ICOT Technical Report: TR-507

TR-507

単純決定性言語の多項式時間学習

石坂　裕毅

September, 1989

ICOT
Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan
(03) 456-3191～5
Telex ICOT J32964

Institute for New Generation Computer Technology

単純決定性言語の多項式時間学習

Polynomial Time Learning of Simple Deterministic Languages

石坂裕毅

Hiroki Ishizaka

ICOT Research Center

Abstract

This paper is concerned with the problem of learning simple deterministic languages. The algorithm described in this paper is essentially based on the theory of model inference given by Shapiro. In our setting, however, nonterminal membership queries, for nonterminals except the start symbol, are not used. Instead of them, extended equivalence queries are used. Nonterminals that are necessary for a correct grammar and their meanings are introduced automatically.

We show an algorithm that, for any simple deterministic language L, outputs a grammar G in 2-standard form, such that $L = L(G)$, using membership and extended equivalence queries. We also show that the algorithm runs in time polynomial in the length of the longest counterexample and the minimum number of nonterminals of a correct grammar.

1 はじめに

本稿では，所属性質問(membership query)と拡張等価性質問(extended equivalence query)を用いて，単純決定性言語(simple deterministic language，以後SDLと略す)を学習する問題について考察する．SDLは，1状態の決定性プッシュダウンオートマトンのプッシュダウンテープを空にする記号列の集合として定義される．SDLのクラスは決定性言語(deterministic language)のクラスの真の部分クラスであり，正則言語のクラスを真に包含する．また，以下で定義されるように，単純決定性文法(simple deterministic grammar，以後SDGと略す)と呼ばれる制限されたGreibach標準形文法によって生成される言語としてSDLを特徴付けることもできる．

Angluin[1]は，等価性質問と非終端記号所属性質問(所属性質問を含む)を用いて，任意のk制限的文脈自由文法を多項式時間で学習するアルゴリズムを示した．本稿で与えられるアルゴリズムは，そのアルゴリズムに基づいている．いずれのアルゴリズムもShapiro[8]のモデル推論において与えられた矛盾点追跡アルゴリズムを利用しており，その意味では，モデル推論の理論に基づいているといえる．

しかし，本稿での設定は，学習アルゴリズムが利用できる質問の種類において，AnguinやShapiroのものと異なっている．すなわち，本稿での設定では，学習アルゴリズムは所属性質問を利用することはできるが，非終端記号所属性質問を利用することはできない．このことは，教師と学習者との対話からは直接観測することのできない，新たな非終端記号をいかにして発見・導入するかという問題をもたらす．

この問題は，事実から一階の理論を学習する際の理論項(theoretical term)[8]の導入問題と関係している．最近，この問題に対していくつかのアプローチが試みられている[4, 7]．しかしながら，ある固定された言語の上での目標の概念に対する有限表現を学習するだけの場合と異なり，その概念を記述

するための言語までも，獲得しなければならないような枠組では，学習アルゴリズムによる学習過程の収束性を保証することが難しくなる．もちろん，学習対象となる概念が，かなり制限された表現系によって表現可能な場合には，そのような枠組での学習アルゴリズムの実現も可能であろう．本稿の結果は，そのようなアルゴリズムの1つを与えている．

本稿での設定のもう1つの特徴は，学習アルゴリズムが拡張等価性質問を利用できる点にある．[3]において定義された等価性質問では，学習対象の概念のクラスに対応する仮説空間の要素のみが提示できる．たとえば，学習対象が言語族 $\{L_1, L_2, \ldots\}$ の場合，通常の等価性質問において提示できる仮説は，$\{L_1, L_2, \ldots\}$ の要素のどれかと等価なものでなければならない．本稿では，この制限を仮定しない．したがって，与えられる学習アルゴリズムが仮説として提示する文法は，必ずしもSDLを生成するとは限らない．実際には，学習の途中で生成される仮説は，2標準形という条件を満たす文法に制限されているだけであり，SDLのクラスを越えるような言語を定義している場合もある．

横森[9]はSDLの多項式時間学習に対して，さらに異なる設定でのアプローチを試みている．その枠組との違いについては最後の節で議論する．

2 準備

まず，本稿で必要となる基本的な概念と記法を[1]および[9]に従って導入する．

2.1 文脈自由文法と文脈自由言語

アルファベットとは，異なる記号の空でない有限集合である．任意のアルファベット X に対し，X 中の記号から成る有限文字列全体の集合を X^* で表す．空な文字列を ε で表し，$X^* - \{\varepsilon\}$ を X^+ で表す．x が文字列の場合，$|x|$ は x の長さを表し，S が集合の場合，$|S|$ は S の要素の個数を表す．

Σ をアルファベットとするとき，Σ 上の**言語** L とは Σ^* の部分集合である．Σ^* の任意の要素 x と Σ 上の任意の言語 L に対し，$x^{-1}L = \{y \mid xy \in L\}$ $(Lx^{-1} = \{y \mid yx \in L\})$ と定義する．$x^{-1}L$ (Lx^{-1}) を L の x に関する**左(右)導言語** (left(right)-derivative) と呼ぶ．任意の文字列 $x = a_1 a_2 \cdots a_n$ に対し，$Pre_i(x)$ で $a_1 a_2 \cdots a_i$ を表し，$Suf_i(x)$ で $a_{i+1} a_{i+2} \cdots a_n$ を表す．

文脈自由文法 G とは，4つ組 (N, Σ, P, S) である．N および Σ はアルファベットである．N の要素を**非終端記号**と呼び，Σ の要素を**終端記号**と呼ぶ．P は $A \to \alpha$ の形をした**生成規則**の集合である(ただし，$A \in N, \alpha \in (N \cup \Sigma)^*$)．$S$ は N の要素であり，**文記号**と呼ばれる．文脈自由文法 G が**Greibach標準形**であるとは，すべての生成規則が $A \to a\alpha$ の形をしていることをいう[1]，ただし，$A \in N, a \in \Sigma, \alpha \in N^*$．Greibach標準形文法で，すべての生成規則の右辺の非終端記号列の長さが高々2であるような文法を**2標準形文法**と呼ぶ．文法 G の**サイズ**を $|N|, |\Sigma|, |P|$ および P 中のすべての生成規則の右辺の長さの総和との合計によって定義する．

$(N \cup \Sigma)^*$ 上の2項関係 $\Rightarrow$ を以下のように定義する．$\beta, \gamma \in (N \cup \Sigma)^*$ に対して，ある $\delta_1, \delta_2 \in (N \cup \Sigma)^*$ および $A \to \alpha \in P$ が存在して，$\beta = \delta_1 A \delta_2$ かつ $\gamma = \delta_1 \alpha \delta_2$ を満たすとき，$\beta \Rightarrow \gamma$．β から γ への**導出** (derivation) とは，各 i に対して，$\beta_i \Rightarrow \beta_{i+1}$ を満たすような有限個の文字列の列 $\beta = \beta_0, \beta_1, \ldots, \beta_n = \gamma$ のことをいう．β から γ への導出が存在するとき，$\beta \Rightarrow^* \gamma$ で表す．すなわち，2項関係 $\Rightarrow^*$ は $\Rightarrow$ の反射的推移閉包を表す．各 β_i の最も左側に出現している非終端記号が置き換えられているような導出のことを**最左導出**と呼ぶ．以下では，最左導出のみを考える．

非終端記号 $A \in N$ の**言語** $L(A)$ を，$L(A) = \{x \in \Sigma^* \mid A \Rightarrow^* x\}$ で定義する．同様に，非終端記号列 $\alpha \in N^*$ に対して，$L(\alpha) = \{x \in \Sigma^* \mid \alpha \Rightarrow^* x\}$ と定義する．特に，対象となる文法を明示する場合は，$S \Rightarrow_G x$ や $L_G(A)$ のように添え字を用いて表す．**文法** G **の言語** $L(G)$ を G の文記号 S に対して $L(S)$ で定義する．

[1] 本稿では，ε を含まない言語のみを対象とする．

2.2 SDGとSDL

文法 G が**単純決定性文法**(SDG)であるとは，G がGreibach標準形であり，かつ任意の $A \in N, a \in \Sigma$ および $\alpha, \beta \in N^*$ に対して，$A \to a\alpha$ と $A \to a\beta$ がともに G の生成規則である場合には $\alpha = \beta$ が成り立つことをいう．Greibach標準形の定義の部分でも注意したように，本稿では，ε 生成規則を含まないようなSDGのみを対象として議論を進める．言語 L が**単純決定性言語**であるとは，$L = L(G)$ なるSDG G が存在することをいう．

たとえば，以下の生成規則をもつSDG $G = (\{S, A, B, C\}, \{a, b\}, P, S)$ は，

$$P = \{S \to aA, A \to b, A \to aB, B \to aBC, B \to bC, C \to b\},$$

SDL $\{a^m b^m \mid 1 \leq m\}$ を生成する．

SDGおよびSDLに関して以下のような命題が成り立つ[5]．

命題 1 $G = (N, \Sigma, P, S)$ を *SDG* とする．任意の $A \in N, x \in \Sigma^+$ および $\alpha \in N^*$ に対し，導出 $A \Rightarrow^* x\alpha$ が存在するならば，$L(\alpha) = xL(A)$ が成り立つ．

命題 2 $G = (N, \Sigma, P, S)$ を *SDG* とする．任意の $A \in N$ に対し，$L(A)$ は *prefix-free* である．すなわち，任意の $x \in L(A)$ と任意の $y \in \Sigma^+$ に対し，$xy \notin L(A)$ である．

命題 3 任意の *SDG* G に対し，G と等価な *2*標準形の *SDG* が存在する．すなわち，次の条件を満たす *SDG* $G' = (N', \Sigma, P', S)$ が存在する．

(1) $L(G) = L(G')$.

(2) P' 中のすべての生成規則は $A \to a, A \to aB, A \to aBC$ のいずれかの形をしている．ただし，$A, B, C \in N', a \in \Sigma$.

命題3により，以下では，SDGとして2標準形のものだけを考える．

2.3 モデルおよび生成規則の正しさ

本稿で与えられるアルゴリズムは，Shapiroのモデル推論アルゴリズム[8]およびAnguinのアルゴリズム[1]に基づいている．それらのアルゴリズムにおいて最も重要な役割を果たしているのは診断ルーチンである．診断ルーチンは，ある負の例(正しくない例)を導くような正しくない仮説の中から，その原因となっている正しくない仮説の構成要素を見つけ出す．ここでは，文法の主要な構成要素である生成規則に対して，“正しい”あるいは“正しくない”といったモデル論的な概念の導入を行なう．

$G = (N, \Sigma, P, S)$ を文脈自由文法とする．各非終端記号 $A \in N$ に対し，Σ^+ の部分集合 $M(A)$ を A の**モデル**と呼ぶ．G の各非終端記号のモデルの集合

$$M = \{M(A_1), M(A_2), \ldots, M(A_{|N|})\}$$

を G の**モデル**と定義する．

置換(replacement)とは，終端記号列 $y_i \in \Sigma^*$ と非終端記号 $A_i \in N$ の対の有限組 $\langle (y_1, A_1), \ldots, (y_n, A_n) \rangle$ (空でもよい)である．置換 $\rho = \langle (y_1, A_1), \ldots, (y_n, A_n) \rangle$ と記号列 $\beta \in (N \cup \Sigma)^*$ に対し，ρ が β に**適合する**(compatible)とは，$\beta = x_0 A_1 x_1 A_2 \cdots A_n x_n$ なる $N+1$ 個の終端記号列 $x_0, \ldots, x_n \in \Sigma^*$ が存在することをいう．空な置換は任意の終端記号列と適合する．ρ が β に適合するとき，β 中の各非終端記号 A_i の出現を y_i で置き換えて得られる終端記号列 $x_0 y_1 x_1 y_2 \cdots y_n x_n$ を ρ による β の**インスタンス**と呼び，$\rho[\beta]$ で表す．

M を G の**モデル**とする．G の生成規則 $A \to \alpha$ が M に関して**正しい**(correct)とは，α に適合する任意の置換 $\rho = \langle (y_1, A_1), \ldots, (y_n, A_n) \rangle$ に対して，$y_i \in M(A_i)$ $(1 \leq i \leq n)$ ならば $\rho[\alpha] \in M(A)$ が成り

立つことをいう[2]．それ以外の場合，すなわち，α に適合する置換 ρ で $y_i \in M(A_i)$ $(1 \leq i \leq n)$ かつ $\rho[\alpha] \notin M(A)$ を満たすものが存在するとき，$A \to \alpha$ は M に関して**正しくない**(incorrect) という．

以上の定義により，以下の命題が成り立つ．

命題 4 $G = (N, \Sigma, P, S)$ を任意の文脈自由文法とする．G の各非終端記号 $A \in N$ に対して，$M(A) = L(A)$ なる G のモデル M とするとき，P 中のすべての生成規則は M に関して正しい．

2.4 質問の種類

L を学習の対象となる未知の SDL とする．学習アルゴリズムに対する教師は L に関する以下の質問に答えられることを仮定する．すなわち，学習アルゴリズムは以下の2つの質問を行なうことが許されている．

(1) 所属性質問：終端記号列 $x \in \Sigma^+$ を提示し，$x \in L$ かどうかを質問する．それに対して，教師は *yes* または *no* で答える．

(2) 拡張等価性質問：2標準形文法 G を提示し，$L(G) = L$ かどうかを質問する．答えは *yes* または *no* であり，*no* の場合には，同時に**反例**(counterexample) が与えられる．反例とは，L と $L(G)$ の対称差(symmetric differece) に含まれているある終端記号列 x である．$x \in L - L(G)$ の場合，x を**正反例**(positive counterexample) と呼び，$x \in L(G) - L$ の場合，**負反例**(negative counterexample) と呼ぶ．反例の選択は任意であるとする．

拡張等価性質問と通常の等価性質問 [3] の違いに注意されたい．等価性質問においては，提示される仮説は，学習対象の概念のクラスのある要素を定義するようなものでなければならない．たとえば，本稿のような SDL の学習の場合，通常の等価性質問によって提示される仮説は，ある SDL を生成するような文法でなければならない．しかし，上の定義にあるように拡張等価性質問によって提示される文法は SDG だけではなく，単なる2標準形文法である．したがって，学習の途中で生成される仮説は一般の CFG である．

Angluin は，等価性質問と所属性質問だけに答えられる教師のことを mininally adequate Teacher[2] と呼んでいる．そこで，本稿では，拡張等価性質問と所属性質問だけに答えられる教師のことを extended minimally adequate Teacher と呼ぶことにする．

3 学習アルゴリズム

以下では，すべての文法は2標準形であるものとする．L をアルゴリズムが学習しようとしている未知の SDL とする．また，$G_0 = (N_0, \Sigma, P_0, S)$ を $L(G_0) = L$ なる SDG の中で，非終端記号の個数に関して極小なものとする[3]．学習アルゴリズムは，終端記号 Σ と文記号 S を知っており，S 以外の非終端記号および生成規則の集合 P を知らないものとする．

本稿の主要な結果は次の定理である．

定理 5 任意の *SDL* L に対し，拡張等価性質問と所属性質問を用い，$|N_0|$ と与えられる反例の最大長との多項式時間で，$L(G) = L$ なる2標準形文法 G を学習するアルゴリズムが存在する．

アルゴリズムによって学習される文法が SDG でなく2標準形文法であることに注意されたい．

---

[2] α が終端記号列の場合には，単に，$\rho[\alpha] = \alpha \in M(A)$ が成り立てばよい．

[3] すなわち，$L(G') = L$ なる任意の SDG $G' = (N', \Sigma, P'S')$ に対して，$|N_0| \leq |N'|$．

3.1 アルゴリズムの概略

まず，非終端記号アルファベット N を $\{S\}$ に初期化し，生成規則集合 P を S だけからなるすべての2標準形生成規則全体の集合に初期化する．初期仮説 G のモデル M としては $\{M(S) = L\}$ を考える．アルゴリズムによって生成される他の非終端記号に関するモデルは次節で与える．

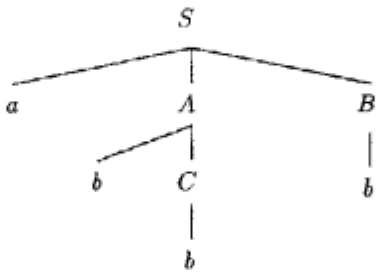
次に，以下のループを繰り返す．現在の仮説 G を提示して拡張等価性質問を行なう．答が *yes*の場合には G を出力して停止する．それ以外の場合には，与えられる反例の仮説上での構文解析を試みる．構文解析に成功した場合，すなわち，与えられた反例が負反例の場合，診断ルーチンを用いてその解析木を診断し，M に関して正しくない生成規則を見つけ出し，それを P から削除する．それ以外の場合，すなわち，与えられた反例が正反例の場合，新たな非終端記号を生成し，それらから構成可能な新たな生成規則のすべてを P に付加する．

学習アルゴリズム

Given: An extended minimally adequate Teacher for L and a terminal alphabet Σ.
Output: A grammar $G = (N, \Sigma, P, S)$ in 2-standard form such that $L(G) = L$.
Procedure:
$N := \{S\}$. $P := \{S \to aSS, S \to aS, S \to a | a \in \Sigma\}$. $G := (N, \Sigma, P, S)$.
repeat
  Make an extended equivalence query with G.
  If the reply is positive counterexample, *then*
    introduce new nonterminals with their models.
    Put all candidate productions into P.
  Else if the reply is negative counterexample, *then*
    diagnose G.
    Remove the incorrect production replied by the diagnosis routine from P.
until the reply is *yes*.
Output G.

以下では，2標準形文法 G に対して，G のサイズと入力文字列 w の長さの多項式時間で動作する構文解析手続きが存在することを仮定する(たとえば，Angluin[1] の構文解析手続き[4])．

診断ルーチンは，非終端記号 A から $w \notin M(A)$ なる文字列 w を生成しているような構文解析木を入力とし，M において正しくない P 中の生成規則を出力する．たとえば，負反例 $abbb$ に対する以下のような構文解析木が与えられたとしよう．



まず，$abbb \notin M(S) = L$ ということが分かっている．診断ルーチンは，非終端記号を根とする各部分

[4] G は2標準形であるから，[1] の補題3および補題4が成り立つ．実際には，Angluin の手続きは parse-DAG (directed acyclic graph) を出力するが，以下の議論には影響を与えない

木に対し，その部分木が生成している終端記号列が，実際にその非終端記号のモデルの要素であるか否かをテストする．たとえば，上の例では，Aを根とする部分木に対し，$bb \in M(A)$か否かをテストする．$bb \notin M(A)$の場合には，その部分木を入力として再帰的に探索を続ける．$bb \in M(A)$の場合には，Sの次の子，すなわちBを根とする部分木に対して同様のテストを行なう．このとき$b \notin M(B)$であるならば，この導出に対応する生成規則$B \rightarrow b$がMに関して正しくなく，$b \in M(B)$であるならば，上位の生成規則$S \rightarrow aAB$が正しくない[5]．

[1]では，上記のテスト($bb \in M(A)$等)は，教師がある正しい文法を知っているという仮定のもとで，$M(A) = L(A)$として非終端記号所属性質問$bb \in L(A)$を行なうことによって達成される．しかし，SDLの学習においては，それらのテストを有限回の所属性質問によって達成することができる．次節で，それらの代替可能性およびそのための非終端記号の生成法について述べる．

学習のある時点において生成された新たな非終端記号の集合をNewとする．アルゴリズムはNを$N \cup New$に変更する．Nから構成可能な2標準形生成規則の中でPに1度も現れていないようなものの集合をP_{New}とする．すなわち，P_{New}は，各$a \in \Sigma$に対し，次の条件を満たすような生成規則$A \rightarrow a\alpha$全体の集合である．

1. $A\alpha \in N^+$, $|\alpha| \leq 2$.
2. $A\alpha$はNewの要素を少なくとも1つ含む．

命題 6 P_{New}はNの多項式時間で計算可能であり，学習の任意の時点において，Pは高々$|N| \times |\Sigma| \times (|N|+1)^2$個の生成規則しか含まない．

3.2 非終端記号の生成とそれらのモデル

ここで与えられる非終端記号およびそれらのモデルの生成法は，[6]における拡張モデルの概念を拡張したものである．

まず，非終端記号生成にとって最も重要となるSDGの特徴を与える．

補題 7 $G = (N, \Sigma, P, S)$を任意のSDGとする．$A, B \in N, \alpha \in N^*, r \in \Sigma^+$に対して，$A \Rightarrow^* rB\alpha$なる導出が存在すると仮定する．また，$t \in L(\alpha)$を，任意の$j$ $(1 \leq j \leq |t|-1)$に対して，$Suf_j(t) \notin L(\alpha)$を満たすような終端記号列とする($\alpha = \varepsilon$の場合は，$t = \varepsilon$)．このとき，任意の$x \in \Sigma^+$に対して，$x \in L(B)$であるための必要十分条件は，(i) $rxt \in L(A)$かつ(ii) $rPre_i(x)t \notin L(A)$ $(1 \leq i \leq |x|-1)$が成り立つことである．

[証明] $x \in L(B)$と仮定すると，$A \Rightarrow^* rB\alpha \Rightarrow^* rx\alpha \Rightarrow^* rxt$であるから，$rxt \in L(A)$. 補題2より，$L(B)$はprefix-freeであるから，任意の$i$ $(1 \leq i \leq |x|-1)$に対して，$Pre_i(x) \notin L(B)$である．したがって，$rPre_i(x)t \in L(A)$，すなわち，$Pre_i(x)t \in \overline{r}L(A) = L(B\alpha)$であるならば，$Pre_i(x)Pre_j(t) \in L(B)$かつ$Suf_j(t) \in L(\alpha)$なる$j$ $(1 \leq j \leq |t|-1)$が存在し，仮定に反する．したがって，$rPre_i(x)t \notin L(A)$ $(1 \leq i \leq |x|-1)$.

逆に，(i)と(ii)が成り立つと仮定する．(i)により，$xt \in \overline{r}L(A) = L(B\alpha)$が成り立つ．任意の$j$ $(1 \leq j \leq |t|-1)$に対して，$Suf_j(t) \notin L(\alpha)$であるから，$Pre_k(x) \in L(B)$かつ$Suf_k(x)t \in L(\alpha)$なる$k$ $(1 \leq k \leq |x|)$が存在する．一方，(ii)により，任意のi $(1 \leq i \leq |x|-1)$に対し，$Pre_i(x)t \notin L(B\alpha)$であるから，任意の$i$ $(1 \leq i \leq |x|-1)$に対して，$Pre_i(x) \notin L(B)$である．したがって，$k = |x|$，すなわち，$Pre_{|x|}(x) = x \in L(B)$である． [証明終わり]

アルゴリズムにあるように，新たな非終端記号が生成されるのは，ある正反例wが与えられた場合である．非終端記号生成ルーチンは，その正反例をもとに非終端記号を生成するのである．

[5] 置換$\rho = \langle (bb, A), (b, B) \rangle$に対し，$bb \in M(A), b \in M(B)$かつ$\rho[aAB] = abbb \notin M(S)$である．

w を $|w| \geq 2$ なる任意の正反例とする．正反例 w から生成される非終端記号アルファベット $N(w)$ を以下のように定義する．

$$N(w) = \{(r,s,t) \mid r,s \in \Sigma^+, t \in \Sigma^* \text{ かつ} rst = w\}.$$

命題 8 $|N(w)| \leq |w|(|w|-1)/2$ であり，$N(w)$ は $|w|$ の多項式時間で計算できる．

たとえば，$w = aabb$ の場合，

$$N(w) = \{(a, abb, \varepsilon), (a, ab, b), (a, a, bb), (aa, bb, \varepsilon), (aa, b, b), (aab, b, \varepsilon)\}$$

となる．

$N(w)$ の各要素 (r,s,t) に対して，$\overline{rs}L$ に属する t の接尾語で最短なものを $\varphi(r,s,t)$ で表す，すなわち，

$$\varphi(r,s,t) = Suf_i(t) \quad \text{ただし} \quad i = \max_{0 \leq j \leq |t|-1} \{j \mid Suf_j(t) \in \overline{rs}L\}.$$

命題 9 $\varphi(r,s,t)$ は高々 t 回の所属性質問，すなわち，"$rsSuf_j(t) \in L$?"$(0 \leq j \leq |t|-1)$ を行なうことによって決定できる．

$N(w)$ 中の各非終端記号 (r,s,t) のモデルは以下のように定義する．

$$M((r,s,t)) = \{x \in \Sigma^+ \mid rx\varphi(r,s,t) \in L \text{ and } rPre_i(x)\varphi(r,s,t) \notin L \text{ for any } i\ (1 \leq i \leq |x|-1)\}.$$

補題 10 N を既知の非終端記号アルファベットとし，w を新たに与えられた正反例とする．このとき，新たな非終端記号および新たな生成規則の導入に要する時間は $|N|$ と $|w|$ の多項式で抑えられる．

[証明] 命題6，命題8 および命題9 より明らか． [証明終わり]

補題 11 L を任意の SDL，w を L の要素，$G = (N, \Sigma, P, S)$ を $L(G) = L$ なる任意の SDG とする．$S \Rightarrow^* w$ の導出に現れる任意の非終端記号 $A \in N - \{S\}$ に対し，$L(A) = M((r,s,t))$ なる非終端記号 $(r,s,t) \in N(w)$ が少なくとも1つ存在する．

[証明] $S \Rightarrow^* rA\alpha \Rightarrow^* rs\alpha \Rightarrow^* rst = w$ と仮定する．$N(w)$ の定義より，3つ組 (r,s,t) は $N(w)$ の中に存在する (G はSDGであり，$A \neq S$ であるから，r も s も ε でない)．

一方，$L(S) = L(G) = L$ であるから，命題1 により，$L(\alpha) = \overline{rs}L(S) = \overline{rs}L$ である．$\varphi(r,s,t)$ の定義より，$\varphi(r,s,t) \in L(\alpha)$ かつ任意の j $(1 \leq j \leq |\varphi(r,s,t)|-1)$ に対して，$Suf_j(\varphi(r,s,t)) \notin L(\alpha)$ である．したがって，補題7 と $M((r,s,t))$ の定義より $L(A) = M((r,s,t))$ が成り立つ． [証明終わり]

上の補題は，ある正反例 w が与えられた場合，学習アルゴリズムが，常に，w を生成するのに必要なすべての非終端記号を生成できることを保証している．結果として，[1]で用いられた非終端記号所属性質問を所属性質問に置き換えることができる．すなわち，任意の $x \in \Sigma^*$ と任意の $A \in N(w)$ に対して，$x \in M(A)$ であるか否かは高々 x 回の所属性質問を行なうことによって確かめることができるのである．

3.3 アルゴリズムの正当性と複雑さ

以下では，L を学習対象の言語とし，$G_0 = (N_0, \Sigma, P_0, S)$ を $L(G_0) = L$ なる非終端記号の個数に関して極小なSDGとする．さらに，$G = (N, \Sigma, P, S)$ をアルゴリズムの仮説とし，$M = \{M(S), M((r_1,s_1,t_1)), \ldots, M((r_{|N|-1}, s_{|N|-1}, t_{|N|-1}))\}$ を前節で定義した G のモデルとする．

補題 12 診断ルーチンへの入力として，$A \in N$ を根とし，$x \notin M(A)$ なる終端記号列 x を生成するような G 上での構文解析木が与えられたとき，診断ルーチンは M に関して正しくない P 中の生成規則を出力する．

[証明]　[1] における補題 5 の証明と同様な議論によって証明できる．　[証明終わり]

補題 13 負反例 w に対する構文解析木を入力としたとき，診断ルーチンの要する時間は $|w|$ とその時点までに与えられた反例の最大長 ℓ との多項式で抑えられる．

[証明]　G は 2 標準形であるから，与えられる構文解析木には高々 $|w|$ 個の非終端記号しか現れない．したがって，高々 $|w|$ 回の非終端記号所属性に関するテストが行われる．$x \in M(A)$ であるか否かを調べる各テストに対して，$A = S$ の場合には，1 回の所属性質問 "$x \in L$?" を行なえばよい．それ以外の場合，すなわち，$A = (r, s, t)$ の場合には，高々 $|x|$ 回の所属性質問 "$rPre_i(x)\varphi(r, s, t) \in L$?" $(1 \leq i \leq |x|)$ を行なえばよい．x は w の部分文字列であるから，診断ルーチンが 1 回の診断に行なう所属性質問の総数は高々 $|w|^2$ である．診断ルーチンが行なう主なオペレーションは，終端記号列 $rPre_i(x)\varphi(r, s, t)$ を生成することと所属性質問を行なうことであるから，定理の主張が成り立つ．　[証明終わり]

補題 14 与えられる正反例の個数は高々 $|N_0|$ である．

[証明]　n 番目に与えられた正反例を w_n とする．w_n の導出に関与する N_0 の部分集合を $N_0(w_n)$ とし，それまでに与えられた正反例の導出に関与する N_0 中の非終端記号だけを含む P_0 中の生成規則の集合を $P_0(w_n)$ とする．すなわち，

$$N_0(w_n) = \{A \in N_0 \mid S \Rightarrow_{G_0}^{*} uA\alpha \Rightarrow_{G_0}^{*} w_n\},$$
$$P_0(w_n) = \{A \to a\alpha \in P_0 \mid a \in \Sigma, A\alpha \in (\bigcup_{i=1}^{n} N_0(w_i))^{+}\}.$$

w_n が与えられると，アルゴリズムは $N(w_n)$ を計算して N に付加し，3.1 節で述べたように，新たな生成規則の候補を計算して P に付加する．

補題 11 により，$N_0(w_n)$ 中の任意の非終端記号 A に対して，$N(w_n)$ のある要素 A' が存在して $L(A) = M(A')$ を満たす．この非終端記号間の対応関係のもとで，$P_0(w_n)$ 中の各生成規則に対応する N 上の生成規則が少なくとも 1 度 P に付加される．命題 4 により，それら対応する生成規則は M に関して正しい．補題 12 により，正しい規則が P から削除されることはない．したがって，$n+1$ 番目の正反例が与えられる場合には，常に，

$$A \in N_0(w_{n+1}) \quad \text{かつ} \quad A \notin \bigcup_{i=1}^{n} N_0(w_i).$$

なる非終端記号 $A \in N_0$ が存在する．よって，与えられる正反例の個数は $|N_0|$ を越えない．[証明終わり]

補題 15 アルゴリズムが導入する非終端記号の総数は高々 $|N_0|\ell_p(\ell_p - 1)/2$ である．ただし，ℓ_p は与えられる正反例の最大長．

[証明]　各正反例 w_i に対して，$|N(w_i)|$ は高々 $|w_i|(|w_i| - 1)/2$ である．補題 14 により，与えられる正反例の個数は $|N_0|$ を越えないから，導入される非終端記号の総数は高々 $|N_0|\ell_p(\ell_p - 1)/2$ である．

[証明終わり]

[定理5の証明] 生成規則の作り方および補題15より，Pに付加される生成規則の個数は高々

$$m = \frac{|N_0|\ell_p(\ell_p-1)}{2} \times |\Sigma| \times \left(\frac{|N_0|\ell_p(\ell_p-1)}{2}+1\right)^2.$$

である．補題12により，負反例が与えられるごとに少なくとも1つの正しくない生成規則が見つかり，Pから削除される．したがって，補題14によって，高々$|N_0|$個の正反例と高々m個の負反例が与えられた後，アルゴリズムは停止して$L(G)=L$なる2標準形文法を出力することが保証される．

ℓを与えられる反例の最大長とする．補題15により，学習プロセスの任意の時点において，Gのサイズは$|N_0|$とℓの多項式で抑えられている．構文解析ルーチンに関する仮定により，アルゴリズムは，与えられた反例が正か負かを$|N_0|$とℓの多項式時間で判定できる．与えられる反例の総数は高々$|N_0|+m$であるから，補題10および補題 13より，アルゴリズム全体の時間計算量は$|N_0|$とℓの多項式で抑えられる． [証明終わり]

4 むすび

単純決定性言語の多項式時間学習について考察を行なった．主要な問題点は，必要な非終端記号の生成と，それらに対する適切なモデルの与え方にあった．ある目標の概念を表現するのに必要(あるいは有用)ではあるが，直接観測されることはない補助的概念を導入する問題は，機械学習におけるもっとも本質的かつ難しい問題である．最近，この問題に対するいくつかのアプローチが試みられている[4, 7]．しかし，十分満足できる結果が得られているとは言い難い．もちろん，本稿で与えた結果自身も非常に制限されたクラスに対するものであり，さらに，一般的かつ有用な手法の開発が必要である．

アルゴリズム自体に関しては，一応，多項式時間学習が保証されてはいるが，かなり次数の高い多項式であり現実的ではない．これは，生成される仮説をSDGに制限していないことによる．もしも，アルゴリズムが生成する各時点での仮説をSDGに制限できるならば，計算量の上での大幅な改善が期待できる．また，当然のことながら，そのような制限は等価性質問に関する拡張を不要とするわけであり，理想的なminimally adequate Teacherからの学習モデルを実現することにもつながる．

横森[9]は，任意のSDL Lに対し，$L(G)=L$なるSDG Gを多項式時間で出力するアルゴリズムを与えている．しかし，本稿での枠組と違う点は，非常に強力な教師が仮定されていることである．教師は2種類の特殊な質問，接頭語所属性質問(prefix membership query)および導言語等価性質問(derivatives equivalence query)に答えることが要求される．接頭語所属性質問は，終端記号列xを提示し，$xy\in L$なる$y\in\Sigma^*$が存在するか否かを問う．それに対する答えは，yesまたはnoであり，yesの場合には，$xy\in L$なる最短のyを教えることが仮定される．導言語等価性質問は，2組の終端記号列の対$(u_1,v_1),(u_2,v_2)$を提示し，$\overline{u_1}L\overline{v_1}=\overline{u_2}L\overline{v_2}$であるか否かを質問する．それに対する答えは，$yes$または$no$である．導言語等価性質問は我々のアルゴリズムにおいて，生成された2つの非終端記号の等価性を判定するのに利用できる．たとえば，非終端記号(u_1,v_1,w_1)と(u_2,v_2,w_2)に対して，$\overline{u_1}L\overline{w_1}=\overline{u_2}L\overline{w_2}$であるならば，それらは同一視してよい．したがって，冗長な非終端記号の生成を抑えることができ効率の改善が期待できる．そのような教師の能力と学習可能性および学習効率の関連については，Angulinが[3]の中で多くの興味深い結果を報告している．

謝辞

本研究に際して，多くの御助力を頂いた古川康一ICOT研究所次長ならびに長谷川隆三ICOT第1研究室長に深謝致します．貴重な時間を割いて議論して頂き多くの有益な御助言を頂いた電通大・横森貴助教授ならびに初稿における等価性質問の定義に関する誤りを指摘して頂いた富士通国際研・榊原

康文氏に感謝します．また，“学習と非単調推論に関するセミナー”を通じて議論して頂いたICOT同僚諸氏に感謝します．最後に，本研究の機会を与えて頂いた渕一博ICOT研究所長に感謝致します．

参考文献

[1] D. Angluin. Learning k-bounded context-free grammars. Research Report 557, Yale University Computer Science Dept., 1987.

[2] D. Angluin. Learning regular sets from queries and counterexamples. *Information and Computation*, 75:87–106, 1987.

[3] D. Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1988.

[4] R. B. Banerji. Learning theories in a subset of a polyadic logic. In *Proc. Computational Learning Theory '88*, pp. 281–295, 1988.

[5] M. A. Harrison. *Introduction to Formal Language Theory*. Addison-Wesley, 1979.

[6] H. Ishizaka. Inductive inference of regular languages based on model inference. To appear in IJCM, 1989.

[7] S. Muggleton and W. Buntine. Machine invention of first-order predicates by inverting resolution. In *Proc. 5th International Conference on Machine Learning*, pp. 339–352, 1988.

[8] 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).

[9] T. Yokomori. Learning simple languages in polynomial time. In *Proc. of SIG-FAI*, pp. 21–30. Japanese Society for Artificial Intelligence, June 1988.