　博紀*2　篠原　武*2
Hiroki Ishizaka　Hiroki Arimura　Takeshi Shinohara

*1 (株)富士通研究所 国際情報社会科学研究所
IIAS-SIS, Fujitsu Laboratories Ltd., Shizuoka 410-03, Japan.
*2 九州工業大学情報工学部
Faculty of Computer Science and Systems Eng., Kyushu Institute of Technology, Iizuka 820, Japan.

1993年3月4日　受理

Keywords: learning, logic program, least generalization, minimal multiple generalization.

Summary

We consider the polynomial time inferability of primitive Prologs from positive facts. The class of primitive Prologs is a proper subclass of that of linear Prologs which is known to be inferable from only positive facts. In this paper, we discuss the polynomial time inferability of the subclass using minimal multiple generalizations. The minimal multiple generalization is a natural extension of the least generalization given by Plotkin in 1970. The minimal multiple generalization generalizes given first order words by several words, while the least generalization does by a single word. The property of the minimal multiple generalization makes it possible to perform fine generalization and to construct the heads of several clauses in a target program at the same time. We give an outline of a consistent polynomial time inference algorithm which identifies the class of primitive Prologs in the limit. The algorithm infers the heads of clauses in a target program as a minimal multiple generalization of a set of given positive facts. Furthermore, we give a similar result on the inferability of a subclass of context-free transformations which includes several well-known Prolog programs.

1. はじめに

Shapiroによるモデル推論システム[Shapiro 81, Shapiro 83]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)[Muggleton 90]という枠組みで盛んに研究が行われている．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[Ling 89, Muggleton 90]．こ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（least generalization）の概念[Plotkin 70]が重要な役割を果たす．

最近，木パターン言語（tree pattern language）の和集合族の推論[Arimura 91a, Arimura 91b]という形で，最小汎化を拡張した概念が与えられた．木パターンとは，一階言語におけるアトム（atom）あるいは項（term）のことであり，木パターン言語とは，その木パターンの基礎代入例（ground instance）全体の集合を指す．最小汎化は，与えられた木パターンの集合 S を最小に覆う1個の木パターンであるが，木パターン言語の和集合族の推論では，S を複数の木パター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ーンのことを極小多重汎化（minimal multiple generalization）と呼ぶことにする．例えば，リスト連接関係を表す述語 $app(\_,\_,\_)$ に

関する以下のような具体例の集合 S に対し，

$$S=\{app([],[],[]), app([a],[],[a]), app([],[b],[b]), app([a],[b],[a,b]), app([a,b],[c],[a,b,c])\}$$

S の最小汎化は単一のアトム $app(X,Y,Z)$ となるが，極小多重汎化はアトムの対

$$\{app([],X,X), app([X|Y],Z,[X|W])\}$$

といった形で与えられる．

[Ishizaka 88, Muggleton 90]などに見られる学習アルゴリズムにおいては，与えられた正事実の最小汎化をもと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，与えられた正事実を適当に分割し，分割されたそれぞれの正事実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基本的かつ重要な概念といえる．

一方，与えられた木パターンに対して，それらの最小汎化あるいは極小多重汎化を見つける問題は，ともに正事実からの帰納推論の一種とみなせる．正事実からの帰納推論可能性に関しては，これもごく最近の結果であるが，[Shinohara 90]において，その能力が極めて高いという事実が示された．そこでは多くの興味深い概念族が正事実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なかの一つである．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は含まれない．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，極小多重汎化を利用することによって正事実からの効率的な帰納推論(学習)を実現する手法について紹介する．

次章で，以下の議論に必要な基本的な用語の定義および表記上の約束を行う．**3章**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対し，その正当性を保証するうえで有用な導モデル保存的具体化の概念およびその性質について述べる．**4章**では極小多重汎化について簡単に紹介し，原始 Prolog のモデルに対する極小多重汎化の性質を与える．**5章**で原始 Prolog に対する最小汎化による導モデル保存的具体化の性質を利用した本体の探索法について述べ，原始 Prolog の多項式時間推論可能性について議論する．最後に，同様の手法を用いて推論できる，より実用的な論理プログラムの部分族について紹介する．なお，紙面の都合により定理などの証明はすべて省略する．詳細については，[Arimura 91b, 有村 92a, Arimura 92b, Ishizaka 92, Ishizaka 93]などを参照されたい．

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引数関数記号とみなす) 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)，論理プログラムなどはすべて $\mathcal{L}$ 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．アトムまたは項のことを**語** (word) と呼ぶ．変数でない語を**非変数語**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 $X_1,\cdots,X_m$ を含む非変数語を $t[X_1,\cdots,X_m]$ で表す*1．すなわち，$\{X_1,\cdots,X_m\}\subseteq var(t[X_1,\cdots,X_m])$ である．$t[X_1,\cdots,X_m]$ 中の変数 X_i に項 r_i を代入して得られる語を $t[r_1,\cdots,r_m]$ で表す．また，語 t のすべての基礎代入例 (ground instance) の集合を $G(t)$ で表す．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．

節 C の頭部を $head(C)$ で表す．以下の制約を満たす論理プログラムを**原始 Prolog** と呼ぶ．

1. 出現する述語記号は1個．
2. たかだか2個の節からなる．
3. 異なる2個の節 C_0, C_1 からなる場合，$G(head(C_0))\cap G(head(C_1))=\phi$ (空集合) である．
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

$$p(t[X_1,\cdots,X_m]) \leftarrow p(X_1),\cdots,p(X_m)$$
$$p(s)$$

の形をした論理プログラムである．ただし，$G(p(t[X_1,\cdots,X_m]))\cap G(p(s))=\phi$ である．

本稿では，論理プログラムを正事実，すなわち，論理プログラムの正しい入出力例だけから効率的に学習する問題を考える．ここでは，その問題を正データからの帰納推論[Angluin 80]の枠組みに基づいて定式化する．「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．

*1　$t[X_1,\cdots,X_m]$ と $t(X_1,\cdots,X_m)$ の記法の違いに注意されたい．後者は相異なる変数 $X_1,\cdots,X_m$ と m 引数関数 (述語) 記号 $t(\_,\cdots,\_)$ から構成される語を表すが，前者は相異なる変数 $X_1,\cdots,X_m$ を含む任意の語を表す．

入力の列 $e_1, e_2, \cdots$ に対して生成される出力の列を $g_1, g_2, \cdots$ とし，$S_i=\{e_1, \cdots, e_i\}$ 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 $e_1, e_2, \cdots$ に対して g_N に**収束**するという．

論理プログラム P の最小 Herbrand モデル（以下では，単にモデルと呼ぶ）を $M(P)$ で表す．論理プログラムの宣言的意味と手続き的意味の等価性より，$M(P)$ は論理プログラム P に関するすべての正しい入出力例の集合と考えられる．$\mathcal{P}$ をある論理プログラムの族とし，$\mathcal{M}=\{M(P)|P\in\mathcal{P}\}$ 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 $e_1, e_2, \cdots$ を M の**正提示**と呼ぶ．モデルの任意の正提示 $e_1, e_2, \cdots$ に対し，推論アルゴリズム $\mathcal{A}$ が生成する出力の列を $P_1, P_2, \cdots$ とし，$S_i=\{e_1, \cdots, e_i\}$ 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 の任意の正提示 $e_1, e_2, \cdots$ 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**正事実から（無矛盾に，保守的に，多項式時間）極限同定**するという．$\mathcal{M}$ を正事実から極限同定する（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**正事実から（無矛盾に，保守的に，多項式時間）推論可能**であるという．

3. 最小汎化と導モデル保存的具体化

本章では，まず，論理プログラムに対する一種のプログラム等価変換である導モデル保存的具体化の概念を導入する．次に，推論対象のモデルの適切な部分集合に対する最小汎化が，この導モデル保存的具体化を導くことを示すことにより，最小汎化や次章で導入する極小多重汎化を用いて節の頭部を推測するような推論手法の正当性を与える．

α を基礎アトム，M を基礎アトムの集合とし，C を $A\leftarrow B_1, \cdots, B_n$ 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*2．

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

C が M において被覆するすべての基礎アトムの集合を $C(M)$ で表し，M に関する C の**導モデル**（derivative model）と呼ぶ．

導モデルの概念は，論理プログラムの不動点意味論を展開する際に導入される写像 T_P の定義のなかに見られる[Lloyd 84]．すなわち，写像 T_P は，導モデルを用いることによって $T_P(I)=\bigcup_{C:=B'} C(I)$ のように定義できる．ただし，I は任意の基礎アトムの集合である．論理プログラム P のモデル $M(P)$ は T_P の最小不動点であるから，$M(P)$ と P 中の節 C の導モデルの間には次の関係が成り立つ．

〈**命題 1**〉 $M(P)=\bigcup_{C:=B'} C(M(P))$. □

論理プログラムの導モデル保存的具体化は以下のように定義される．

【定義 1】 P を論理プログラムとし，C を節とする．このとき，任意の $M\subseteq M(P)$ に対して，$C\theta(M)=C(M)$ を満たす C の代入例 $C\theta$ を，P に関する C の**導モデル保存的具体化**(derivative model preserving instance)と呼ぶ．ただし，θ は適当な代入である． □

【定義 2】 P 中の節 C を P に関する C の導モデル保存的具体化 C' によって置き換えて得られる論理プログラムを P の**導モデル保存的具体化**と呼び，$dmp(P)$ で表す． □

［**例 1**］ 以下の節 C_0, C_1 からなる論理プログラムを P とする．

$$C_0=mem(X, [X|Y]),$$
$$C_1=mem(X, [Y|Z]) \leftarrow mem(X, Z).$$

M を $M(P)$ の任意の部分集合とし

$$C_1'=mem(X, [Y, Z|W]) \leftarrow mem(X, [Z|W])$$

とする．このとき，M の任意の要素 $mem(t, list)$ に対して，$list$ は長さが 1 以上のリストであるから，C_1 が M において被覆するすべての基礎アトムを，C_1' も M において被覆できる．すなわち，$C_1'(M)\supseteq C_1(M)$ である．また，C_1' は C_1 の代入例であるから，$C_1'(M)\subseteq C_1(M)$ が成り立つことは明らか．したがって，C_1 を C_1' で置き換えることによって得られるプログラム $P'=\{C_0, C_1'\}$ は P の導モデル保存的具体化の一つである． □

導モデル保存的具体化は論理プログラムの変換であるが，次の定理はこの変換が論理プログラムのモデルを保存することを保証する．

〈**定理 2**〉 任意の論理プログラム P に対し，$M(P)=M(dmp(P))$. □

ここで，[Plotkin 70]に従って最小汎化の概念を復習しておく．t, s を語とする．$t\theta=s$ なる代入 θ が存在

*2 C の本体が空である（すなわち，$n=0$）場合は，$\alpha=A\theta$ なる代入 θ が存在することをいう．

するとき，t は s(s は t) よりも**一般的（具体的）**であるといい，$t \geq s$ で表す．$t \geq s$ であるが $s \geq t$ でない場合，t は s(s は t) よりも**真に一般的（真に具体的）**であるといい，$t > s$ で表す．また，$t\theta = s$ なる代入 θ が存在しないことを $t \not\geq s$ で表す．一方，$t \geq s$ かつ $s \geq t$ 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，$t > \alpha$ となる語 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 $t' \geq t$ 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[Plotkin 70]．

さて，導モデルの定義より，任意の論理プログラム P と P 中の節 C に対し，C の頭部 $head(C)$ は $C(M(P))$ の汎化であることがわかる．ところが，一般に我々が記述する論理プログラムにおいては，$head(C)$ が $C(M(P))$ の最小汎化よりも真に一般的である場合がある．実際，例1のプログラム P は所属性判定のために我々がよく用いるプログラムであるが，$C_1(M(P))$ の最小汎化は C_1' の頭部である $mem(X, [Y, Z|W])$ となる．一方，例1でも述べたように C_1 を C_1' で置き換えて得られるプログラム P' は P の導モデル保存的具体化であるから，この例の場合，定理2により，C_1 を頭部が $C_1(M(P))$ の最小汎化であるような節 C_1' で置き換えてもモデルは保存されることになる．以下の補題3および定理4は，このことが一般的に成り立つ性質であることを保証している．

〈**補題3**〉 任意の論理プログラム $P=\{C_1, C_2, \cdots, C_m\}$ と $lg(C_i(M(P)))=head(C_i)\theta_i$ なる代入 θ_i ($1 \leq i \leq m$) に対し，P 中の各節 C_i を $C_i\theta_i$ で置き換えることによって得られるプログラム $P'=\{C_1\theta_1, C_2\theta_2, \cdots, C_m\theta_m\}$ は P の導モデル保存的具体化となる． □

上の補題で定義される P の導モデル保存的具体化を**最小汎化による導モデル保存的具体化**と呼び，$dmplg(P)$ で表す．定理2より，次の定理が成り立つ．

〈**定理4**〉 任意の論理プログラム P に対し，$M(P)=M(dmplg(P))$． □

$dmplg(P)$ 中の節の頭部が

$$lg(C_1(M(P))), lg(C_2(M(P))), \cdots, lg(C_m(M(P)))$$

であることに注意されたい．命題1より，このことは，$M(P)$ の適当な部分集合 $C_i(M(P))$ の最小汎化を求めることによって目標のプログラム P と同じモデルを持つプログラムの節の頭部が得られることを意味している．なお，$C_i(M(P))$ は一般に無限集合であるが，[Lassez 88]における無限集合に対する最小汎化の議論から，$C_i(M(P))$ のある適当な有限部分集合の最小汎化を計算することによって $lg(C_i(M(P)))$ を求めることができる．

4. 極小多重汎化

前章で述べたように，有限個の正事実から最小汎化を用いて適切な節の頭部を推測できる可能性があるわけだが，問題は与えられた正事実のなかからいかにして各 $C_i(M(P))$ の部分集合を切り出すかという点にある．本章で述べる極小多重汎化は，この問題に対する極めて有効な枠組みを提供する．

【定義3】 k を非負整数とする．語の集合 S に対し

$$S \subseteq G(t_1) \cup G(t_2) \cup \cdots \cup G(t_m), \quad (1 \leq m \leq k)$$

かつ，任意の

$$S \subseteq G(s_1) \cup G(s_2) \cup \cdots \cup G(s_n), \quad (1 \leq n \leq k)$$

なる $s_1, \cdots, s_n$ に対し

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

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

[Lassez 88]の補題8より，対象となる一階言語が2個以上の関数記号を有する場合，すなわち，$|\Sigma| \geq 2$ の場合，任意の語 t に対して $lg(G(t)) \equiv t$ が成り立つから，上で定義した極小多重汎化は最小汎化の自然な拡張になっている．

図1は最小汎化と極小多重汎化の違いを表したものである．図中の円は一つの語の基礎代入例全体の集合を表す．最小汎化が一個の語で S を覆うのに対し，極小多重汎化は複数の語で覆うことにより，よりむだのない（すなわち，空白の部分が少ない）汎化を実現するのである．

〈**定理5**〉 ある固定された非負整数 k に対し，$|\Sigma| \geq k+1$ と仮定する．任意の基礎語の有限集合 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．[Arimura 91a, Arimura 91b]で与えられたアルゴリズム（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

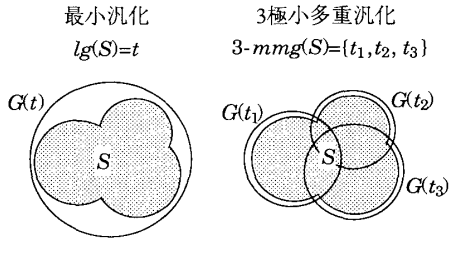


図1 最小汎化と極小多重汎化

関しては保証されていない．しかし，そのなか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 を入力したときの，k-mmg アルゴリズムによるすべての可能な出力の集合を k-$mmg(S)$ で表す．

ここで，原始 Prolog $P=\{C_0, C_1\}$ のモデル $M(P)$ に対する極小多重汎化について考えてみる．ただし

$$C_0=p(s),$$
$$C_1=p(t[X_1, \cdots, X_m]) \leftarrow p(X_1), \cdots, p(X_m)$$

と仮定する．命題1より，$M(P)=C_0(M(P)) \cup C_1(M(P))$ であり，原始 Prolog の定義から，$C_0(M(P)) \cap C_1(M(P))=\phi$ が成り立つ．さらに，C_0 は本体を持たないから $C_0(M(P))=G(p(s))$ である．したがって，[Arimura 91 a] の定理*3により，アトムの対 $\{lg(C_0(M(P))), lg(C_1(M(P)))\}$ は $M(P)$ の2極小多重汎化であることが示せる．さらに，その対は実際に 2-mmg アルゴリズムによって計算可能であることが次の補題で保証される．

〈**補題 6**〉 P を任意の原始 Prolog とし $|\Sigma| \geq 3$ と仮定する．このとき，$S \subseteq S_\Delta$ なる任意の有限集合 $S_\Delta \subseteq M(P)$ に対し，

$$\{lg(C_0(M(P))), lg(C_1(M(P)))\} \in 2\text{-}mmg(S_\Delta)$$

となる有限集合 $S \subseteq M(P)$ が存在する． □

すなわち，原始 Prolog P のモデル $M(P)$ の正提示 $e_1, e_2, \cdots$ が与えられたとき，$S_i=\{e_1, e_2, \cdots, e_i\}$ とすると，ある有限時点 n（上の定理における S に対し，$S \subseteq S_n$ となる時点）以降，$dmplg(P)$ の頭部は常に 2-$mmg(S_i)$ の出力に含まれるのである．このことにより，頭部の推測に関しては一応のめどが立ったことになる．次章では，$dmplg(P)$ の頭部が与えられたときに，その本体を正事実から推測する方法について述べる．

5．正事実からの本体の推測

推論対象の原始 Prolog P が非単一節を含まない場合，すなわち，たかだか2個の単一節 C_i $(0 \leq i \leq 1)$ だけからなる場合，命題1により

$$M(P)=\bigcup_{C_i \in P} C_i(M(P))=\bigcup_{C_i \in P} G(C_i)$$

が成り立つから，$M(P)=P'$ なる P' を推論する問題は，単に $M(P)$ の2極小多重汎化を推論する問題に還元される．この後者の問題は[Arimura 91a]において正事実から無矛盾かつ保守的に多項式時間推論可能であることが示されている．したがって，以下では，目標の原始 Prolog が1個の単一節と1個の非単一節からなる場合を考える．

まず，$dmplg(P)$ の頭部と P 中の非単一節の本体との間に成り立つ関係について述べる．$P=\{C_0, C_1\}$ を原始 Prolog とする．ただし

$$C_0=p(s),$$
$$C_1=p(t[X_1, \cdots, X_m]) \leftarrow p(X_1), \cdots, p(X_m).$$

このとき，次の性質が成り立つ．

〈**補題 7**〉 $|\Sigma| \geq 2$ とする．P の最小汎化による導モデル保存的具体化は $dmplg(P)=\{C_0, C_1\theta\}$ の形で与えられる．ただし，$\theta=\{X_1/r_1, \cdots, X_m/r_m\}$ であり，かつ $r_1 \equiv \cdots \equiv r_m$ である． □

$dmplg(P)$ は P 中の節の代入例によって構成されるわけだが，上の補題はもとのプログラムにおける非単一節の本体に出現する変数だけが代入の対象となること，および代入される項はすべて同一の項の変種であることを示している．すなわち

$$dmplg(P)=\left\{\begin{array}{l} C_0=p(s) \\ C_1\theta=p(t[r_1, \cdots, r_m]) \leftarrow p(r_1), \cdots, p(r_m) \end{array}\right\}$$

かつ $r_i \equiv r_j$ $(1 \leq i, j \leq m)$ である．

さらに，次の定理8により与えられる $dmplg(P)$ 中の節の頭部と本体との間に成り立つ性質により，本体推測のための指針が得られる．

〈**定理 8**〉 任意の $1 \leq i \leq m$ に対し，$p(r_i) \equiv lg(\{p(s), p(t[r_1, \cdots, r_m])\})$． □

すなわち，$dmplg(P)$ の本体を構成する各アトムはその頭部の最小汎化の変種になっているのである．したがって，$dmplg(P)$ の頭部の対 h_0, h_1 が与えられたと仮定すると，それらの最小汎化として得られるアトム $lg(\{h_0, h_1\})=p(r)$ 中の項 r の変種 r_i をその頭部から探し出し，本体に $p(r_i)$ を付加するという方法によって $dmplg(P)$ 全体を推測するという単純な推論手法が考えられる．

前章で述べたように，h_0, h_1 は与えられる正事実の極小多重汎化として得られる．上の定理は，正事実だけからの $dmplg(P)$ 全体の推論可能性を示唆している．もちろん，極小多重汎化の解として得られるアトムの対には，どちらが非単一節の頭部に対応するものであるかという情報は含まれない．また，仮に h_1 が非単一節の頭部であることが判明したとしても，$r_1, \cdots, r_m$ 以外にも r の変種であるような部分項が h_1 中に存在するかもしれない．以下の二つの補題はこの疑問点に対する解答を与える．

〈**補題 9**〉 $P'=\{C_0', head(C_1\theta)\}$ とする．ただし，

*3 $|\Sigma| \geq m+1$ のとき，$G(t) \subseteq G(t_1) \cup \cdots \cup G(t_m)$ ならば，ある i $(1 \leq i \leq m)$ に対して $G(t) \subseteq G(t_i)$ が成り立つ．

$C_0'=p(s)\leftarrow p(r')$ 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．□

〈**補題 10**〉 $P'=\{C_0, C_1'\}$ とする．ただし，$C_1'=p(t[r_1, \cdots, r_m])\leftarrow p(r')$ であり，r' は $r'\equiv r_i$ $(1\leq i\leq m)$ かつ $r'\notin\{r_1, \cdots, r_m\}$ なる $t[r_1, \cdots, r_m]$ の部分項である．このとき，$M(P)-M(P')\neq\phi$ が成り立つ．□

すなわち，補題 9 により，非単一節の頭部たるべきアトムを取り違え，もとのプログラムにおける単一節に対応するアトムに本体を付加したような仮説 P' に対しては，正反例，すなわち正事実でありながら $M(P')$ には含まれないようなものが存在することが保証される．また，補題 10 により，頭部の選択は正しくても本体の選択を誤った場合には，たとえそれが 1 か所であっても，正反例が存在することが保証されるのである．

6. 推論アルゴリズムの概略と問題点

前章までの議論により，原始 Prolog の族を正事実だけから効率的に推論するアルゴリズムを与えるための準備が完了した．本章では，そのようなアルゴリズムの概略を与え，我々がこれまでに証明した結果および未解決の問題点について述べる．

アルゴリズム 1 の 8 行目にある「$S\subseteq M(P)$ なる極大な本体」とは，$t_1, \cdots, t_m$ 以外の任意の $t\in T$ に対し

$$P'=\left\{\begin{matrix}h_i^- \\ h_i\leftarrow p(t_1), \cdots, p(t_m), p(t)\end{matrix}\right\}$$

と置くと，$S\not\subseteq M(P')$ となるような本体 $p(t_1), \cdots, p(t_m)$ のことである．すなわち，仮説 P が与えられた正事実に矛盾しない範囲で可能な限り多くの条件を本体に付加するのである．上のアルゴリズムの概略においては，5 行目と 7 行目に非決定的な選択が存在している．5 行目の選択に関しては，さらに詳細な議論が必要となるので省略する．7 行目における選択に関しては，前節の補題 9 により，誤った選択を行って本体を付加した場合，必ず正反例が存在することが保証されているから，ある時点において $S\subseteq M(P)$ を満たさなくなり，その仮説すなわち本体は却下されることになる．したがって，本体の付加が可能なほうを選択しておけばよい．

```
入力: e1, e2, ...: ある原始 Prolog のモデル M の正提示
出力: P1, P2, ...: 原始 Prolog の代入例の列.
1  S := {};
2  repeat
3      次の正事実 e を読み込む; S := S ∪ {e};
4      2-mmg(S) を計算する;
5      2-mmg(S) の中からアトムの対 {h0, h1} を選択する;
6      lg({h0, h1}) = p(r) を計算する.
7      hi ∈ {h0, h1} を選択し，r の変種であるような hi 中の部分項の集合を T とする.
8      C = hi ← p(t1), ..., p(tm) を，P = {hi⁻, C} に対し，S ⊆ M(P) なる極大な本体
           をもつ節とする．ただし，ti ∈ T かつ hi⁻ ∈ {h0, h1} − {hi} とする.
9      P を出力する.
10 until forever
```

アルゴリズム 1 推論アルゴリズムの概略

次の定理が，原始 Prolog の多項式時間推論に関して，我々がこれまでに証明できた結果である．

〈**定理 11**〉 $|\Sigma|\geq 3$ のとき，任意の原始 Prolog P に対し，$M(P)$ を正事実から多項式時間極限同定する無矛盾な推論アルゴリズムが存在する．□

この結果における推論アルゴリズムに関して注意すべきことは，**2 章**で定義した推論アルゴリズムの性質の一つ，保守性が満たされていないという点である．**2 章**では触れなかったが，多項式時間推論アルゴリズムの正当性のためには，無矛盾性と保守性の両方の条件が満たされることが必要である．なぜなら，そのどちらか一方を無視することによって容易に多項式時間推論を実現する狡猾な手法が存在するためである．

例えば，ここで取り扱った正事実からの推論において保守性を無視することを考えてみる．推論アルゴリズムは推論の各時点において，それまでに受け取った正事実の集合 S をもとに，その時点における仮説を $size(S)$ の多項式時間内に計算しなければならない．狡猾な手法は，ある時点での計算が多項式時間を越えるような場合，とりあえず，ダミーの仮説として $lg(S)$ を出力しておき，S に対する真の仮説の計算をバックグランドで続行する．その計算が終了した時点でおもむろにその結果を出力し，その時点での新たな正事実の集合 S' に対し真の仮説の計算を開始する．このような過程を繰り返すことによって，たとえ，実際の計算時間が指数オーダの推論アルゴリズムであっても，多項式時間で出力するという条件を満足できることになる．$lg(S)$ は過剰に一般化された仮説であり，以後の正事実とは矛盾しないにもかかわらず目標を正しく捉えていないかもしれない．保守性を仮定した場合，そのような仮説を出力することはできない．なぜなら，いったん，過剰一般化された仮説を出力してしまうと，以後，その仮説の変更が許されないため，正しいプログラムに収束するという極限同定の条件を満たせなくなるからである．

もちろん，我々が与えた推論アルゴリズムは，真の仮説の出力を故意に遅らせることによって多項式時間極限同定を実現するものではないが，定理 11 の内容が満足のいく結果でないことも事実である．

7. 文脈自由変換

前章で与えた結果において，保守性が実現できなかった直接の原因には，原始 Prolog のモデルに対する2極小多重汎化の一意性のなさがあげられる．対象となる基礎語の集合が，$G(t) \cup G(s)$ のような語の基礎代入例全体，あるいはその十分に大きな部分集合からなる場合には，極小多重汎化の一意性が保証されるのだが，対象が論理プログラムのモデルのような場合には複数の極小多重汎化が存在する場合がある．本稿では，この問題に関するこれ以上の議論は省略するが，本章で紹介する論理プログラムの部分族 $\mathcal{CFT}_{FB}^{uniq}$ は，その要素のモデルが一意な2極小多重汎化を持つという性質を用いることによって，無矛盾かつ保守的な多項式時間推論アルゴリズムの存在が保証されるプログラム族となっている[有村 92a，Arimura 92b]．

$\mathcal{CFT}_{FB}^{uniq}$ とは以下の条件を満たすたかだか2個の節

$C_0 = p(s_1, \cdots, s_n)$

$C_1 = p(t_1, \cdots, t_n) \leftarrow p(X_1, \cdots, X_n)$

からなる論理プログラム P の族である．

1. 各項 s_i は0引数の関数記号であるか，または変数記号である．
2. 各 X_i は頭部における対応する項 t_i 中に1度だけ出現する（すなわち t_j $(i \neq j)$ には出現しない）変数である．
3. $M(P)$ に対する一意な2極小多重汎化が存在する．

この族は[Shapiro 81]において導入された**文脈自由変換**(context-free transformation)の部分族になっている．また，与えられた論理プログラム P が $\mathcal{CFT}_{FB}^{uniq}$ に属するか否かは $size(P)$ の多項式時間で判定可能である．

［**例2**］ $\mathcal{CFT}_{FB}^{uniq}$ の要素としては，以下のようなプログラムが知られている．

大小関係：$lesseq(X, X)$.
$lesseq(X, s(Y)) \leftarrow lesseq(X, Y)$.

加　　算：$plus(X, 0, X)$.
$plus(X, s(Y), s(Z)) \leftarrow plus(X, Y, Z)$.

リスト連接：$app([\,], X, X)$.
$app([A|X], Y, [A|Z]) \leftarrow app(X, Y, Z)$.

接尾語関係：$suffix(X, X)$.
$suffix(X, [A|Y]) \leftarrow suffix(X, Y)$.

$\mathcal{CFT}_{FB}^{uniq}$ の帰納推論に関しては，次の結果が示せる．

〈**定理12**〉 $|\Sigma| \geq 3$ のとき，任意の $P \in \mathcal{CFT}_{FB}^{uniq}$ に対し，$M(P)$ を正事実から多項式時間極限同定する無矛盾かつ保守的な推論アルゴリズムが存在する． □

すなわち，$app(\_,\_,\_)$ のような単純な論理プログラムは正の具体例だけから効率的に学習可能なのである．

本章の冒頭でも述べたように，原始 Prolog の場合と対照的に保守性まで達成できた理由は，$\mathcal{CFT}_{FB}^{uniq}$ の定義における3番目の条件，すなわち，$M(P)$ に対する2極小多重汎化の一意性にある．この条件により，ある正事実の集合に対する2極小多重汎化 $\{h_0, h_1\}$ で $M(P)$ 全体を覆うようなものはただ一つしかないことが保証される．すなわち，2極多重汎化によって見つかる節の頭部の候補のうち，過剰一般化をもたらす可能性のあるものはただ一つしか存在しないのである．さらに，$\mathcal{CFT}_{FB}^{uniq}$ に属するプログラム P に対しても補題6と同様の結果が成り立つから，その頭部の候補は $dmplg(P)$ の頭部であることが保証される．本体の探索に関しても，**5章**で述べたのと類似の性質が成り立つため，同様の手法によって正しい本体を見つけることができる．

8. おわりに

以上，極小多重汎化を用いた論理プログラムの学習に関して，これまでに我々が得た結果を簡単に紹介した．本稿では，特に，正の具体例だけからの学習に焦点を置いたが，このほかにも正負両方の具体例を用いる学習モデル，あるいは教師への質問が許される学習モデルにおいて極小多重汎化の利用を考えることも興味深い課題である．また，正の具体例だけからの学習を，データベースからの知識の抽出などに応用することも今後の重要な課題である．

◇参考文献◇

[Angluin 80] Angluin, D.: Inductive Inference of Formal Languages from Positive Data, *Information and Control*, Vol. 45, pp. 117-135 (1980).

[Arimura 91a] Arimura, H., Shinohara, T. and Otsuki, S.: Polynomial Time Inference of Unions of Tree Pattern Languages, *Proc. ALT '91*, eds. Arikawa, S., Maruoka, A. and Sato, T., pp. 105-114, Ohmsha (1991).

[Arimura 91b] Arimura, H., Shinohara, T. and Otsuki, S.:

A Polynomial Time Algorithm for Finite Unions of Tree Pattern Languages, *Proc. the 2nd Int. Workshop on Nonmonotonic and Inductive Logics* (1991), To appear in LNCS.

[有村 92 a] 有村, 篠原, 石坂：極小汎化に基づく PROLOG プログラムの正事実からの多項式時間推論, 1992 年度人工知能学会全大, pp. 173-176 (1992).

[Arimura 92 b] Arimura, H., Ishizaka, H. and Shinohara, T.: Polynomial Time Inference of A Subclass of Context-free Transformations, *Proc. COLT'92*, pp. 136-143 (1992).

[Ishizaka 88] Ishizaka, H.: Model Inference Incorporating Generalization, *J. Information Processing*, Vol. 11, No. 3, pp. 206-211 (1988).

[Ishizaka 92] Ishizaka, H., Arimura, H. and Shinohara, T.: Efficient Inductive Inference of Primitive Prologs from Positive Data, *Proc. ALT'92*, eds. Doshita, S., Furukawa, K. and Nishida, T., pp. 135-146 (1992).

[Ishizaka 93] Ishizaka, H.: Generalization and Predicate Invention in Learning Logic Programs, Research Report RR-93-1E, IIAS-SIS, Fujitsu Lab. Ltd. (1993).

[Lassez 88] Lassez, J-L., Maher, M. J. and Marriott, K.: Unification Revisited, *Foundations of Deductive Databases and Logic Programming*, ed. Minker, J., pp. 587-625, Morgan Kaufmann (1988).

[Ling 89] Ling, X.: Learning and Invention of Horn Clause Theories--A Constructive Method, *Methodologies for Intelligent Systems, 4*, ed. Ras, Z. W., pp. 323-331, North-Holland (1989).

[Lloyd 84] Lloyd, J. W.: *Foundations of Logic Programming*, Springer-Verlag (1984).
佐藤雅彦・森下真一 訳：論理プログラミングの基礎, ソフトウェアサイエンスシリーズ, 産業図書 (1987).

[Muggleton 90] Muggleton, S.: Inductive Logic Programming, *Proc. ALT '90*, eds. Arikawa, S., Goto, S., Ohsuga, S. and Yokomori, T., pp. 42-62, Ohmsha (1990).

[Plotkin 70] Plotkin, G. D.: A Note on Inductive Generalization, *Machine Intelligence 5*, eds. Meltzer, B. and Michie, D., pp. 153-163, Edinburgh University Press (1970).

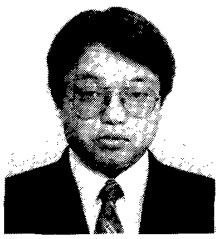
[Shapiro 81] Shapiro, E. Y.: Inductive inference of theories from facts, Technical Report 192, Yale University Computer Science Dept. (1981).
有川節夫 訳：知識の帰納推論, 共立出版 (1986).

[Shapiro 83] Shapiro, E. Y.: *Algorithmic program debugging*, PhD thesis, Yale University Computer Science Dept. (1982), Published by MIT Press (1983).

[Shinohara 90] Shinohara, T.: Inductive Inference of Monotonic Fomal Systems from Positive Data, *Proc. ALT '90*, eds. Arikawa, S., Goto, S., Ohsuga, S. and Yokomori, T., pp. 339-351, Ohmsha (1990).

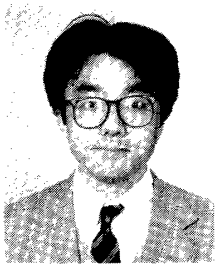
〔担当編集委員：古川康一, 査読者：沼尾正行〕

著者紹介



石坂　裕毅（正会員）

1960 年生まれ. 1984 年九州大学理学部数学科卒業. 1986 年同大学院総合理工学研究科修士課程修了. 同年, 富士通(株)国際情報社会科学研究所入社. 1988 年(財)新世代コンピュータ技術開発機構出向. 1992 年(株)富士通研究所国際情報社会科学研究所復職, 現在に至る. 博士（理学）. 言語の帰納推論, プログラム自動合成などの研究に従事. 1992 年度人工知能学会全国大会優秀論文賞受賞. 情報処理学会会員.



有村　博紀（正会員）

1965 年生まれ. 1988 年九州大学理学部物理学科卒業. 1990 年同大学院総合理工学研究科修士課程修了. 同年, 九州工業大学情報工学部助手, 現在に至る. 論理プログラムおよび計算論的学習理論の研究に従事. 1992 年度人工知能学会全国大会優秀論文賞受賞. 日本ソフトウェア科学会, 情報処理学会各会員.

篠原　武（正会員）

1955 年生まれ. 1980 年京都大学理学部卒業. 1982 年九州大学大学院総合理工学研究科修士課程修了. 同年, 九州大学大型計算機センター助手. 1987 年九州工業大学情報工学部助教授, 現在に至る. 理学博士. 研究分野は, 情報検索, 文字列照合アルゴリズム, および計算論的学習理論などの研究に従事. 1990 年度人工知能学会研究奨励賞, 1992 年度人工知能学会全国大会優秀論文賞受賞. 情報処理学会会員.