拡張チョムスキー標準形を用いた文脈自由文法の学習

河島 輝† 中村 克彦‡

††東京電機大学 理工学部

1 まえがき

われわれは文脈自由文法 (CFG: Context Free Grammar) の文法推論のための Synapse システムの開発を進めてきた [1]. Synapse はブリッジ法による規則生成と，規則集合の探索によって CFG の漸次学習を実現している．ブリッジ法は構文木の欠けた部分を補うような最小の規則を追加していく手法である．さらに Synapse は確定節文法 (DCG:Definite Clause Grammar) の合成を行うように拡張され，コンパイラの学習などに応用されている [2]．DCG は CFG を拡張した文法で CFG より複雑な言語を扱うことができる．

現在の Synapse による CFG の学習では変形チョムスキー標準形と呼ばれる形式の規則が合成される．本報告ではチョムスキー標準形を拡張チョムスキー標準形と呼ばれる形式に拡張し，この形式の規則の合成法と CFG の学習について述べる．拡張チョムスキー標準形はひとつの規則に多くの終端記号が含まれることを許しているため，少ない規則で文法を表現することができ，効率の高い規則合成を行うことができる．

2 拡張チョムスキー標準形

文脈自由文法は，$G = (N, T, P, S)$ で表される．ここで N，T，P はそれぞれ非終端記号，終端記号，(生成) 規則の有限集合であり，$S \in N$ は開始記号である．規則は一般に次のような形式をもつ．

$$A \to \beta \quad (A \in N, \beta \in (T \cup N)^+)$$

ある非端記号 A から記号列 $w \in (N \cup T)^+$ が導出できるとき，$A \overset{*}{\Rightarrow} w$ と表す．すべての文脈自由言語 (CFL) は次のチョムスキー標準形 (CNF: Chomsky Normal Form) の規則からなる CFG によって表すことができる．

$$A \to BC \quad (A, B, C \in N)$$
$$A \to \gamma \quad (\gamma \in T)$$

拡張 CNF の規則はその右側に含まれる非終端記号を 2 個以下に制限したものである．この規則は一般に次のいずれかの形式をもつ．ただし $A, B, C \in N, \alpha_1, \alpha_2, \alpha_3 \in T^*, \alpha_0 \in T^+$ である．

$$A \to \alpha_1 B \alpha_2 C \alpha_3, \qquad A \to \alpha_1 B \alpha_2, \qquad A \to \alpha_0$$

現在の Synapse においては次のような形式の変形 CNF(過去の論文 [1, 2] においては拡張 CNF と呼んでいた) の規則を合成しており，CNF の規則を含む．

$$A \to \delta\lambda \quad (A \in N, \delta, \lambda \in (T \cup N))$$
$$A \to \delta$$

3 構文解析

拡張 CNF の規則からなる CFG の構文解析は CYK(Cocke-Younger-Kasami) アルゴリズムを基にしたボトムアップ解析を用いることができる．拡張 CNF に対しても一般の CNF の場合と同様に CYK アルゴリズムによって記号列の長さ n に対し $O(n^3)$ の計算量で構文解析を行うことができる．

4 ブリッジ法を用いた規則合成

ブリッジ法では，まず正例の記号列 $a_1 a_2 \cdots a_n$ に対してボトムアップ構文解析を行う．構文解析に失敗したとき，得られた不完全な構文木に対してブリッジ法の規則が適用される．以下，$w_{mn} = a_{m+1} \cdots a_n$, ただし $m = n$ のとき $w_{mn} = \epsilon$ であり，$A, Q, R \in N$，P を規則集合とする．部分記号列 w_{mn} の非終端記号 A からの不完全な導出木に対して以下の規則を非決定的に適用される．

1. $Q \overset{*}{\Rightarrow} w_{ij}$ のとき，規則 $A \to w_{mi}\, Q\, w_{jn}$ を生成し P に加える．
2. $A \to w_{mi}\, Q\, w_{jn} \in P$ のとき，目標を Q に置き換えて w_{ij} に対して再帰的にブリッジ法の規則を適用する．
3. $Q \overset{*}{\Rightarrow} w_{ij}$, $R \overset{*}{\Rightarrow} w_{kl}$ のとき, 規則 $A \to w_{mi}\, Q\, w_{jk}\, R\, w_{ln}$ を生成し P に加える．
4. $A \to w_{mi}\, Q\, w_{jk}\, R\, w_{ln} \in P$ であり，$Q \overset{*}{\Rightarrow} w_{ij}$ のとき, 目標を R に置き換えて w_{kl} に対して再帰的にブリッジ法の規則を適用する．
5. $A \to w_{mi}\, Q\, w_{jk}\, R\, w_{ln} \in P$ であり，$R \overset{*}{\Rightarrow} w_{kl}$ のと

Learning Context Free Grammar of Extended Chomsky Normal form

Kawashima AKIRA†, Katsuhiko NAKAMURA‡

††Tokyo Denki University, School of Science and Engineering 350-0394 Ishizaka Hatoyama-cho Hiki-gun Saitama-ken Japan

†highjumn@naklab.k.dendai.ac.jp ‡nakamura@k.dendai.ac.jp

表 1: 拡張 CNF と変形 CNF の比較

言語	規則	合成された文法規則	R	GR	$T[sec]$
括弧言語	拡張 CNF	$s \to ab,\ s \to asb,\ s \to ss$	3	61	0.64
	変形 CNF	$p \to sb,\ s \to ab,\ s \to ap,\ s \to ss$	4	46	0.13
回文言語	拡張 CNF	$s \to a,\ s \to b,\ s \to aa,\ s \to asa,\ s \to bb,\ s \to bsb$	6	267	2.36
	変形 CNF	$p \to sa,\ q \to sb,\ s \to a,\ s \to aa$ $s \to ap,\ s \to b,\ s \to bb,\ s \to bq$	8	277	0.16
$\{w \in \{a,b\} \mid$ $\sharp_a(w) = \sharp_b(w)\}$	拡張 CNF	$s \to ab,\ s \to ba,\ s \to asb,\ s \to ss,\ s \to bsa$	5	1159	17.6
	変形 CNF	$p \to sb,\ q \to sa,\ s \to ab,\ s \to ap,$ $s \to ba,\ s \to bq,\ s \to ss$	7	578	3.13
$\{w \in \{a,b\} \mid$ $\sharp_a(w) < \sharp_b(w)\}$	拡張 CNF	$s \to b,\ s \to asb,\ s \to bs, s \to sas,\ s \to ssa$	5	5675	66.9
	変形 CNF	$p \to as,\ p \to sa,\ s \to b,\ s \to bs,\ s \to pb,\ s \to sp$	6	1390	9.19

き, 目標を Q に置き換えて w_{ij} に対して再帰的にブリッジ法の規則を適用する.

6. $Q \overset{*}{\Rightarrow} w_{ij}$ のとき,k,l を $j \leq k < l \leq n$ の範囲で非決定的に決める. $A \to w_{mi}\, Q\, w_{jk}\, R\, w_{ln}$ を生成し P に加え, 目標を R に置き換えて w_{kl} に対して再帰的にブリッジ法の規則を適用する.
7. $R \overset{*}{\Rightarrow} w_{kl}$ のとき,i,j を $m \leq i < j \leq k$ の範囲で非決定的に決める. $A \to w_{mi}\, Q\, w_{jk}\, R\, w_{ln}$ を生成し P に加え, 目標を Q に置き換えて w_{ij} に対して再帰的にブリッジ法の規則を適用する.
8. $A \to w_{mi}\, Q\, w_{jk}\, R\, w_{ln} \in P$ のとき,i,j と k,l をそれぞれ $m \leq i < j \leq k < l \leq n$ の範囲で非決定的に決める. 目標を Q と R に置き換えて, Q は w_{ij} に対して再帰的にブリッジ法の規則を適用する. R は w_{kl} に対して再帰的にブリッジ法の規則を適用する.
9. $A \to w_{mn}$ を生成する.

5 規則集合の探索

反復深化を用いて次の手順により規則集合の探索を行なう.

1. 初期値として, 規則集合の大きさの制限値 $K = 0$, 規則集合 $P = \phi$ とする
2. すべての正例の文字列 w に対して次の操作を行なう. 失敗したら K に 1 を加えてこれを繰り返す.
   (a) w に対して構文解析を行い, P が w を導出したら終了する.
   (b) w を導出するような K 個以内の規則を P に加える.
   (c) すべての負例に対して構文解析を行なう. 導出したら失敗する.

6 文法規則の合成

拡張 CNF の規則合成と Syanpse による変形 CNF の規則合成の比較の結果を表 1 に示す. ただし R は生成された規則の数, GR は目的の文法を得るまでにシステムが生成した規則数であり計算量の指標として用いている. T は規則合成にかかった時間 [sec] である. また $\sharp_a(w)$ は文字列 w に含まれる a の数を示している.

Synapse より少ない規則数で規則を合成することができている. GR 及び規則合成にかかった時間は Synapse に対し 1 倍から 4 倍の GR を要している.

7 むすび

CFG のための規則の形式拡張 CNF を提案し, この規則の合成法と CFG の学習について述べた. この形式の CFG は CNF の文法と同様に記号列の長さ n に対し $O(n^3)$ の計算量で構文解析ができる. 拡張 CNF による CFG の漸次学習の結果, より少ない規則数の文法が合成されることが示された.

今後の課題として拡張 CNF による規則合成の高速化と DCG への拡張があげられる.

参考文献

[1] K. Nakamura. Incremental learning of context free grammars by bridging rule generation and search for semi-optimum rule sets. *Lecture Notes in Computer Science*, 4201:72, 2006.

[2] K. Imada and K. Nakamura. Towards Machine Learning of Grammars and Compilers of Programming Languages. In *Proceedings of the European conference on Machine Learning and Knowledge Discovery in Databases-Part II*, pages 98–112. Springer, 2008.