

複数の接続制約の LR 表への組み込みと それによる解析の統合化

綾部寿樹, 徳永健伸, 田中穂積
東京工業大学大学院情報理工学研究所
email:{ayabe,take,tanaka}@cs.titech.ac.jp

1 はじめに

自然言語の統語解析のアルゴリズムである GLR 法は, 先読み語の持つ品詞情報を利用しつつ解析を進めるもので, 経験的にもっとも効率が良いとされている [1][2].

GLR 法は, 最近では統語解析のみならず, 形態素解析及び音声認識にも応用されるようになってきている. 形態素間の接続表の制約を LR 表に組み込むことで, 形態素解析と統語解析の統合を行なう方法が開発されている [3][4]. 異音間の接続表の制約を LR 表に組み込むことで, 音声認識に有効であることも示されている [5][6].

ここで, 形態素間の接続表と異音間の接続表といった, 異なるレベルの複数の接続表の持つ制約とともに LR 表に組み込むことが可能であれば, GLR 法という1つの枠組で, 複数の接続表の持つ接続制約および CFG による文法的制約を同時に用いた解析が可能になる. この場合, 音声認識と形態素解析, 及び統語解析の3つの解析を統合して扱うことができる.

解析の統合を行なうことは工学的にも認知科学的にも重要である. なぜならば, 人間は多種多様な制約を同時に用いて, 解析途中で発生する様々な曖昧性を早期に解消して効率良く言語を理解していると考えられるからである.

本稿では, 複数の接続表の制約を LR 表に組み込む1つの方法を説明する. 2章では, 接続表で表し得る局所制約をすべて CFG で記述した場合の問題点を整理し, 1つの接続表の制約を LR 表へ組み込む方法の利点を示す [7]. 3章では, 複数の接続表の制約を LR 表に組み込む方法を説明する.

2 接続表の制約の LR 表への組み込み

自然言語の統語解析に用いる文法的枠組として, CFG がある. 一方, 形態素解析では形態素間の接続可能性という局所的な制約がよく用いられる. この形態素間の接続可能性という制約を CFG で記述した場合, CFG 規則の数が不必要に増え, CFG 規則の体系が複雑化するという問題がある. 解決すべき問題は以下の3つである [7].

1. CFG 記述者は, 接続可能性を考慮した新たな非終端記号の導入を行なうことなしに, CFG の記述が可能であること
2. 接続制約の記述者は, CFG 規則とは無関係に制約を記述可能であること
3. 局所的制約検査のタイミングは, なるべく早く行なうこと

これらを同時に解決する方法として, CFG 規則とは独立に終端記号間の局所的制約を接続表という形で記述しておき, この制約を CFG 規則から得られた LR 表に組み込む方法が提案されている [7]. この方法により, 形態素解析と統語解析, もしくは音声認識と統語解析を統合することが可能であることが示されている [3][6]. また音声認識と統語解析を統合した場合においては各解析を別々に行なった場合に比べ音素レベルの音声認識の予測精度と認識精度が向上することが確認されている [6].

さらに, この方法により使用記憶空間を大幅に削減し, 最終的な LR 表を得る時間を一桁以上短縮できることが報告されている [5].

3 複数の接続表の制約の LR 表への組み込み

田中の方法を応用して隣接形態素間の接続制約と隣接異音間の接続制約という2つの接続制約（接続表）を LR 表に組み込むことを考える。辞書項目を異音と音素の列として記述すると、異音が終端記号になるので、異音間の接続表の制約はこれまでの方法をそのまま適用して容易に LR 表に組み込むことが可能である。しかし、LR(1) アイテムの先読み記号が異音であるため、異音間の接続制約しか組み込むことができない。したがって、形態素間の接続表についてはその制約を組み込むことができない。

本章では、2つの接続表の制約を LR 表へ組み込む方法を説明する。その方法としては、LR 表の生成を2段階に分割するという方法をとる。

なお本章以降ではギリシャ文字の α, β, \dots などは次のいずれかの記号列を表す：終端記号の列、非終端記号の列、終端記号と非終端記号の混在した列。

3.1 文脈自由文法の層

2つの接続表の制約を LR 表へ組み込むためには CFG が層を持たなければならない。ここでは CFG の層という概念を述べる。

層を持つ CFG とは、CFG 規則の集合を、開始記号からあるレベル L2 の記号列（例えば形態素列）を導出する規則と、そのレベル L2 の記号列から終端記号の記号列（例えば異音列）を導出する規則とに排他的に分割できる CFG のことである。

1	$S \rightarrow XYZ$	6	$a \rightarrow a1$
2	$X \rightarrow a$	7	$a \rightarrow a2$
3	$Y \rightarrow c$	8	$c \rightarrow c1$
4	$Y \rightarrow e$	9	$c \rightarrow c2$
5	$Z \rightarrow e$	10	$e \rightarrow e1$
CFG2 P2		11	$e \rightarrow e2$

P1 CFG1

図1：層を持つ CFG

例えば図1のCFG1は、CFG2という層を持っている。規則集合P1が、規則集合P2とP1-P2の部分集合に重複なく分かれていて、P2では開始記号から記号a,c,eの記号列を、またP1-P2では記号a,c,eの記号列からa1,c1などの記号列を導出する。

以降の説明において、CFG1がCFG2という層を持つものと仮定し、CFG1の終端記号の集合をVt1、CFG2の終端記号の集合をVt2とする。また、CFG1規則の集合をP1、CFG2規則の集合をP2と呼ぶこととする。

3.2 LR(1) アイテムの拡張

CFG1からGLR法で生成するアイテムは、その先読み記号は終端記号すなわちVt1に属す記号であった。したがって、例えばアイテム $[V \rightarrow \cdot \{X; v_i]$: $v_i \in Vt1$ の生成時に、Xと v_i の接続を調べる際、Vt1のレベルの接続表（以降接続表1とする）を用いて接続可能かどうかを調べることはできるが、Vt2のレベルの接続表（以降接続表2とする）を用いて接続可能かどうかをチェックすることはできない。

この問題を解決するために、先読み記号としてVt1に属す記号をとるアイテムと、先読み記号としてVt2に属す記号をとるアイテムを別々に定義する。ここでは便宜上、前者を型1のアイテム、後者を型2のアイテムと呼ぶ。

[型1のアイテムの定義]

型1のアイテム $[X \rightarrow \alpha \cdot \beta; v_i]$

ただし $v_i \in Vt1, X \rightarrow \alpha\beta \in P1-P2$

[型2のアイテムの定義]

型2のアイテム $[X \rightarrow \alpha \cdot \beta; v]$

ただし $v \in Vt2, X \rightarrow \alpha\beta \in P2$

3.3 2つの接続表の制約の LR 表への組み込み

CFG1がCFG2という層を持つ場合、2つの接続表の制約を順番に LR 表に組み込むことが出来る。本節ではその方法を説明するが、その前に必要な関数を定義しておく。

3.3.1 関数定義

Last(X) 記号Xを根とする解析木の右分枝の先端に現れるVt1に属す記号の集合。

First(X) 記号Xを根とする解析木の左分枝の先端に現れるVt1に属す記号の集合。

Last2(X) 記号 X を根とする解析木の右分枝の先端に現れる $Vt2$ に属す記号の集合。

First2(X) 記号 X を根とする解析木の左分枝の先端に現れる $Vt2$ に属す記号の集合。

特に記号 X が $Vt2$ に属す記号の場合には、
 $First2(X)=Last2(X)=\{X\}$ である。

また記号 X が $Vt1$ に属す記号の場合には、
 $First(X)=Last(X)=\{X\}$ である。

3.3.2 LR 表生成アルゴリズム

(1)CFG2 における接続表 2 の制約を組み込んだ closure の生成 (接続表 1 の制約についても考慮)

CFG2 における closure を生成し、GOTO グラフ G2 を作成する。アイテムの生成には以下のアイテム生成手続き 1 を用いる。

[アイテム生成手続き 1]

CFG2 と接続表 1 と接続表 2 を用いて、次の 1,2 にしたがって型 2 のアイテムを生成する。

1. アイテム $[V \rightarrow \cdot \xi X; v] : v \in Vt2$ の生成は以下の条件を満たす場合にのみ行なう。

- $Last2(X)$ に属する要素と v の対のうち、接続表 2 においてこの順に接続可能な対が存在し、かつその対を u, v としたとき、 $Last(u)$ に属する要素と $First(v)$ に属する要素の対のうち、接続表 1 においてこの順に接続可能な対が存在する。

2. 記号 X をシフトして到達した新しい状態におけるアイテム $[V \rightarrow \cdot W\xi; v] : v \in Vt2$ の生成は、条件 1 を満たしかつ以下の条件を満たす場合にのみ行なう。

- $Last2(X)$ に属する要素と $First2(W)$ に属する要素の対のうち、接続表 2 においてこの順に接続可能な対が存在し、かつその対を x, w とした時、 $Last(x)$ に属する要素と $First(w)$ に属する要素の対のうち、接続表 1 においてこの順に接続可能な対が存在する。

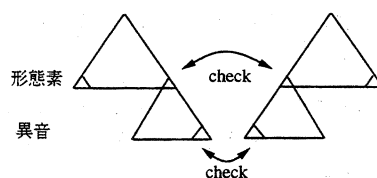


図 2：接続のチェック

このようにして GOTO グラフ G2 を作成する。

(2)P1-P2 の規則を用いた拡張

GOTO グラフ G2 の各状態の、ドットのすぐ右に $Vt2$ に属する記号を持つアイテムを核として、P1-P2 に属する規則の closure を生成し、GOTO グラフ G1 を作成する。アイテムの生成には以下のアイテム生成手続き 2 を用いる。

[アイテム生成手続き 2]

CFG と接続表 1 を用いて次の 1,2 を満たす場合のみ型 1 のアイテムを生成する。

1. アイテム $[V \rightarrow \cdot \xi X; v_i] : v_i \in Vt1$ の生成は次の条件を満たす場合にのみ行なう。

- $Last(X)$ に属する要素と v_i の対のうち、この順に接続可能な対が存在する。

2. 記号 X をシフトして到達した状態におけるアイテム $[V \rightarrow \cdot W\xi; v_i] : v_i \in Vt1$ の生成は、条件 1 を満たしかつ次の条件を満たす場合にのみ行なう。

- $Last(X)$ に属する要素と $First(W)$ に属する要素の対のうち、この順に接続可能な対が存在する。

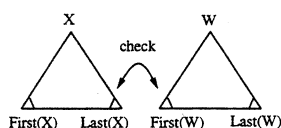


図 3：接続のチェック

ここで得られた GOTO グラフ G1 には、接続表 1 と接続表 2 の 2 つの接続制約が組み込まれている。

(3)LR 表の生成

GOTO グラフ G1 から LR 表を作成する。ただし、次の 2 つの点に注意する。

1. Vt2に属する記号のシフトはそのまま GOTO にしてよい。
2. レデュース用のアイテムの先読み記号が Vt2 に属する記号 w の場合、先読み記号は以下のように求める。

そのアイテムの規則の右辺最右の記号を X とすると、 $\text{First}(w)$ に属する記号のうち、接続表 1 において $\text{Last}(X)$ に属する記号のいずれかと接続可能な記号すべてを先読み記号とする。

(4) 制約伝播

(3) で得られた LR 表に制約伝播を施す。制約伝播とは、接続表の制約を組み込みながら LR 表を作成したことにより生成されなかったアクションがあるため、それにともなって不要となるようなアクションを削除する手法である。

これは 1 つの接続表を組み込む場合と同一であるので [7] を参照されたい。

以上のようにして 2 つの接続制約を組み込んだ LR 表の作成が可能である。この手順を繰り返すことにより、3 個以上の接続表の制約を LR 表に組み込むことが可能になる。

4 おわりに

本稿では 2 つの接続表の制約を LR 表へ組み込む方法を示した。この方法によって、形態素間の接続表、異音間の接続表を両方共組み込み、音声認識、形態素解析、統語解析をすべて GLR 法の枠組で統合して行なうことができる。本方式の利点は、接続制約が増えると LR 表上のアクションの数、状態数共に減少することにある。これは、従来の方法では制約の数が増えると一般にアルゴリズムが複雑化するのといふ対照をなす。3 つの解析すべての制約を満たすアクションだけを残した LR 表を作成することが可能であることから、音声認識に例をとれば、より精密な音素の予測が可能となり、音声認識の精度向上が期待できる。

われわれは、すでにこの方法による LR 表生成プログラムをワークステーション上に実装することに

成功している（使用言語は C）。大規模な CFG に対して実際に接続表を 2 つ組み込んだ LR 表を作成し、本稿で述べたことを実証することが今後の課題である¹。

参考文献

- [1] A.V. Aho, S. Ravi, and J.D. Ullman. *Compilers, Principle, Techniques, and Tools*. Addison Wesley, 1986.
- [2] M Tomita. *Generalized LR Parsing*. Kluwer Academic Publishers, 1991.
- [3] 植木正裕, 徳永健伸, 田中穂積. EDR 辞書を用いて形態素解析と統語解析を行なうシステム. EDR 電子化辞書利用シンポジウム論文集, pp. 33-39, 1995.
- [4] H. Tanaka, T. Tokunaga, and M. Aizawa. Integration of morphological and syntactic analysis based on lr parsing algorithm. *Journal of Natural Language Processing*, Vol. 2, No. 2, pp. 59-74, 1995.
- [5] Suresh K.G. Li H. and Tanaka H. Incorporation of connection constraints into generation process of allophone-base lr table. 情報処理学会第 50 回全国大会講演論文集, 1995.
- [6] 田中穂積, 李輝, 徳永健伸. Incorporation of phoneme-context-dependence in lr table through constraint propagation method. 人工知能学会第 8 回言語・音声理解と対話処理研究会, pp. 15-22, 6 1994.
- [7] 田中穂積, 李輝, 徳永健伸. 自然言語解析の新しい方法—LR 表工学の提案 (1). 人工知能学会, 1995.

¹ 文法規則数 2000 程度の CFG についての実験は行ない、動作を確認したが、その際には、異音の列を導出する CFG に対して、非終端記号のレベルにあたる形態素の接続表のみを組み込んだ実験しか行なっていない。このこと自体も新しい試みではあるが、2 つの接続表を同時に組み込む利点を実証したことにはなっていない。