

パージング

—制約統合型モデルの提案—

Incorporating Multiple Syntactic Constraints into a Parsing Technology

田中 穂積*

Hozumi Tanaka

* 東京工業大学大学院情報理工学研究科
Graduate School of Information Science and Engineering, Tokyo Institute of Technology.

1996年4月22日 受理

Keywords: parsing, syntactic analysis, morphological analysis, generalized LR(GLR) parsing.

1. はじめに

カスケード型・ヘテロジーニクス型から
制約統合型モデルへ

広義の意味での自然言語のパージングにはさまざまな意味が含まれている。本稿では、狭義の意味でのパージング、すなわち文脈・意味解析などといった高次のレベルではなく、低次のレベル(統語解析まで)のパージングについて、最近筆者らが提案している手法を中心に説明する。これが、将来のパージング技術の目指す方向を示唆していると思われるからである。

これまでの自然言語処理は、異なるレベルの制約(情報)を異なるモジュールで処理し、それらをカスケードに結合して解析を行う方式が中心であった(図1)。モジュール相互間に弱いインタラクションを持たせるモデルも考えられたが(図1中の点線)、それでも解析モジュールごとに個別の解析アルゴリズムを用意しなければならないことには変わりなかった。

このことを、分かち書きされていないべた書きの日本語の文の統語解析に見ることができる。通常二つの解析モジュールをカスケードに結合した計算モデルを用いる。形態素解析モジュールと統語解析モジュール

である。前者は形態素間の接続可能性に関する制約(接続表)を用いて形態素を抽出する(分かち書きする)。後者は統語的な制約(文脈自由文法など)を用いて統語構造を抽出する。形態素解析モジュールでは、接続表のみを利用するため制約があまく、通常多数の形態素解析結果が得られてしまう。解析結果を絞り込まなければ、後に控える統語解析モジュールの負担が大きくなり、効率的な解析はおぼつかない。そこで解析結果を絞り込むために、対象分野に強く依存した経験則を導入したり、辞書に特殊な項目を登録して対応することになる。

ここで人間がどのようにして自然言語や音声を解析し理解しているかを考えてみたい。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さまざまな曖昧性を早期に解消し効率良く解析を進めていると思われる。おそらく形態素間の接続可能性に関する制約や文法的な制約などは同時に利用しているに違いない。多数の制約を同時に利用する方法として、制約ごとに異なるモジュールを用意し、それらをヘテロジーニクスに(ネットワーク状に)結合した計算モデルが考えられる。人工知能の分野でこのような計算モデルの必要性が強く主張されたことがある。しかし、

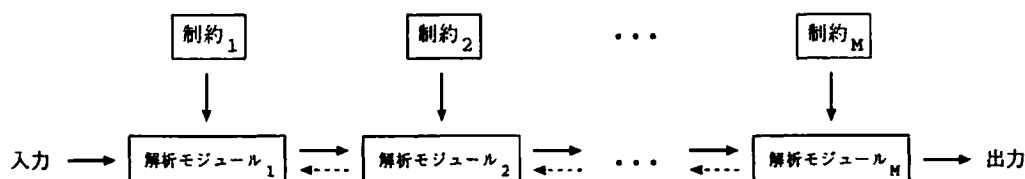


図1 カスケード結合による処理

こうしたモデルはモジュール相互間をどう結合するか、複数のモジュールの動作をどう制御するかが問題になり、システム全体を見通し良く設計し、正常に動作するシステムを構築することが難しい。そのための一般的な原則があるようにも思えない。

制約に応じて異なるモジュールを用意するのではなく、ただ一つのモジュール(解析用の枠組み)を用いて、異なるレベルの複数の制約を同時に利用し効率良く解析を行う計算モデルがあれば申し分ない(図2)。自然言語の解析用として、このような計算モデルの構築は可能だろうか。制約の種類によってはそれは可能である。筆者らは最近そのようなモデルを提案している。以下の章ではそれを紹介する。我々のスローガンを一口でいえば「カスケード型・ヘテロジニアス型から制約統合型のパーズングへ」ということになる。

上記したスローガンに沿った筆者らのパーズングモデルでは、解析用の枠組みとして、統語解析の典型的なアルゴリズムの一つである一般化LR(GLR: generalized LR)法をそのまま用いる[Aho 86, Chapman 87, Tomita 86, Tomita 91]。この唯一の解析用の枠組みで、異なるレベルの複数の制約を同時に利用する場合、解析用の枠組みが複雑化するのであれば問題である。興味深いことに、GLR法で用いる表(LR表*)は異なるレベルの制約をさらに組み込むことにより簡単化することがある。筆者らの計算モデルでは、形態素間の接続可能性を示す接続表の制約と文脈自由文法(CFG)の制約を一つのLR表に組み込むのであるが、形態素間の接続可能性の制約を組み込むことによりLR表のサイズは縮小する(4章参照)。この縮小したLR表を用いて、GLR法という一つの枠組みで二つの制約を同時に用いた解析を行うことができる。こうして、むしろ解析用の枠組み(図2の解析モジュール)は簡単化する。

ところで、隣接する終端記号間の接続可能性に関する制約をLR表に組み込むにはどうしたらよいらろうか。2章ではその基本的な考え方を説明する。3章では制約伝播という考え方を導入する。それにより2章で組み込んだ制約をLR表に伝播させ、LR表をさらに簡単化することができる[Tanaka 95a]。こうして、形態素解析と統語解析とを、GLR法という一つの枠組みで完全に統合化した解析を行うことができる。これを

*1 文法が与えられれば、これからLR表を作ることができる。したがって、文法がCFGであれば、この表にはCFGの制約が組み込まれている。LR表は入力文を先読みし、パーズの動作を決める指令書である。CFGからLR表を作成する方法は[Aho 86, Chapman 87]に詳しい。

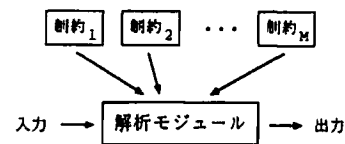


図2 制約統合化による処理

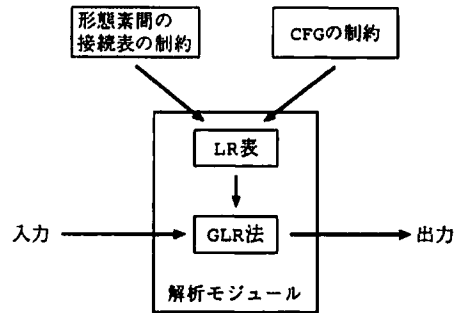


図3 形態素解析と統語解析の統合化処理

図2と対応させれば、解析モジュールではGLR法を用い、制約として形態素間の接続表とCFGを用いることになる(図3)。本方法の有効性を確認するための実験を行った。その結果を4章で述べる。なお、以下の説明では、必ずしもGLR法の詳細を知らない読者でも読み進むことができるように配慮したつもりである。

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がよく使われてきた。形態素解析では、形態素間の接続可能性という局所的な解析がよく使われてきた。この局所的な制約は、後述する接続表として表現することができる。隣接する二つの記号間の接続可能性というきわめて局所的な制約は、CFGの枠組みで原理的に記述可能であるにしても、いくつかの問題点がある。これらを解決する方法を以下で説明する。

本章以降の説明で表れるギリシャ文字の 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が混在した記号列。

2.1 二つの制約: CFGと接続表

局所的な文法的制約の一つに、終端記号の集合 $V_t (= \{v_1, v_2, \dots, v_n\})$ の各要素間の接続可能性についての制約がある。この制約は日本語など、単語と単語との間に空白を置かない言語の形態素解析によく用いられている。例えば、ら行五段活用の動詞の語幹(決ま)には、ら行五段活用の動詞語尾(ら, り, る, れ)はつく

| | | | | | | | | |
|------|----------------|----------------|-----|----------------|-----|----------------|-----|----------------|
| | RIGHT | | | | | | | |
| | v ₁ | v ₂ | ... | v _i | ... | v _j | ... | v _n |
| LEFT | v ₁ | | | | | | | |
| | ... | | | | | | | |
| | v _i | | | 1 | | 1 | | |
| | ... | | | | | | | |
| | v _j | | | 0 | | 1 | | |
| | v _n | | | | | | | |

図4 接続表の例

- (1) $V \rightarrow \gamma v_i$
- (2) $V \rightarrow \delta v_j$
- (3) $W \rightarrow v_i \zeta$
- (4) $W \rightarrow v_j \eta$
- (5) $X \rightarrow \alpha V$
- (6) $Y \rightarrow w \beta$
- (7) $Z \rightarrow X Y$

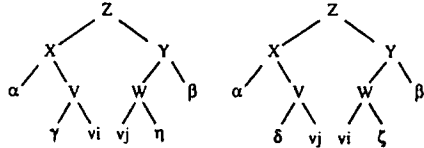


図5 CFGと接続可能性制約

が、か行五段活用の動詞語尾(か, き, く, け)はつかないなどという接続可能性の制約である。このような隣接する記号(要素)間の接続可能性は n 行 n 列の表(connect)として表すことができる。これを接続表と呼び、以下のように定義する(図4参照)。

- (1) 記号 v_i と記号 v_j が、この順に接続可能なら、
connect [v_i, v_j]=1
- (2) 記号 v_i と記号 v_j が、この順に接続不可能なら、
connect [v_i, v_j]=0

二つの記号間の接続可能性の制約は CFG 規則を用いてももちろん記述できる。例えば CFG 規則の右辺に、文法カテゴリーの接続可能性を直接記述する。

$$X \rightarrow \alpha v_i v_j \beta$$

この規則の右辺は、 v_i と v_j がこの順に隣接可能であることを直接記述している。

ここで図5に示す CFG を考えてみる。図5の各 CFG 規則の右辺のどこにも、 v_i と v_j (または v_j と v_i) の接続可能性が記述されていない。 v_j と v_i がそれぞれ異なる CFG 規則の右辺に現れているので、 v_j と v_i がこの順に接続不可能であるという制約がこのままでは記述されていないことになる。この場合、 v_j と v_i がこの順に接続不可能であり、それ以外の接続($v_i v_i, v_i v_j, v_j v_i$)が可能であるという制約を CFG でどう記述したらよいだろうか。

新しい非終端記号を導入し図6に示す CFG にすることが考えられる。新しい非終端記号を導入して、 v_i と v_j の接続可能性を、 $\{X_i, X_j\}$ の要素と $\{Y_i, Y_j\}$ の要素間の接続可能性を記述した(9)~(11)の規則で置き換えるのである。

図6の CFG を一べつすると、新しい非終端記号の

- (1) $V_i \rightarrow \gamma v_i$
- (2) $V_j \rightarrow \delta v_j$
- (3) $W_i \rightarrow v_i \zeta$
- (4) $W_j \rightarrow v_j \eta$
- (5) $X_i \rightarrow \alpha V_i$
- (6) $X_j \rightarrow \alpha V_j$
- (7) $Y_i \rightarrow w_i \beta$
- (8) $Y_j \rightarrow w_j \beta$
- (9) $Z \rightarrow X_i Y_i$
- (10) $Z \rightarrow X_i Y_j$
- (11) $Z \rightarrow X_j Y_j$

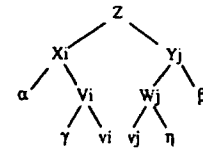


図6 接続可能性を考慮した CFG

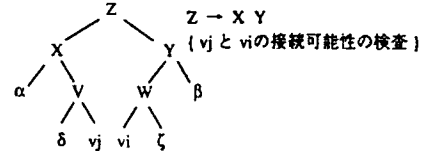


図7 接続可能性検査のタイミング

導入により CFG 規則の数が組合せ数的に増大し、CFG の骨格が不透明になっていることがわかる。これは文法記述の立場から好ましくない。解析の立場からも、CFG 規則の増大は時間的・空間的な消費をもたらすので好ましくない。

新しい非終端記号を導入しない方法として、図7に示すように、例えば図5の各 CFG 規則に、接続可能性の制約を付記しておく方法や、ユニフィケーション機構を利用する方法が考えられる。CFG 規則に付記する制約は、手続き付加、補強項などと呼ばれている[田中89]。解析過程で CFG 規則の適用時に、そこに付記されている制約をチェックするのである。このとき、補強項で使う制約は解析木に沿って移動し、他の文法規則の補強項でも使われることに注意したい。ところが、他の文法規則の補強項のことまで考慮に入れた補強項の記述は複雑になり、特に大規模な文法規則の記述を困難にする。CFG 規則でカバーする先の方法と同様に、接続可能性のチェックのタイミングが遅れるという欠点もある。さらに図7の例でいえば、もし v_j と v_i がこの順に接続不可能なら、 Y という大きな構造を作成したこと自体がむだになる。

以上の考察から、文法記述の立場から解決すべき問題を列挙すると以下ようになる。

- (1) 文法記述者は、接続可能性を考慮した新たな非終端記号の導入を行うことなしに、CFG の記述が可能であること。
 - (2) 接続可能性を表す接続表の記述者は、CFG 規則とは無関係に接続表の記述が可能であること。
 - (3) 局所的な制約である接続可能性の検査のタイミングは、早ければ早いほどよい。
- (1)と(2)は、文法記述と接続表記述とを独立させたい

ということである。GLR法の枠組みで、(1)~(3)の条件をすべて満足する解決法を2.2節で説明する。

2.2 接続表の制約のLR表への組み込み：基本的な考え方

終端記号の集合 $V_t = \{v_1, v_2, \dots, v_n\}$ の各要素を先読み語としよう。そして $v_j v_i$ という接続がこの順に不可能で、 $v_i v_i, v_i v_j, v_j v_j$ という接続がこの順に可能であるとしてしよう。

connect $[v_i, v_i] = 1$

connect $[v_i, v_j] = 1$

connect $[v_j, v_i] = 0$

connect $[v_j, v_j] = 1$

図5に示したCFGでは、上記した接続表の制約が記述されていないということをすでに説明した。そこで以下では、図5に接続表の制約を組み込む方法を考えてみたい。

図5のCFGから、図8に示すLR表が得られたとする。ここで v_i, v_j などは先読み語で、 m, n などはパーザの状態である。sh n, re 2 はそれぞれシフトアクション、レデュースアクションと呼び、パーザが行う動作である。GLRパーザはLR表に従った動作を行うが、これをGLR法によるパーシングと呼ぶ。GLRパーザは、現在のパーザの状態と先読み語とから決まるアクションをLR表から決めて動作を行う。例えば図8では、現在の状態が m で先読み語が v_j のとき、パーザが行うアクションは sh n である。この sh n は、 v_j をパーザが読み込み、パーザの状態を n に移行(状態 n にシフト)するアクションを表す。こうして、パーザの状態が n にシフトした直後の先読み語は、 v_j ではなく v_j の次の語になっていることに注意。

一方、sh n を実行して到達した新しい状態 n において、先読み語が v_i のときに行うアクション re 2 は、図6の(2)の $V \rightarrow \delta v_j$ という規則を適用し、図8の右に示す木を作る動作を行う。この場合には先に読み込んだ語 v_j と先読み語 v_i とがこの順に接続していることになるが、接続表からこの接続が許されていないとすれ

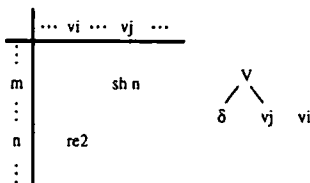


図8 LR表の例1

ば、re 2(状態 n で先読み語が v_i のときに実行するアクション)を実行しても制約に違反した構造を作ることになり意味がない。そこでこれをLR表から削除する。このことは、シフトアクションにより v_j を読み込み移行して到達した状態 n で、先読み語を v_i として実行するアクションがレデュースであろうとシフトであろうとそれとは無関係に、そのアクションは接続表の制約に違反するので(LR表から)削除可能である、ということの意味している。

以上をまとめると、接続表の制約により、LR表からアクションを削除する手続きを次のように述べることができる。

アクション Act の直前に実行するアクションがシフトアクション Sh であるようなすべての Act に対して、Sh の先読み語が v 、Act の先読み語が w であるとき、connect $[v, w] = 0$ なら Act をLR表から削除する。

こうして接続表の制約を組み込むことによりLR表のサイズが縮小し簡単化するのであるが、実はLR表をさらに簡単化することができる。これについて次章で述べる。

3. 制約伝播によるアクションの削除

2.2節で述べた方法により、接続表の制約に違反するアクションをLR表からすべて削除する。その結果、LR表の他のアクションの削除が連鎖的に可能になることがある。これを制約伝播と呼ぶ。例えば、2.2節に述べたアクションの削除により、LR表上で、アクションがすべて削除された状態(空状態)ができてしまうことがある。空状態にシフトするアクションがあるとき、空状態へシフトしてもその後の解析が続けられないので意味がない。そこで空状態へシフトするシフトアクションも削除することができる。

これを一般化して述べると次のようになる。Act と呼ばれるアクションを実行した直後に実行すべきアクションがLR表中に一つもなければ、Act の実行直後にエラーが生じるのは明らかである。したがって、このActを実行しても解析が続けられないのであるから、これを削除してよい。一方、Actの直前に実行するアクションが一つもなければ、このActは永久に実行されない無効なアクションであるから、このアクションを削除してもよい。これらを以下にまとめてみる*2。

(a) LR表の初期化手続き：アクション Act に対して、Actの直前に実行するアクションがシフトアクション Sh であるようなすべての Act に対し

*2 手続き(a)と(b)は[Tanaka 94]を一般化し洗練したものである。

て, Sh の先読み語が v, Act の先読み語が 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: 削除すべきアクションがなくなるまで, 以下を繰り返す.

LR 表中の各アクション 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 表の空の行を圧縮する.
(d) LR 表の空の列を圧縮する.

4. 実験結果

これまでに説明した自然言語処理の方法の利点を確認するための実験を行った. 実験には ATR が試作中の音声認識用の日本語文法(会議予約対話文用)を用いる.

2.1 節では, 文法記述の立場から解決すべき問題を列挙したが, そのなかで文法記述と接続表記述を独立させることの利点が指摘されている. ATR の音声認識用日本語文法は, 形態素の接続可能性を考慮した CFG を用いて, 形態素解析と統語解析の統合化を図り, 音声認識の精度を向上させようとしている. 2.1 節で説明したように, このような方法では, 新しい非終端記号の導入に伴い文法規則の数が増え, 文法体系の見通しが悪くなるので, 文法規則を修正したり拡張する場合の障害になる.

そこでこの文法規則を, 形態素間の接続関係は接続表として記述し, 文法規則は形態素より上位のレベルの規則のみとして分離して記述することを試みた. 分離以前の文法を旧 ATR 文法, 分離後の文法を新 ATR 文法と呼ぶことにすれば, それぞれの規則数は 1809 と 858 であった. 新 ATR 文法の規則数は旧 ATR 文法の半分以下(47.4%)で, 規則数にして 951 減った*3.

ここで, 新 ATR 文法規則には, 形態素より上位のレベルの記号と形態素レベルの記号とを結ぶ規則が含まれている. これは, 形態素より上位の非終端記号と形態素の間を結ぶユニット規則(これを細品詞規則と呼ぶ[Tanaka 95a])として実現される*4. 新 ATR 文法規則にはこれら細品詞規則も含まれており, この細品

詞規則数は 283 であった. そこで, これらを差し引くと実質的な新 ATR 文法規則数は 575 である. したがって, 実質的な文法規則数で比較すれば, 新 ATR 文法の規則数は旧 ATR 文法のその 31.8% であるに過ぎない. 以上のことから, 新 ATR 文法の記述が旧 ATR 文法の記述に比べて容易であることは明らかだろう.

形態素間の接続表を手作業で作成するのは労力を要する. しかし, 形態素レベルのタグ付きコーパスがあれば, 形態素間の接続表はこれから自動的に抽出できる. 実際植木らは EDR コーパスから, 形態素間の接続表を自動的に抽出している[植木 95].

次に, 制約伝播により最終的に LR 表がどの程度単純化されるかを見てみよう. 新 ATR 文法に対して, 形態素間の接続表の制約を LR 表に組み込む以前と以後の LR 表のサイズがどう変わるかを, 状態数, アクション数で比較してみた*5.

| | シフトアクション数 | レデュースアクション数 | 状態数 |
|------------|-----------|-------------|-------|
| 組込み前の LR 表 | 46 301 | 911 855 | 3 532 |
| 組込み後の LR 表 | 8 012 | 7 762 | 793 |

形態素間の接続表の制約を組み込むことにより, アクション数, 状態数とも激減する. 筆者らの研究室で開発した日本語文法でも同様な傾向を得ている. このことから, 接続表の制約を組み込むことにより, LR 表のサイズは劇的に減少し単純化すると結論することができる. このことは, メモリ空間だけでなく解析速度の向上にも寄与する.

5. あとがき

筆者らの方法を要約してみよう.

- ・接続可能性を考慮した新たな非終端記号の導入なしに CFG の記述が可能.
- ・接続表の記述者は, CFG 規則とは独立に接続表の記述が可能.
- ・接続表の制約を LR 表に書き込むことにより, CFG の制約と接続表の制約とを同時に用いた解析が可能. 解析に用いる枠組みはただ一つ, GLR 法である. これをそのまま用いて, 形態素解析と統語解析とを完全に統合化した解析が可能.
- ・接続表の制約を LR 表に組み込むことにより, LR 表のサイズが小さくなり, 単純化する. それによりむだな動作を避けた効率的なパーシングが可能.

*3 新 ATR 文法は衛藤純司氏(ランゲージウェア社)が田中研究室で試作した文法である.

*4 例えば名詞の細品詞に固有名詞があれば, 細品詞規則は単純に, 名詞→固有名詞, である. EDR 辞書からは細品詞規則を自動的に抽出することができる[Tanaka 95].

*5 ただし, シフトアクション数には GOTO アクション数が加算されている.

本文中で触れられなかったその他の利点を以下に述べる。

本稿で説明した自然言語解析の新しい方法は、形態素解析と統語解析とを統合する方法として有効であるだけでなく[Tanaka 95a, 植木 95], 音声認識にも有効であることが実験的に示されている[Li 94, Tanaka 94]. この場合には、形態素間の接続表の代わりに異音間の接続表を用いる。

これまで述べてきた接続表やCFGによる制約とは異なる制約として統計的な制約がある。この制約は、解析結果の絞り込みによく使われるようになってきた。幸いにして、この種の制約をLR表に素直に組み込むことができる。確率文法(PCFG)[Fujisaki 91]の各規則に付与された確率を、LR表のアクションに割り付ける方法はWrightが与えている[Wright 90]. コーパスをGLR法で解析し、LR表中の各アクションの実行頻度から確率を推定しそれをアクションに割り付け、後の解析で利用する方法は[Briscoe 93]が与えている。

英語の“of”や日本語の助詞「の」などを含む名詞句や前置詞句付加(PP付加: PP-attachment)の解析結果には、構造的な曖昧性が多数含まれることはよく知られている。この種の構造的な曖昧性解消には意味的な情報が必要になる。そこで統語解析のレベルでは、右下りあるいは左下りの統語解析結果のみ抽出することにし、正しい構造は、後続する意味解析や文脈解析に任せる方法が考えられる。CFGに手を加えることなく、右下りあるいは左下りの統語解析結果のみを得るための方法がある。LR表のエントリーに複数のアクションがある(コンフリクトが生じた)とき、一つのアクションのみを残し他をすべて消去するという方法である[Aho 75].

頑健なパーズングについてGLR法はどうか。現実の音声対話に含まれる「あー」、「えーと」などの冗長語や、音声認識誤りによるノイズをスキップし無視するGLR*と呼ばれる方法がある[Lavie 93]. これは、GLR法に頑健さを与えるものである。ノイズ以外の現象として、どもり、繰返し、倒置など、これまでの解析法では不具合であった言語現象に対しても適応可能な方法を今後研究する必要がある。なお、筆者らはGLR*に似た方法を、長文に含まれる並列句の効率的な検出に応用可能かどうかを検討している。

辞書引きもGLR法の枠組みで行うことができる。それによれば、LR表を用いてトライ構造以上に辞書データを圧縮することができる[センリ 96]. この場合

には、辞書引きもGLR法の枠組みで行うことになるので、図3の接続表の制約とCFGの制約と同様な位置づけで、辞書の箱が付け加わることになるだろう。

説明を簡単にするために、2.2節では既存の標準的なLR表作成アルゴリズムをそのまま使う最もナイーブな方法を述べた。この方法の欠点は、CFG規則数が増え数千の規模になると、(制約伝播を施す前の)中間段階で生成するLR表のサイズが莫大になり実用に耐えないことである。この問題を解決するために、筆者らは、LR表生成アルゴリズムに手を加え、LR表を作るときに接続表の制約を組み込みむだなメモリを使わないLR表作成アルゴリズムを開発している[Li 95, 田中 95b]. 実験によれば、標準的なワークステーション上でのLR表生成時間も、規則数が1208の規模のCFGに対して30秒程度と高速である。

今後の課題として、複数の接続表の制約をLR表に組み込む方法の開発がある。例えば音声認識の場合には、CFGの制約のほかに、異音間の接続表の制約と形態素間の接続表の制約をともにLR表に組み込みたいことがあるからである。我々は一つの解法をすでに見出ししているが、これについては別の機会に譲りたい。

本稿では、GLR法という一つの確立した自然言語解析の枠組みをそのまま用いて、形態素解析、統語解析を完全に統合化した自然言語処理を行う方法を説明した。この方法は、これに音素レベルの制約を統合可能であり、音声認識から形態素・統語解析までを、GLR法の枠組みだけで完全に扱うことが可能なことについても触れた。さらに、この枠組みのなかに、統計的な制約を組み込むことも容易であることにも触れた。筆者は、本稿で説明した方法にはさまざまな技術的な利点があり、人間の行っている自然言語処理の方法にも近いと考えている。

最後に、「カスケード型・ヘテロジーニアス型から制約統合型へ」というスローガンで、パーズング技術の一つの新しい方向を説明してきたつもりであるが、このことは、パーズング技術以外の人工知能の分野で、つねにこうした方向を目指すべきであるということも必ずしも意味していない。パーズング以上に複雑な問題解決の分野では、カスケードやヘテロジーニアス型の計算モデルが必要なこともあるだろう。ただし、その場合であっても本稿で述べたように、単純で理論的な裏づけを有する制約統合型の計算モデルを模索することは意味があるだろう。そうした計算モデルが発見できれば、そのほうが筆者の美的感覚には合っている。

◇ 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 (1975).
- [Aho 86] Aho, A., Ravi, S. and Ullman, J.: *Compilers, Principle, Techniques, and Tools*, Addison Wesley (1986).
- [Briscoe 93] Briscoe, T. and Carroll, J.: Generalized Probabilistic LR Parsing of Natural Language (Corpora) with Unification-Based Grammars, *Computational Linguistics*, Vol. 19, No. 1, pp. 25-59 (1993).
- [Chapman 87] Chapman, N. P.: *LR Parsing—Theory and Practice*, Cambridge University Press (1987).
- [Fujisaki 91] Fujisaki, T. and Jelinek, F., et al.: A Probabilistic Method for Sentence Disambiguation, *Proc. 1st Int. Workshop on Parsing Technologies*, pp. 105-114, CMU, Pittsburgh (1989).
- [Lavie 93] Lavie, A. and Tomita, M.: An Efficient Noise-skipping Parsing Algorithm for Context Free Grammars, *Int. Workshop on Parsing Technologies*, pp. 123-134 (1993).
- [Li 94] Li, H., Takezawa, T., Singer, H., Hayashi, T. and Tanaka, H.: An efficient phoneme-context-dependent LR table and its applications in continuous speech recognition, *Proc. the Acoustic Society of Japan (ASJ) autumn meeting*, 3-8-9, pp. 125-126 (Nov. 1994).
- [Li 95] Li, H., Suresh, K. G. and Tanaka, H.: Incorporation of Connection Constraints into Generation Process of Allophone-Base LR Table, 情処学会第 50 回全大講演論文集 (1995).
- [センリ 96] センリ, 李 輝, 田中穂積: LR 表によるトライ辞書構造の共通辞圧縮に関する研究, 言語処理学会第 2 回年次大会発表論文集, pp. 153-156 (1996).
- [田中 89] 田中穂積: 自然言語解析の基礎, 産業図書 (1989).
- [Tanaka 94] Tanaka, H., Li, H. and Tokunaga, T.: Incorporation of Phoneme-Context-Dependence in LR Table through Constraint Propagation Method, *Proc. AAAI-94 Workshop on the Integration of Natural Language and Speech Processing*, pp. 15-22 (1994).
- [Tanaka 95a] Tanaka, H., Tokunaga, T. and Aizawa, M.: Integration of Morphological and Syntactic Analysis based on LR Parsing, *J. Natural Language Processing*, Vol. 2, No. 2, pp. 59-74 (1995).
- [田中 95b] 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).
- [Tomita 86] Tomita, M.: *Efficient Parsing for Natural Language*, Kluwer Academic Publishers (1986).
- [Tomita 91] Tomita, M., ed.: *Generalized LR Parsing Technologies*, Kluwer Academic Publishers (1991).
- [植木 95] 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 (1991).
- [Wright 90] Wright, J. H.: LR Parsing of Probabilistic Grammars with Input Uncertainty for Speech Recognition, *Computer Speech and Language*, Vol. 4, pp. 297-323 (1990).

— 著 者 紹 介 —



田中 穂積(正会員)

1964年東京工業大学工学部情報工学科卒業, 1966年同大学院理工学研究科修士課程修了, 同年, 電気試験所(現電子技術総合研究所)入所, 1980年東京工業大学助教授, 1983年同大学教授, 現在, 同大学院情報理工学研究科計算工学専攻教授, 工学博士, 人工知能,

自然言語処理に関する研究に従事, 情報処理学会, 電子情報通信学会, 日本認知科学会, 言語処理学会, 計量国語学会, Association for Computational Linguistics 各会員.