GLR 法に基づく統語解析過程の制御法
— LR 表工学の提案 —

田中 穂積
東京工業大学大学院 情報理工学研究科

1 はじめに

GLR 法は，Knuth による LR 法 [Knuth 65] を，一般の CFG が扱えるように拡張したものである．LR 法と同様に GLR 法でも，与えられた CFG から統語解析で用いる表 (LR 表) をあらかじめ作成しておく．LR 表により，先読み語（先読み記号）の前終端記号と現在の解析状態を用いて，次に行う統語解析動作を決めることができる．GLR 法は，LR 表を用いて無駄のない統語解析を行うことができるので，経験的にもっとも効率の良い統語解析アルゴリズムであるとされている [Tomita 86].

本稿では，CFG の形式では記述が煩雑になる細かな制約を LR 表に組み込み，より精密な統語解析を行う技法について述べる．この技法を我々は「LR 表工学」とよぶ．以下では「LR 表工学」により，従来の統語解析では不可能であったことが可能になったり，可能であるにしても煩雑で困難であったことが容易に実現できることを具体的に示す．

2 コンフリクトの解消法

2.1 統語的曖昧性の恣意的解消

英語では “and” で結合された文，名詞の並びからなる複合名詞句，名詞が “and” や “of” で結合された名詞句を解析するための CFG 規則は極めて多数の統語解析結果（統語的曖昧性）を生み出す．

S → S and S, N → N N,
NP → NP and NP, NP → NP of NP

Church らは，この種の曖昧性を every way ambiguous construction とよんでいる．統語的曖昧性の数は n の増加とともに組み合わせ数的に増加するにもかかわらず，妥当な統語解析結果は多くの場合ただ一個である．妥当な統語解析結果は正しい意味構造を反映しているものであるが，正しい意味構造は統語解析の段階ではまだ知られていない．このような場合には統語解析の段階で，文中での名詞句の位置などを知る程度の単純な統語解析結果だけを恣意的に得るにとどめておきたい[1]．次節では GLR 法で用いる LR 表中のコンフリクトの解消という観点から，この問題を考える [Aho 75].

[1]このことは問題を先送りにしただけだと考える読者もいるかもしれない．しかし意味解析により意味構造を構成する段階でさまざまな制約を働かせて中間での計算量を絞り込むことができる．

2.2 コンフリクトの解消例

やや複雑な例を取り上げて，解消すべきコンフリクトおよび削除すべき動作の同定が必ずしも容易でないことを説明する．そのために次の CFG を考える．

(1)	S → NP VP	(6)	NP → NP PP
(2)	S → S PP	(7)	NP → NP and NP
(3)	S → S and S	(8)	VP → v NP
(4)	NP → n	(9)	VP → v S
(5)	NP → det n	(10)	PP → p NP

図 1: Grammar I

図 1 に示す CFG を用いて主語 NP の直後に複数の PP が並ぶ文，たとえば “Mary:n with:p a:det book:n for:p cooking:n loves:v John:n.” を解析すると図 2 に示す 2 つの解析結果 (骨格のみ) を得る．

```
1) [S [NP [NP n]
          [PP p [NP [NP det n]
                    [PP p [NP n]]]]]
      [VP v [NP n]]]

2) [S [NP [NP [NP n]
              [PP p [NP det n]]]
          [PP p [NP n]]]
      [VP v [NP n]]]
```

図 2: PP 付加の 2 つの統語解析木

GLR 法による統語解析で恣意的に右下がりの 1) の構造だけを得たいとしよう．図 1 の CFG から図 3 に示す LR 表を得る（表中の*印，√印，×印の意味は後述）．結論を言えば，状態 15 の先読み記号 p（前置詞）で生じるコンフリクト中の動作 re10 を削除すればよい．これをどうして知ることができるだろうか．

2.3 コンフリクト解消のために削除すべき動作の決定法

解消すべきコンフリクトの場所と削除すべき動作を決める簡便で一般的な技法を与える．まずはじめに，所望の統語解析結果（統語構造）を与え，この解析結果だけが得られるように LR 表中の動作を次々に起動する GLR パーザを用意しておく．このパーザを

状態	action						goto			
	n	det	and	v	p	$	S	NP	VP	PP
0	*sh2	sh1					4	3		
1	sh5									
2			*re4	*re4	*re4	*re4				
3			*sh10	*sh6	*sh8				7	9
4			*sh12		sh8	*acc				11
5			re5	re5	re5	re5				
6	*sh2	sh1					14	13		
7			*re1		re1	re1				
8	*sh2	sh1						15		
9			re6	*re6	√re6	re6				
10	*sh2	sh1						16		
11			re2		√re2	2				
12	*sh2	sh1					17	3		
13			*re8/×sh10	sh6	×re8/*sh8	*re8			7	9
14			re9/sh12	re9	re9/sh8	re9				11
15			re10/sh10	*re10	×re10/*sh8	*re10				9
16			×re7/*sh10	*re7	re7/sh8	re7				9
17			×re3/*sh12		re3/sh8	*re3				11

図 3: Grammar I から得た LR 表

GLRT(GLR Tracer) とよぶことにする．GLRT を用いて次の手順でコンフリクト中の削除すべき動作を容易に決定することができる．

1) 所望の各統語解析結果を GLRT に与え，GLRT は実行した LR 表中の動作に*印を付ける．

2) *印付きと無印の動作が混在するコンフリクトがあれば，無印の動作を削除してコンフリクトを解消する．

3) 制約伝播（後述）を施し LR 表を圧縮する．

前記の文 1 の他に以下に示す文 2，3，4 に対しても所望の統語解析結果が GLRT に与えられるとしよう．LR 表は図 3に示すもの（ただし図中の記号を無視）を用いる．

1. Mary:n with:p a:det book:n for:p cooking:n loves:v John:n. 図 2の 1).
2. John:n loves:v Mary:n with:p a:det book:n for:p cooking:n.

```
[S [NP n]
   [VP v [NP [NP n]
             [PP p [NP [NP det n]
                       [PP p [NP n]]]]]]]
```

3. Mary:n and:and Betty:n and:and John:n keep:v sheep:n.

```
[S [NP [NP n] and
       [NP [NP n] and
           [NP n]]]
   [VP v [NP n]]]
```

4. Mary:n likes:v John:n and:and Betty:n likes:v John:n and:and John:n likes:v Mary:n.

```
[S [S [NP n] [VP v [NP n]]] and
   [S [NP n] [VP v [NP n]]]]]
```

上記の 1，2 は主名詞に付加する PP が右下がりの構造に，2，3 の and でつながれた S と N は，ともに右下がりの構造にする．これらの統語解析結果を GLRT に与えて実行した動作が図 3に*印で示されている．×印のついた動作が削除すべき動作である。4章で説明するが，√印の付いた動作は制約伝播により削除される動作である．

2.4 コンフリクト解消の限界

GLRT によるコンフリクト解消技法が適用できない場合がある．次の CFG を考えてみる．

(1) S → N V　　(3) N → n

(2) N → N N　　(4) V → n v

図 4: Grammar 2

図 4の CFG から得た LR 表を用いて GLRT に次の統語解析結果を与える：

[S [N [N n] [N [N n] [N n]]] [V n v]]

GLRT は唯一のコンフリクトを構成する二つの動作 re2 と sh4 に共に*印を付けるのでコンフリクトが解消できず，与えた統語解析結果の他の統語解析結果をも得てしまう．

3 規則の再帰的な適用可能回数の制限

再帰的な CFG 規則とは，規則の左側 (LHS) に現れる非終端記号が，規則の右側 (RHS) にも現れるものを

いう: $X \rightarrow \alpha X \beta$. ここで$\alpha \neq \epsilon$なら, この再帰規則を右方再帰規則とよぶ.

前記した右方再帰規則を再帰的に適用することにより, X の左と右に, それぞれαが n 個, βが n 個続く構造が受理可能になる. これを避けたいことがある. 自然言語を解析するための CFG を子細に観察すると多数の再帰規則が含まれていることが分かる. たとえば高倉 [高倉 84] が作成した 394 個の英語の CFG 規則がある[2]. この中にはたとえば次の右方再帰規則が含まれている : ADJP → not so ADJP ASCOMP
この CFG 規則は, 任意個の "not so" が ADJP (形容詞句) の左に現れる構造を受理することができるが, これは明らかに文法記述者の意図とは異なる. 文法記述者は ADJP の左にただ 1 度の "not so" が現れる構造を受理するものとしてこの規則を記述している. したがってこの規則の適用回数を 1 回に制限する方策が必要になる[3].

事前に修正した LR 表を用いることで, 特定の右方再帰規則, たとえば$Y \rightarrow \alpha Y \beta$ の適用回数を容易に 1 回に制限することができる. これは次のようにする.

LR 表を作成する段階で作成するクロージャI (状態 I) に, アイテム $[Y \rightarrow \cdot \alpha Y \beta; ...]$ がはじめて付加されたと仮定する. クロージャI からαに相当する GOTO を次々に行って到達したクロージャJ にはアイテム $[Y \rightarrow \alpha \cdot Y \beta; ...]$ が含まれている. これを核 (nucleus) アイテムとして, クロージャJ にアイテム $[Y \rightarrow \cdot \alpha Y \beta; ...]$ を付加する操作をクロージャの拡張という. ところが拡張操作により付加するアイテムはαを引き続き受理するためのアイテムであるから, もしαの受理を一回に限定したければ, クロージャJ の拡張時に, アイテム $[Y \rightarrow \cdot \alpha Y \beta; ...]$ を付加してはならない. このようにして得た GOTO グラフより LR 表を作成する. この LR 表を使うことにより特定の右方再帰規則の適用回数を 1 回に制限することができる。

4 接続制約の LR 表への組み込み

自然言語の統語解析で用いる文法的な枠組みとしてこれまで CFG がよく使われ, また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使われてきた. この局所的な制約は接続表として表すことができる. 隣接する記号間の接続可能性という局所的な制約は, CFG の枠組みで原理的に記述可能であるにしても, 規則数が増えるという問題がある. CFG 規則に手続きを付加し, これを動作させて隣接する記号間の接続可能性を調べる方法もあるが, これは手続きの記述を複雑にするので避けたい. この問題を解決する技法を以下で説明する.

隣接する記号間の接続可能性を n 行 n 列の接続表として表すことができる.

> 記号 vi と記号 vj が, この順に接続可能なら connect[vi,vj] = 1, さもなければ connect[vi,vj] = 0

以上により, CFG の記述と接続表の記述とを独立させることができる. 接続表の制約を LR 表に組み込む手続き (a), (b) を以下に与える (詳細は [田中 95]).

(a) LR 表の初期化手続き: 動作 Act に対して, Act の直前に実行する動作がシフト動作 Sh であるようなすべての Act に対して, Sh の先読み記号が v, Act の先読み記号が w であるとき, connect[v, w] = 0 なら Act を削除する.

(b) 制約伝播手続き: 削除すべき動作がなくなるまで, 以下を繰り返す.

- LR 表中の各動作 Act について, Act の直前または直後に実行するアクションが一つもなければ, Act を削除する.

(c) LR 表の空の行と列を圧縮する.

手続き (a) により, 接続表の制約を満たさない動作が削除される. この削除により, 手続き (b) で新たな動作の削除が連鎖的に可能になることがある. これが制約伝播である. CFG のサイズが大きくなれば, LR 表の状態数, レデュース数, シフト数が制約伝播により大幅に減少する [Li 96]. この技法の応用として, 日本語の形態素解析と統語解析とを統合化して行なう MSLR システム [Tanaka 95] がある.

LR 表を作成してから接続表の制約を加えて LR 表中の動作を削除する上記の技法は, 計算時間, メモリー効率の点で問題がある. 実際には LR 表を生成する段階で接続表の制約を使う [田中 95]. 綾部は音素のレベル, 形態素のレベルといった複数の接続表を LR 表の生成段階で組み込むアルゴリズムを開発している [綾部 98]. これは将来音声認識システムに組み込む予定である.

5 新しい確率 GLR

CFG 規則に確率を付与した PCFG の考え方が自然言語解析に応用され注目されている [Fujisaki 89]. 解

[2] 高倉が省略可能とした文法記号を無視した不完全な CFG が Grammar IV として [Tomita 96] の付録に採録されている. そこには 35 個の右方再帰規則があり, そのうち 19 個は再帰的な規則適用回数が 1 回限りの規則であった.

[3] 現実の文では ADJP の左に "not so" が繰り返し並ぶことはないので, 再帰規則の適用回数を制限する仕組みは使われない. したがって本章で提案する技法は, 実際に使われることのない技法であり意味があるとは思えないと反論する読者がいるかも知れない. しかし音声認識に CFG を利用しようとするとこの問題が直ちに顕在化する. 音声認識システムでは, 次に現れる音素や単語の予測の精度を上げることが探索の範囲を狭め, システムの高速化と認識の精度を上げることに直結する. 本章で説明する技法を用いないと, "not so" の次に本来予測すべきでない "not so" の存在を予測してしまうので予測の精度が低下する. これは好ましいことではない. 予測の精度を上げるために本章で説明する技法は重要である.

析木を構成するCFG規則のもつ確率の積を，解析木のスコア（確率値）とするのであるが，これでは脈絡に依存した確率の計算ができない．

個々のCFG規則に確率を付与するのではなく，LR表中の各動作に確率値を振っておき，動作を実行する毎に確率値の積を計算する方法が開発されている [Briscoe 93]．これを確率GLR法(PGLR)とよぶ．LR表中の動作は，先読み記号を右脈絡，スタックトップの状態を左脈絡として実行する．そのため動作に振る確率には，自然と左脈絡と右脈絡が反映される．PGLRでは動作が実行される度に，その動作に振られた確率の積を計算して統語解析結果のスコアとする．

我々は最近，Briscoeらの提案した各動作に振る確率値の計算法には，正規化に関する重大な理論的欠陥があることを指摘し，理論的に正しい確率値の計算法を見い出している．その詳細は [Inui 97] に譲るが，それにより，日本語のCFGと [田中 97]，約1万の正しい統語解析木 (構造付きコーパス) を用いてLR表中の動作に確率を振り，[Tanaka 95] に述べたMSLRシステムを用いた統語解析の実験を行なった．最もスコアの高い統語解析木を一つ取り出す実験では，極めて高い解析精度（90%から95%）を得ている．過去の方法（PCFG，Briscoeらの方法，Charniakらの方法）と比べても，我々のPGLR法が最も解析精度が高い [Sornlertlamvanich 98]．

最後に田中研究室では，バイグラムの確率をLR表中の各動作に与える方法を開発している [Li 96][Imai 97]．それによれば，バイグラムの確率をLR表中の動作に与える方法は，バイグラムのみを用いる方法より，Test Set Sentence Perplexity(TSSP) を減少させることが確認されている．これは音声認識システムの作成に好都合であり，音声認識実験によっても確認されている [Li 96]．われわれは現在この方法とPGLR法とをTSSPの立場から比較検討している．

6 おわりに

LR表工学と名付けた自然言語処理の新しい方法を提案した．それによれば，LR表に修正を加えることで，これまで可能であるにしても煩雑で困難であった問題が容易に，しかもGLR法とよばれる一つの統一した枠組みの中で自然に解決できることを示した．今後の課題として以下のものがある．

第一に，LR表工学のためのソフトウェアツールを準備し，多くの研究者にLR表工学の研究環境を提供したい．次に，対話システム，機械翻訳システム，音声認識システムなど大規模な言語処理応用システムの構築を試みたい．

第二に，一つのLR表に三つ以上の接続表の制約を組み込み，4章の方法を一般化することも理論的には意味あることと思われる．

第三に，PGLRでは構造付きの大量のコーパスが必要になる．これは現在手作業で作成しているため，量的な拡大に問題がある．5章で説明した方法を用いて，自動的に構造付きのコーパスを作成する試みにも着手したい．

参考文献

[Aho 75] Aho, A.V., Johnson, S.C. and Ullman, J.D. *Deterministic Parsing of Ambiguous Grammars.* Comm. of ACM, vol.18, No 8, pp. 441-452(1075).

[綾部 98] 綾部寿樹. 複数の接続表の制約の *LR* 表への組み込みと実装化. 東京工業大学大学院情報理工学研究科修士論文 (1998).

[Briscoe 93] Briscoe T. and Carroll J. *Generalized Probabilistic LR Parsing of Natural Language (Corpora) with Unification-Based Grammars.* Computational Liguistics, vol.19, No 1, pp. 25-59(1993).

[Fujisaki 89] Fujisaki T. Jelinek F. et.al. *A Probabilistic Method for Sentence Disambiguation.* 1st International Workshop on Parsing Technologies, CMU, pp. 105-114(1989).

[Imai 97] Imai H., Li H. and Tanaka H. *Incorporating Bigram Constraints into an LR Table.* Proc. of ROCLING X, pp. 76-88, (1997).

[Inui 97] Inui K, Sornlertlamvanich V., Tokunaga T. and Tanaka H. *A New Formalization of Probabilistic GLR Parsing.* International Workshop of Parsing Technologies, pp. 123-134 (1997).

[Knuth 65] Knuth D.E. *On the Transition of Languages from Left to Right.* Information and Control, 9, 607-639 (1965).

[Li 96] Li H. *Integrating Connection Constraints into a GLR Parser and its Applications in a Continuous Speech Recognition System.* TR96-003, Dept. of Computer Science, Tokyo Institute of Technology, (1996).

[Sornlertamvanich 98] Sornlertamvanich V., Inui K., Tanaka H. and Tokunaga T. *Effectiveness of a New Generalized LR Parsing.* submitted to Colin98.

[高倉 84] 高倉伸. *Prolog* による英語のボトムアップ構文解析. 東京工業大学工学部学士論文 (1984).

[田中 95] 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)*, 人工知能学会研資 SIG-J-9501-1, pp. 1-8 (1995).

[Tanaka 95] Tanaka H., Tokunaga T., and Aizawa M. *Integration of Morphological and Syntactic Analysis based on LR Parsing.* Journal of Natural Language Processing, 2, 2, pp. 59-74(1995).

[田中 97] Tanaka H., Takezawa T. and Etoh S. *MSLR* 法を考慮した音声認識用日本語文法–*LR* 表工学 *(3)*. 音声言語情報研究会資料, 情報処理学会, 15-25, pp. 145-150(1997).

[Tomita 86] Tomita M. *Efficient Parsing for Natural Language.* Kluwer Academic Publishers (1986).