文法推論のためのホーン節の帰納推論

文法推論のためのホーン節の帰納推論

花川賢治*

Induction of Horn Clauses for Grammatical Inference

Kenji HANAKAWA*

ABSTRACT

Grammatical inference from positive data is a computing technique for learning languages or patterns from valid examples. This technique is widely applicable to acquisition of knowledge from information space, such as Internet. Some heuristics based grammatical inference systems have been developed by researchers. However, these systems use ad hoc inference rules, which are depend on several researchers. This paper proposes a grammatical inference system based on induction of Horn clause. Horn clause is a logical form which is useful for both representing grammar and deductive inference. It can be also used for inductive inference by rewriting of a Horn clause set. The system rewrites a Horn clause set according to three induction rules, which are produced by logical operations from provable formulas. This paper describes the basic implementation of the system, and shows how the system can learn regular grammar.

Key Word: Grammatical inference, Induction, Learning, Definite clause grammar

1 はじめに

ある機械があり，それから “aa”, “aaaa”, “aaaaaaaa” という 3 種類の文字列が出力されたとすると，大抵の人は「この機械は偶数個の a からなる文字列を出力する機械であり，“aaaaaa”, “aaaaaaaaaa” も出力するに違いない」と推測するだろう．本稿では，このような推測と同様のことをコンピュータにさせる方法について論じる．

上で使った言葉をコンピュータ科学の言葉に置き換えると，機械は「文法」，観測された 3 種類の文字列は「例」，機械が出力するすべての文字列の集合は「言語」になる．このような言語の部分集合である例から言語あるいは文法を同定する問題は，文法推論と呼ばれ盛んに研究されてきた．

文法推論は，正例 (言語に含まれる文字列) と負例 (言語に含まれない文字列) の両方を使う推論と，正例のみを使う推論の二種類に大別される．Gold は正例のみを使う推論では非常に簡単な言語でさえ学習が不可能であることを理論的に証明した [4]．しかしながら，現実の問題では負例を用意するのは困難な場合が多いので，正例だけからの文法推論の方が実用的な価値が高く，その研究は続けられている．正例だけからの文法推論の実現性についての議論には，人間の子供の学習が参考にされることがあった．人間の子供は負例すなわち誤った文を示されることがほとんどないのにもかかわらず母国語の学習が可能であることが心理学者から報告されている．ある研究者たちはこの事実を Chomsky がとなえた人間の言語能力の生得説の裏付けであるとし，別の研究者たちは正例だけからの文法推論の可能性が否定されるべきではないと考えた．

著者は，人間が同様の推論を行えることから，ある程度の正例からの文法の学習は可能であると考えている．本研究では，正例だけからの文法推論の可能性を示す目的で，ホーン節の書き換えに基づく文法推論システムを開発し，そのシステムが簡単な言語の学習ができることを示す．

ホーン節とは形式論理で使われる論理式の形式の一つであり，ホーン節の書き換えとは帰納推論の手法の一つである．帰納推論とは，われわれが経験から様々なことを学習する際に行っている例からの一般化を形式化したものである．推論には，帰納推論の他に演繹推論がある．通常，推論というと演繹推論のことを指す．以下に，我々人間が普段行っている推論例を示す．ただし A,B,C は論理式，→は含意 (ならば)，⊢は推論で導かれることを表している．

1998 年 4 月 9 日受理

* 電子情報工学科 (Department of Electrical Engineering and Computer Science)

A と (A ならば B) ということから B を導く (3 段論法):

$A \quad A \rightarrow B \vdash B$

B と (A ならば B) ということから A を導く:

$B \quad A \rightarrow B \vdash A$

後者は常に推論結果が正しいとは限らない誤った推論であるが，人間はこのような推論を使って有効に問題を解決しているのも事実である．本研究では，推論結果が常に正しいとは限らないが正しいことが多い推論を，それを指示する論理式の組が多数あるばあいに適用することで帰納推論が可能であると考える．この考えに基づき，本稿で述べる文法推論システムは，最初に与えられた正例だけを受理する文法をホーン節の形式でデータベースに登録し，帰納推論を適用することによりホーン節集合を少しずつ書き換えていくことで文法の一般化を行う．

以下，**2** 章では，過去の文法推論の研究と，今後期待される文法推論の応用について述べる．**3** 節では，ホーン節の書き換えによる文法推論の基本的な考え方とそれに基づいた文法推論システムについて説明する．**4** 節では，簡単な正規文法について，このシステムが正しく推論できるかどうかを実験的に調べる．**5** 節では，文脈自由文法の学習についての実験結果を示し，**6** 節では，**4** 節の例題よりも実用的な推論を行うための拡張について述べる．**7** 節で結論と今後の課題について述べる．

2 文法推論

2.1 過去の研究

まず，コンピュータ科学での「言語」について簡単に説明する．言語には，人間が日常的に使う「自然言語」とコンピュータのプログラミングなどに使われる「人工言語」がある．これらの言語を抽象化したものが「形式言語」である．

形式言語は次のように定義される．

1. 終端記号 —— その言語の単語
2. 非終端記号 —— その言語の構文的範疇
3. 生成規則 —— その言語の構造の規則
4. 開始記号 —— 生成規則による書き換えの出発点

以下に簡単な言語の例を示す．この例は a のあとに b が一つ以上繰り返される文字列を表す．

1. 終端記号：a, b
2. 非終端記号：S, B
3. 生成規則：$S \rightarrow aB$, $B \rightarrow bB$, $B \rightarrow b$
4. 開始記号：S

表 1: 主な形式文法のクラスとその性質

文法のクラス	許される書き換え規則	表現力
正規文法	$A \rightarrow aB$ $A \rightarrow a$	連接、選択、閉包からなる言語が表現できる
文脈自由文法	$A \rightarrow \beta$	自然言語に近いレベルの言語が表現できる
文脈依存文法	$\alpha \rightarrow \beta$ $\lvert\alpha\rvert \leq \lvert\beta\rvert$	強力
句構造文法	$\alpha \rightarrow \beta$	文脈依存文法よりさらに強力

A, B：非終端記号，a：終端記号，α, β：終端または非終端記号の長さ 1 以上の系列

正規文法については，グラフ形式 (決定性オートマトン) でも記述可能である．この他に本稿でも用いる論理式で記述する方法がある．この方法については **3** 節で述べる．

形式言語はその能力により，低いものから正規言語，文脈自由言語，文脈依存言語，句構造言語にクラス分けされる．自然言語はほぼ文脈自由言語に属し，ほとんどのコンピュータのプログラミング言語は完全に文脈自由言語に属すると言われている．正規文法は文字列の検索パターンを表現するのによく用いられる．形式言語に関する詳しい説明は参考文献 [7] を参照されたい．本研究は基本的には正規言語を対象としているが，一部文脈自由言語を対象とした実験結果についても述べる．

文法推論とは，言語の学習であり．帰納推論の一種である [10]．コンピュータに学習をさせる方法には，帰納推論のような記号処理に基づく方法の他に，ニューロコンピュータに代表されるような数値処理に基づく方法がある．両者は対照的な特徴を持つ技術であり，様々な相違があるが，そのひとつに記号処理に基づく学習は人間の思考過程と対応づけて理解し易いのに対し，数値処理に基づく学習はそれが容易でないという違いがある．たとえば，一般人にはニューロコンピュータの学習結果であるシナプスの重みの値が何を意味するのか理解することはできない．したがって，人間が応用できるような知識を獲得するには，記号処理に基づく方が有利であると思われる．しかし，記号処理に基づく学習は，連続的な学習対象には適さないため応用範囲

は限られるという欠点もある．現在，帰納推論の研究としては文法推論の他にプログラムの自動生成，データからの概念獲得などがあるが比較的範囲が限られている．

過去の文法推論の研究は，理論指向の研究と実用指向の研究に大きく分類することができる．実際の研究が明確にどちらかに分類できるわけではないが，研究の興味とアプローチに二つの異なった方向がある．

理論指向の文法推論の研究は広範囲の研究者により研究され，成果が充実している．現在のところ著者が把握できているのは比較的良く知られた一部分のみであるが，それを整理すると以下のようになる．

1. プログラムが自由に，ある記号列が言語に属するか否か，ある文法が正しいか否か質問できるならば，正規言語の効率的な学習は可能である [2].

2. 正例からだけでは正規言語でさえ学習は不可能である [4].

3. 正例だけからの学習が可能な興味深い言語のクラスがいくつか存在する [1, 8].

ここでの効率的な学習が可能とは，正しい文法を獲得するための多項式時間のアルゴリズムが存在するという意味であり，学習が可能とは，文法が同定できたことを判定するアルゴリズムが存在するという意味である．**1** の質問は人間がいちいち答えてやらねばならないので，この方法の文法推論が応用できるシナリオはかなり限定されるだろう．**3** の正例だけからの学習が可能な言語のクラスは理論的に導かれたもので，われわれがよく使用する言語のほとんどは表現できない．これらの研究成果を総合すると，文法推論の広範な応用は難しいということになる．

実用指向の研究では，実際に文法推論ができるシステムを構成することが目的とされ，文法の一般化を行うヒューリスティクスの収集に重点が置かれた．実用指向の研究でも，負例を使った学習も行われたが，理論的に不可能とされていた正例だけからの正規言語および文脈自由言語の学習も研究された [3].

一見すると両者の研究は矛盾し合うように思える．しかし，両者の主張の違いは問題設定の相違が原因にあり，互いに否定し合う関係にはない．つまり，理論指向の研究は一意の言語を求める問題を扱っているのに対し，実用指向の研究は必ずしも学習される言語が一意であることを要求しないという根本的な違いがある．他にも，後者は学習される文法が簡単であることを暗黙的に仮定しているようである．たとえば，生成規則の数に上限が与えられたならば文法推論は易しくなるが，実用での問題はそれに近いと考えられる．

実用指向の研究の問題はヒューリスティクスの収集の方法が研究者の経験に依存した場当たり的なものであり，必然的に不十分なものにならざるを得ないという点である．本研究では，この点を克服するために論理に基づいた方法を提案する．いずれにせよ，これまでの研究では実用に耐えるような文法推論システムを開発することには十分に成功していない．

2.2　応用の可能性

現状では，文法推論の応用については，入力項目をユーザに示す機能を持つデータエントリシステム，正規表現を書く代わりに文字列の変換例を入力する初心者向けエディタ，プログラミング言語の設計を支援するシステムなど，比較的低いクラスの小規模な文法の学習が研究されている [9].

文法推論には多くの可能性が秘められている．著者が最も有望と考えているのはデータマイニングへの応用である．現在のデータマイニングはデータベースからの知識の自動獲得が主流をしめるが，今後自然言語文書を源とする知識の自動獲得の必要性が高まるであろう．文書には，明示的に表現された知識と，文書の表現上の制約として現れる暗黙的な知識が存在していると考えることができる．文法推論は後者の知識の獲得に有効である．つまり，自然言語文に含まれる単語を終端記号に対応させ文法推論を行うことにより一般化された非終端記号が得られ，それを人間が所有する概念に対応づけることができるのである．このばあいの記号は，統語的なカテゴリではなく，意味的なカテゴリとして捉えられる．一般化のレベルを変えることにより，非終端記号の包含関係が獲得でき，概念体系の自動獲得が実現される．このようにして得られる概念体系は人手により作成した辞書，シソーラスあるいは知識ベース等と比較して緻密さ，正確性，一般性は劣るが，それでも有効な利用方法はあると考えられる [5]．この技術を確立することにより，低コストで既存の大量データを活用するという新しい情報処理の形態が発展することが期待できる．ただし，このような応用を実現するには，文脈自由言語の効率的な学習が必要になる．

表 2: 生成規則と確定節文法

生成規則	Prolog	引数を省略したProlog
$S \rightarrow NP\ VP$	$s(X,Z) : -np(X,Y), vp(Y,Z).$	$s \leftarrow np \wedge vp$
$NP \rightarrow DT\ N$	$np(X,Z) : -dt(X,Y), n(Y,Z).$	$np \leftarrow dt \wedge n$
$VP \rightarrow V\ NP$	$vp(X,Z) : -v(X,Y), np(Y,Z).$	$vp \leftarrow v \wedge np$
$N \rightarrow boy$	$n(X,Y) : -boy(X,Y).$ $boy([boy\|X],X).$	$n \leftarrow boy$ boy
$N \rightarrow dog$	$n(X,Y) : -dog(X,Y)$ $dog([dog\|X],X).$	$n \leftarrow dog$ dog
$DT \rightarrow the$	$dt(X,Y) : -the(X,Y).$ $the([the\|X],X).$	$dt \leftarrow the$ the
$DT \rightarrow a$	$dt(X,Y) : -a(X,Y).$ $a([a\|X],X).$	$dt \leftarrow a$ a
$V \rightarrow beats$	$v(X,Y) : -beats(X,Y).$ $beats([beats\|X],X).$	$v \leftarrow beats$ beats

3 ホーン節の書き換えによる文法推論

3.1 確定節文法

これまでの文法推論の研究では，文法の記述に書き換え規則が用いられたが，本稿では一階述語論理を用いる．文法を一階述語論理で表現する方法はPerira と Warren により確定節文法として提案された [6]．確定節文法についての説明は多くの Prolog の教科書で述べられているので詳細はここでは述べない．要するに，書き換え規則は一対一でホーン節と呼ばれる一階述語論理式に置き換えることができる．文法全体はホーン節の集合で記述されることになる．集合のすべての要素のホーン節を AND 結合したものが真，開始記号を表す述語も真のばあいに入力系列は受理される．事実はその記号が入力文に含まれていたら真となり，規則は，←の右辺が真であるとき左辺が真になる．

確定節文法のホーン節を構成する述語は部分文字列を引数とする Prolog 述語を用いて書かれるが，本稿では記述を簡潔にするために，一般的な Prolog による表記とは異なる表記を採用することにする．まず，規則を構成する述語の引数はすべて共通で固定しているので省略する．事実は単独の述語であるが，述語名と引数の文字列を一致させ，その引数も省略する．Prolog 特有の記号である ",", ":-" の代わりに，通常の形式論理と同じ "∧", "←" を使う．この記法では，一階述語論理式は命題論理式と同じ記述形式になり，論理式の意味と操作も命題論理と同様になる．

簡単な正規文法を生成規則と確定節文法で記述した例を表 2示す．"a boy beats the dog" はこの文法が受理する文の一つである．

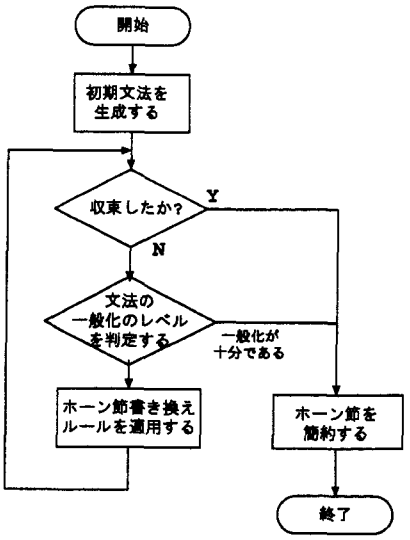

図 1: 文法推論のアルゴリズム

$p1 \leftarrow boy \wedge p3$
$p2 \leftarrow the \wedge boy$
$p4 \leftarrow p2 \wedge beats$
$p6 \leftarrow p2 \wedge p7$
$p7 \leftarrow beats \wedge a$
$p9 \leftarrow p8 \wedge a$
$s \leftarrow p6 \wedge dog$
boy
dog

$p1 \leftarrow p8 \wedge p5$
$p3 \leftarrow beats \wedge p5$
$p4 \leftarrow the \wedge p8$
$p6 \leftarrow p4 \wedge a$
$p8 \leftarrow boy \wedge beats$
$s \leftarrow p2 \wedge p3$
$s \leftarrow the \wedge p1$
$beats$

$p1 \leftarrow p9 \wedge dog$
$p3 \leftarrow p7 \wedge dog$
$p5 \leftarrow a \wedge dog$
$p6 \leftarrow the \wedge p9$
$p9 \leftarrow boy \wedge p7$
$s \leftarrow p4 \wedge p5$
the
a

図 2: 初期文法

3.2 文法推論のアルゴリズム

文法推論のアルゴリズムを図 1に示す．

まず，例文だけを証明する初期文法を生成する．

図 2に "the boy beats a bog" という文から生成した初期文法を示す．

次にホーン節を書き換える操作を繰り返す．節の書き換えは，真理値が似ている論理式の対を推論ルールとして使用する．以下はそのような推論ルールの一つである．P,A,Q,X は述語が代入される変数を表す．

$P \leftarrow A \wedge X \quad Q \leftarrow X \vdash P \rightarrow A \wedge Q \; Q \leftarrow X$

この推論式の両辺の真理値をまとめると表 3のようになる．

$P = 0, A = 1, X = 0, Q = 1$ の解釈のみ真理値が一致しない．この解釈では左辺が真で右辺が偽であるので，右辺から左辺が演繹的に導かれるが，逆は成り立たない．このように大部分の解釈で真理値が一致するが，少数の解釈で真偽が一致しない論理式の対をホーン節の書き換えに使用する．ホーン節の書き換えは，データベースから片方の論理式にマッチするホーン節の組

表 3: 推論式の真理値表

P	A	X	Q	$P \leftarrow A \wedge X \quad Q \leftarrow X$	$P \rightarrow A \wedge Q \; Q \leftarrow X$
0	0	0	0	1	1
0	0	0	1	1	1
0	0	1	0	0	0
0	0	1	1	1	1
0	1	0	0	1	1
0	1	0	1	1	0
0	1	1	0	0	0
0	1	1	1	0	0
1	0	0	0	1	1
1	0	0	1	1	1
1	0	1	0	0	0
1	0	1	1	1	1
1	1	0	0	1	1
1	1	0	1	1	1
1	1	1	0	0	0
1	1	1	1	1	1

み合わせを探索し，見つかればそれを削除し，他方の論理式に相当するホーンをデータベースに追加する方法で行う．この方法で書き換えを行う前のホーン節集合が受理する言語が書き換えを行った後のホーン節集合が受理する言語の部分集合であるのは明らかである．

推論が収束し適用可能なホーン節書き換えルールがなくなるか，一般化が十分であると判断されたときに繰り返しから抜け出てホーン節集合を簡約する．簡約は等価な複数のホーン節があれば，複雑な方を削除する処理である．簡約を行った前後で受理される文の集合は変化しない．簡約を行うことにより，ホーン節は必要最小限のものだけになる．

4　正規文法の推論

ここでは，3種類のホーン節書き換えルールを使って正規文法を学習させた結果について述べる．

実験では，正例として，著者が学習対象の文法が生成する文字列の中からその文法の特徴を良く持ったものいくつか選択しプログラムに入力した．文法推論を実行させた結果，ホーン節集合の内容が正例を生成したもとの文法と一致したら文法推論が成功したと判断した．

正例は多数与えるほど問題は易しくなる．本研究では最低いくつの正例が必要なのかという点については十分に検討していない．以下の実験では，正例の選択基準と個数は著者の主観的な判断によるものであり，現状では客観性と信頼性のある評価はできていない．

以下に基本的な3種類の節書き換えルールを示す．実際のルールはもう少し複雑で，同様の条件が複数重なったとき起動するようになっているが，わかりやすさのために簡略化したものを示す．ルールに現れるA,B,P,Q,X,Yは述語が代入される変数を表す．ルールの上辺はルールが起動される条件を示し，ルールの下辺は変更すべき状態を示す．つまり，ルールが適用されると，上辺にないが下辺にある節は追加され，上辺にあるが下辺にない述語は削除される．

非終端記号の発見

$$\frac{P:-A,X \quad P:-A,Y}{P:-A,Q \quad Q:-X \quad Q:-Y} \tag{1}$$

非終端記号の拡張

$$\frac{P:-A,X \quad Q:-X}{P:-A,Q \quad Q:-X} \tag{2}$$

非終端記号のマージ

$$\frac{P:-A,B \quad Q:-A,B}{P:-A,B \quad Q:-A,B \quad P=Q} \tag{3}$$

以下では幾つかの正規文法を学習例を示す．すべての例について最終的なホーン節データベースの内容が学習対象と一致した．なお紙面の都合で以下には重要な学習過程のみ示す．

4.1　a の後に b の繰り返し

- 学習対象:

  終端記号: a, b

  非終端記号: S, B

  生成規則: $S \rightarrow aB, B \rightarrow bB, B \rightarrow b$

- 正例:

  ab　　abb　　abbb　　abbbb

- 学習過程:

	追加されたホーン節	
1	$S \leftarrow a \wedge B \quad B \leftarrow b \quad B \leftarrow bb$ $B \leftarrow bbb \quad B \leftarrow bbbb$	(1)
2	$B \leftarrow b \wedge B$	(2)

4.2　a の後に b または cd の繰り返し

- 学習対象:

  終端記号: a, b, c, d

  非終端記号: S, A, B

  生成規則: $S \rightarrow aA, A \rightarrow dA, A \rightarrow bB$
  $A \rightarrow d, B \rightarrow c, B \rightarrow cA$

- 正例:

abc	abcbc	abcbcbc	ad
adbc	adbcd	addd	addbcbc
abcd			

- 学習過程:

	追加されたホーン節	
1	$S \leftarrow a \wedge Q1$ $Q1 \leftarrow bc \quad Q1 \leftarrow bcbc \quad Q1 \leftarrow bc$ $Q1 \leftarrow d \quad Q1 \leftarrow dbc \quad \cdots$	(1)
2	$Q1 \leftarrow b \wedge B$ $B \leftarrow c \quad B \leftarrow cbc \quad B \leftarrow cd \quad \cdots$ $Q1 \leftarrow d \wedge Q3$ $Q3 \leftarrow bc \quad Q3 \leftarrow bcd \quad Q3 \leftarrow dd \quad \cdots$	(1)
3	$B \leftarrow c \wedge Q4$ $Q4 \leftarrow bc \quad Q4 \leftarrow bcbc \quad Q4 \leftarrow d$	(1)
4	$Q3 \leftarrow d \wedge Q4$	(2)
5	$Q5 = Q1 = Q3$	(3)
6	$A = Q4 = Q5$	(3)

4.3 パリティ

- 学習対象:

終端記号: $0, 1$

非終端記号: S, Q

生成規則: $S \rightarrow 1, S \rightarrow 0S, S \rightarrow sQ$
$Q \rightarrow 0, Q \rightarrow 1S, S \rightarrow 0$

- 正例:

01	10	001	010
100	111	1011	0001
1000	0111	1101	01110
01011	11111		

- 学習過程:

	追加されたホーン節	
1	$S \leftarrow 0 \wedge Q1$ $Q1 \leftarrow 1 \quad Q1 \leftarrow 01 \quad Q1 \leftarrow 10 \cdots$ $S \leftarrow 1 \wedge R$ $Q2 \leftarrow 0 \quad Q2 \leftarrow 00 \quad Q2 \leftarrow 11 \cdots$	(1)
2	$Q1 \leftarrow 1 \wedge Q3$ $Q3 \leftarrow 0 \quad Q3 \leftarrow 011 \quad Q3 \leftarrow 11 \cdots$ $Q2 \leftarrow 0 \wedge Q4$ $Q4 \leftarrow 0 \quad Q4 \leftarrow 00 \quad Q4 \leftarrow 11 \cdots$ $Q2 \leftarrow 1 \wedge Q5$ $Q5 \leftarrow 1 \quad Q5 \leftarrow 01 \quad Q5 \leftarrow 111 \cdots$	(1)
3	$Q1 \leftarrow 0 \wedge Q5$ $Q1 \leftarrow 1 \wedge Q4$ $s \leftarrow 1 \wedge Q3$ $Q1 \leftarrow 1 \wedge Q2$	(2)
3	$s = Q1 = Q5$ $Q = Q2 = Q3 = Q4$	(3)

4.4 ＋を挟んだ0または1の繰り返し

- 学習対象:

終端記号: a

非終端記号: S, A

生成規則: $S \leftarrow 0, S \leftarrow 1, S \leftarrow 0A$
$S \rightarrow 1A, A \rightarrow +S$

- 正例:

0	1	1+0	0+1
1+0+0	0+1+1	1+1+1	0+0+1
1+1+1+1	0+0+0+0	1+0+0+1	1+0+1+0
1+0+1+1	1+1+0+0	0+1+0+1	

- 学習過程:

	追加されたホーン節	
1	$S \leftarrow 0 \wedge Q1$ $Q1 \leftarrow +1 \quad Q1 \leftarrow +0+1 \quad \cdots$ $S \leftarrow 1 \wedge Q2$ $Q2 \leftarrow +0 \quad Q2 \leftarrow +0+0 \quad \cdots$	(1)
2	$Q1 \leftarrow + \wedge Q3$ $Q3 \leftarrow 1 \quad Q3 \leftarrow 0+1 \quad \cdots$ $Q2 \leftarrow + \wedge Q4$ $Q4 \leftarrow 0 \quad Q4 \leftarrow 0+0 \quad \cdots$	(1)
3	$Q4 \leftarrow 0 \wedge Q5$ $Q5 \leftarrow +0 \quad Q5 \leftarrow +0+1 \quad \cdots$ $Q4 \leftarrow 1 \wedge Q6$ $Q6 \leftarrow +1 \quad Q6 \leftarrow +1+1 \quad \cdots$	(1)
4	$Q5 \leftarrow + \wedge Q7$ $Q7 \leftarrow 0 \quad Q7 \leftarrow 0+1 \quad \cdots$ $Q6 \leftarrow + \wedge Q8$ $Q8 \leftarrow 1 \quad Q8 \leftarrow 1+1 \quad \cdots$	(1)
5	$Q1 \leftarrow + \wedge Q8$ $Q2 \leftarrow + \wedge Q8$ $Q3 \leftarrow 0 \wedge Q6$ $s \leftarrow 0 \wedge Q6$ $s \leftarrow 1 \wedge Q6$	(2)
6	$R = Q1 = Q2$ $U = Q3 = Q4$	(3)
7	$V = R = Q10$ $W = U = Q7$ $X = U = Q8$	(3)
8	$A = V = Q6$ $S = X = W$	(3)

4.5 偶数個のa

- 学習対象:

終端記号: a

非終端記号: S, A

生成規則: $S \leftarrow aa, S \leftarrow aA, A \leftarrow aS$

- 正例:

aa	aaaa	aaaaaa	
aaaaaaaaaa			

- 学習過程:

	追加されたホーン節	
1	$S \leftarrow a \wedge Q1$ $Q1 \leftarrow a \quad Q1 \leftarrow aaa \quad Q1 \leftarrow aaaaa$ $Q1 \leftarrow aaaaaaaaa$	(1)
2	$Q1 \leftarrow a \wedge Q2$ $Q2 \leftarrow aa \quad Q2 \leftarrow aaaa$ $Q2 \leftarrow aaaaaaaa$	(1)
3	$Q2 \leftarrow a \wedge A$ $A \leftarrow a \quad A \leftarrow aaa \quad A \leftarrow aaaaaaa$	(1)
4	$S \leftarrow a \wedge A$	(2)
5	$S = Q2$	(3)

5　文脈自由文法の推論

文脈自由文法は，正規文法よりも能力が高く応用範囲が広いが，より文法推論は難しいと考えられている．

{a,(,),+}の4種類の文字を使って作られる7種類の文

"a","a+a","a+a+a","a+(a+a)",
"(a+a)+(a+a)","(a+a+a)","a+a+a+a"

を正例として文法推論させる実験を行った．使用したアルゴリズムおよび推論ルールは，正規文法の推論と同じものを使用した．

ただし，現段階では簡約の処理が実装できていないため，推論の成功を学習した結果のホーン節集合を直接評価することができない．そこで，推論結果のホーン節集合から生成される文を使って評価を行った．

実験の結果，表4に示す文を生成する文法を得た．一度目の推論サイクル後は，比較的例に似た文が生成されるが，二度目の推論サイクル後は，例と類似性が低い文も生成されるようになり，生成される文の数が多くなっている．なお，偶数の長さの文は生成されなかった．この表からは文法推論が成功していると判断できる．

現段階では，研究が不十分で不明な部分が多いが，文脈自由文法の学習についても本稿の方法が有効であるようなので，今後重点的に研究を行っていきたい．

6　付加的情報の利用

ここまでに示した例は非常に小さな言語を対象としたが，現実の問題では，もっと大きな言語を扱う必要がある．そのためには，前節で述べた基本的な仕組みに加えて，ここで述べるような付加的情報の利用のための仕組みが必要となる．逆に言うと，このような工夫により，効率の悪いアルゴリズムしか発見できていない状況でも，文法推論の実用化が期待できる．

表4: 文脈自由文法の学習結果

長さ	例文	最初の推論サイクル後	2度目の推論サイクル後
1	a	a	a
3	a+a	a+a	a+a (a)
5	a+a+a	a+a+a	(a)+a a+a+a a+(a) (a+a) ((a))
7	(a+a+a) a+(a+a) a+a+a+a	(a+a)+a a+a+a+a a+(a+a) (a+a+a)	((a))+a (a+a)+a (a)+a+a (a)+(a) a+(a)+a a+a+a+a a+a+(a) a+(a+a) a+((a)) (a+a+a) ((a)+a) (a+(a)) ((a+a)) (((a)))
9		a+(a+a)+a (a+a)+a+a a+a+(a+a) a+a+a+a+a	(((a)))+a ((a+a))+a (a+(a))+a ((a)+a)+a (a+a+a)+a (((a+a))) ((((a))))
11	(a+a)+(a+a)	a+a+(a+a)+a a+(a+a)+a+a (a+a)+a+a+a a+a+a+a+a+a+a a+a+a+(a+a) (a+a)+(a+a)	((((a))))+a (((a+a)))+a ((a+(a)))+a (((a)+a))+a ((a+a+a))+a (a+((a)))+a ((((a+a)))) (((((a)))))

6.1　既存の文法規則の利用

文字(終端記号)の種類が多いと文法推論はうまく働かない．たとえば，"06-123-4567","0720-20-6401"といった数字だけの文字列が-でつながっているような言語を学習させることができない．文字の種類が多いばあいは，文字のカテゴリについての情報を与えることが有効である．上の例では，数字というカテゴリの情報を与えると，実質的な文字の種類は数字と-の2種類になり推論ルールが適用されやすくなる．

具体的には，初期文法に $number \leftarrow 0$, $number \leftarrow 1$ といったホーン節を加えるとよい．

6.2　構造の情報の利用

文脈自由言語が複雑で例文が長くなると，推論に要する時間が非常に大きくなる．その解決方法として，正例に構造についての情報を付加すること(たとえば括弧で範囲を示す)が有効である．構造を与えることにより初期文法が非常に小さくなり，計算時間が飛躍的に小さくなることが確認できている．

7 まとめ

本稿では，ホーン節の書き換えに基づいた文法推論システムを提案し，そのシステムが簡単な正規言語の正例だけからの学習ができることを示した．それにより，文法推論にホーン節の書き換えを利用する本研究のアプローチの正当性が確認できた．

このシステムの特長は二つある．一つは，従来の実用指向の文法推論の手法が研究者の発見的手法による場当たり的なものであったのに対し，ホーン節の書き換えルールは形式論理に基づき導かれるため，有効な推論を確実に実行することができると考えられる．もう一つの特長は，推論における一般化の強さを制御できる点である．ホーン節の書き換え前後でどの解釈において論理式の真理値が異なるかという点で，一般化の強さを調べることができ，似た働きをするが一般化の強さが異なるルールを複数開発することも容易である．また，ルールとマッチするホーン節の個数からルールを適用するか否かを判断させることができる．このとき確率論に基づいた手法が適用できる可能性もある．

ただし，本稿で述べた3個のホーン節の書き換えルールは著者が恣意的に選んだもので，有効な論理式を組織的に調べて得たわけではない．また，一般化の強さを制御することについての研究もまだ行っていない．今後は文脈自由文法を学習するシステムの開発とあわせて，論理式を組織的な調査と一般化の強さを制御について研究を行っていきたい．

参考文献

[1] D. Angluin. Inductive inference of formal languages from positive data. *Inform. Control*, Vol. 45, No. 2, pp. 117–135, May 1980.

[2] D. Angluin. Learning regular sets from queries and counterexamples. *Inform. Comput.*, Vol. 75, No. 2, pp. 87–106, November 1987.

[3] K. Fu and L. Booth. Grammatical inference: Introduction and survey - part 1. In *IEEE Transactions on Systems, Man and Cybernetics SMC-5*, pp. 95–111, 1975.

[4] E. M. Gold. Language identification in the limit. *Inform. Control*, Vol. 10, pp. 447–474, 1967.

[5] 花川賢治, 武田英明, 西田豊明. 柔軟な問い合わせのための弱構造化表現. システム制御情報学会論文誌, Vol. 10, No. 7, pp. 341–350, 1997.

[6] 松本祐治. 論理型言語による自然言語へのアプローチ. 淵一博（編）, 自然言語の基礎理論, pp. 51–86. 共立出版, 1986.

[7] 西田豊明. 自然言語処理入門. オーム社, 1988.

[8] T. Shinohara. Inductive inference of monotonic formal systems from positive data. *New Generation Computing*, Vol. 8, pp. 371–384, 1991.

[9] 高田裕志. 計算論的学習理論の応用システム. 情報処理, Vol. 32, No. 3, pp. 264–271, 1991.

[10] 横森貴, 篠原武. 形式言語の学習. 情報処理, Vol. 32, No. 3, pp. 226–235, 1991.