

英語の決定的構文解析の一手法
A Deterministic Parser of English

武倉広幸
(東京工業大学 理学部)

1. はじめに

現在の計算機による構文解析では、多くの場合後戻りを繰り返しながら処理が行なわれる。これに対し、人間が文章を読む際、途中で想定しなかった要素に出会い前に戻って読み直すということはまれである。これを可能にする人間の内的な機構を、単純な方法で計算機上に実現できれば、処理効率の向上が期待できる。ここでは、このような方法の候補としてひとつの構文解析手法を提案し、その有効性を、実際に計算機による実験を行なった結果によって示す。この方法は、自然言語(英語)の構文にのみ焦点をあて、この範囲内で人間の内的な文理解に使われているであろう方策を、できるだけ単純に模倣しようとするものである。

この研究は、次のような人間の英語の文章理解の過程に対する(筆者の主観的)観察による仮定に基づいている。すなわち、文の構文構造の認識はそのほとんどを意味的な情報を用いなくても行なうことができるというものである。従って、見方を変え、この研究は、意味に頼らず構文要素だけを考慮して、どこまで解析することができるかを明らかにする研究となっている。ただし、ここで決定される“構文構造”は、一般的に使われるものよりゆるいものである。一般的な意味での構造の決定には、構文的情報だけでは不十分であることは、例えば次の文に現われるような前置詞句の修飾先の決定の問題を考えれば明らかである。

I saw the man with the telescope.

しかし、修飾先の決定はできないにしても、“with the telescope”という単語列が前置詞句を形成しているということは意味的な情報なしに決定できる。このことに注目すれば、他の部分の解析に悪い影響をおよぼさない限り、最初の段階で決まらない部分については、その局所的な構造のみを決定しておいて、後の段階で意味的な要素を加えて他の構文要素との関係を決するという方式も可能であることがわかる。上で述べた、“ゆるい”構文構造の認識という意味は、この最初の段階だけを行なうことをいう。細かな点ではいくつか未決定の部分を残しているが、少なくとも主部・述部といった大きな構造の決定は行なわれた状態というわけである。従ってこの段階では、前置詞や形容詞節の修飾先、等位接続詞で結ばれている範囲などは、そのはっきりとした決定をおこなわず、上に述べた局所的な構造のみをとらえる。例えば、上の“with the telescope”という前置詞句は、仮に、直前の名詞句を修飾しているものとしておく。

先に述べた仮定をもとに、この最初の段階の処理(以下ではこの処理を単に構文解析と呼ぶことにする)をできるだけ人間に近いと思われる方式で行なおうとするのが、ここで述べる方法である。人間の観察にその基盤をおくということから当然導かれる性質として、この方法では、文を左から右に順に(大域的な)後戻りをせずに解析する。このように、文を後戻りせずに解析してその構文構造を決定

しようという試みは、Marcusの決定性パーサ[1]が初めてであろう。それ以来、このような試みがいくつか([2][3][4])行なわれているが、これらのパーサは、いずれも単語の構文的曖昧さ(いくつもの品詞を持つこと)を完全には扱っておらず、また、構文カテゴリ単位の先読みを許している。これに対して、ここで述べる方法では、単語の各品詞に優先度をいれてその構文的曖昧さに対する解決をはかり、また、単語単位の先読み(実現上は局所的な後戻り)のみを用いるものである。

計算機上での実現を考えると、できるかぎり単純な枠組みで全体をとらえることも重要なことである。ここでは、それぞれの観察に基づき採用する方策を実際に実現する方法、すなわち、解析の具体的メカニズム及びそれをコントロールする文法規則の記述法も示す。また、現在この方法に基づき、先に述べた“ゆるい”意味での構文の認識を行なうプログラム(以下では、これをパーサと呼ぶことにする)を作成して各方策の有効性をテストしており、この実験結果についても述べる。

本稿の以下の構成は次の通りである。次の章では、この解析手法の基礎となっている人間の言語理解のモデルを、特にその機構の計算機上での実現法に力点を置いて述べる。それにつづいて、3章では構文解析の細かいメカニズム及びそれをコントロールする文法規則について述べる。4章では、作成したパーサによる実験の結果を見る。5章では、他の関連のある研究とここで提案する方法との比較をする。最後の章では、結論及び今後の展望について触れる。

2. 人間の言語理解過程のモデル

この章では、人間の言語理解過程を分析し、その分析結果を計算機による文の認識に生かす方法を検討する。ここで見るもののいくつかは、すべての自然言語に共通と思われるが、以後は対象を英語に限る。また、先に述べたように言語の構文的側面についてのみ議論する。以下の各節で4つの性質をあげるが、それらは互いに深い関係を持っている。

2. 1. 先読み

人間が内的に行なっている決定の遅延を反映させるために、先読みを導入する必要がある。この先読みは、人間の理解過程を考えると、単語をその単位に限る方が、[1]や[2]で用いている構文カテゴリ単位の先読みを用いるよりも適切である。また、計算機による処理も前者の方が単純な機構で済み好ましい。

我々が文を理解する際、その途中で、それまで構成したすべての構文構造の子細をはっきりと記憶してはおらず、構成しようとしている要素の方を強く意識してと思われる。(これは、[5]にも述べられている。)一旦、次にくるであろう構文要素を(意識してか無意識的にかは別として)予測してしまうと、我々の関心はその新しい予測を満たすことに移行してしまう。この機構によって、かなり長い文でも一気に理解が可能になるのである。後で述べる等位接

統詞を処理するような場合を除いては、現在構築中の構文要素を完成するために、振り返って前に作り上げた要素を意識することは少ない。構文要素単位の先読みを許すことは、この振り返って前の要素を見ることを許すことに対応する。この枠組みにおいては例えば次のようなルールの記述が許されてしまう。

IF 次の要素が他動詞で、その次の要素が名詞句ならば、
THEN それらを動詞句の子ノードとせよ。

あるいは、もっと極端に、次のような記述さえ可能となる。

IF 次の要素が名詞句で、その次の要素が動詞句ならば、
THEN その名詞句と動詞句を文Sの子とせよ。

このような枠組みは、人間の内的モデルとしてはあまりに強力過ぎ、また、単純なパーサを実現する基盤としても適当ではない。ここでは、[1] や [2] で構文単位の先読みが用いているものの役目を、以下に述べる方策、特に適切でかつより一般的な予測と、限られた長さの語単位の先読みによって果たせると考える。

作成したパーサでは、現在、この先読みの長さを2語にして、つまり、現在の処理の中心になっている語とその次の語を見ることを許す形で、実験を行なっている。なお、パーサの実際の動作は、処理の単純さと効率を考慮して、この長さの範囲で後戻り (backtracking) を許す形で実現されている。つまり、現在は、文のn番目の単語の処理が始まると、n-2番目の語の処理で決定した構造はけして取り消されない。処理がここへ戻った場合は、解析に失敗したものとみなしている。

2. 2. 適切な予測

提案する方策のうち最も重要な要素が、本節で述べる適切な予測である。我々が英語の文章を読んだり聞いたりする際、常に、次に何がくるか、あるいは何がこないのかを陰に陽に予測をしながら進んでいく。この事実を考慮せずに後戻りをしない構文の認識は不可能である。

そこで問題になるのは、この予測をどのように計算機による処理に反映させるかである。通常のトップダウン解析もこの予測を使っていると見ることができる。たとえば、次のような句構造文法規則のその右辺を予測だと考えることができる。

A-1 VP --> vt NP
A-2 VP --> vi
A-3 VP --> VP PP

A-1 や A-2 は、現在動詞句を処理しようとしている段階であり、次には他動詞がきて、続いてその目的語がくるとか、自動詞がきてその動詞句が終わるとか、予測しているものとみることができる。A-3 の場合は、動詞句の先頭でその動詞句がさらに動詞句と前置詞句からなっていることを予測することに相当する。A-1 や A-2 の場合は、右辺の最初が品詞となっているので、これを先読みして手がかりにすれば、A-1 と A-2 のどちらかをうまく選択できる。しかし、A-3 のようなルールはとても人間の予測を素直に表現して

いるとはいえない。無制限に先を見ることを許さないかぎり、前置詞句の存在まで予測することはできない。通常使われる句構造文法規則は、本文を“生成”するものであるもので、そのままの形で文の認識に用いることは適当とは考えられない。

そこで、A-1 や A-2 のルールのように、いつもその右辺の先頭要素を品詞とするようルールを表現することが考えられる。予測を反映させるという面からは、これは大きな意味をもっている。人間は出会った単語に影響を受けて次々と予測をしていく。この点に注目して英語の構文を解析したものとして久野らのpredictive analyser[6]がある。このシステムで使われる文法規則は、上のA-1, A-2 のルールのように、右辺の最初の要素が品詞であることが要求されるGreibach標準形で書かれている。この形式によると、右辺の2番目以降に現われる要素を予測とみなすことができる[7] が、ルールをこの記述方法に直すだけでは、人間の行なう予測を反映するのに十分ではない。下にあげる例文とGreibach標準形で書かれた規則B-1, C-1, C-2 によりこの点を検討してみよう。これらのルールは、文の生成という観点から見ると十分なものである。これらを生成の途中で使うことにより、(1) から(3) の文を生成することは容易にできる。なお、ルール中の記号NP, art, NP#, PA, ACは、それぞれ、名詞句、冠詞、冠詞に続く名詞句の構成要素、分詞、形容詞節を表わす。

- (1) There lived a girl.
- (2) There lived a girl
named Kaguya-hime.
- (3) There lived a girl
who loved her parents very much.

B-1 NP --> art NP#
C-1 NP --> art NP# PA
C-2 NP --> art NP# AC

人間の文の理解の過程とそれぞれのルールを比較してみる。我々が文を読んでいる途中で、名詞句が次に来ることを予測しているときに冠詞が登場したとする。この場合、B-1 のように名詞句の主名詞(head noun) がまだでてきていないと予測することは自然なことである。しかしこの時点で、この名詞句の後ろに分詞や形容詞節が現われ修飾するといったことを予測することはない。これらの後置修飾語句の存在をはっきりと認識するのは、主名詞に到達した後で修飾する要素があることに気がついてからである。したがって、C-1 及びC-2 は、人間のする予測を忠実に表現したものとは見ることはできない。また、計算機でこれらのルールを使って後戻りをせずに解析を行なおうとすると、無制限の長さの先読みを必要としてしまう。

ここで提案する方法は、このGreibach標準形によるルールの記述を自然に拡張することにより、人間の行なう予測をより忠実に反映するものである。また、ここで施す拡張は、句構造文法ルール、そのうちでも特にGreibach標準形のものに対して用いることのできる、単純なスタックを使う解析方法[7] を不可能にするものではない。

拡張の第1点は、ルール中に現われる構文要素を必須なものとしておかないもののふたつに分けることである。これらを必須予測、および任意予測と呼ぶことにする。先のルールの代わりに、次のD-1 からD-4 のルールを使えば、人

間のする予測をより忠実に表現できる。なお、ここにあげる
 ルールの表現は、焦点を任意予測にあわせるため、実際に
 用いられているものとやや異なっている。実際のルールの
 形式については次の章で詳しく述べる。

- D-1 NP --> art NP#
- D-2 NP# --> noun *NP-N
- D-3 NP-N --> (vt ing) NP
- D-4 NP-N --> rel-pro S#

D-2 の '*NP-N' のように、'*' がついているのは任意予測
 である。NP-N という予測は、名詞句の中心となる主名詞が
 既に現われており、ここで名詞句が終わり得るという状態
 を表わしている。当然、この後に名詞を修飾する分詞や関
 係節が続くこともあり、このような場合はルールD-3 ある
 いはD-4 で処理される。D-3 の右辺の先頭に現われる(vt
 ing)は、他動詞のing 形を表わす。後置修飾する分詞は、
 NP-N という予測を満たそうとして、この語(品詞)に出会
 うことにより、D-3 のルールが適用され処理される。D-4
 のルールに現われるrel-pro は、関係代名詞を表わす。関
 係節は、このルールによって分詞と同じように処理される。
 S#は関係代名詞に続く名詞句の欠けた節を表わす。このよ
 うな語句がなく名詞句が終了する場合、NP-Nは何もせずに
 消滅する。この節の冒頭であげた動詞句の処理も、同様に
 動詞句の後に続き得る要素を任意予測によってとらえるこ
 とで解決される。

拡張の第2点は、構文単位の集合を一度に予測すること
 を許すことである。これを束予測と呼ぶことにする。この
 拡張によって、動詞の後に続く要素の予測や、and などの
 等位接続詞に続く要素に対して行なうであろう予測を素直
 に表現できる。ここでは、この束予測について、be動詞に
 続く要素を例にして説明する。後で触れる特別な語の処理
 のところでは、等位接続詞の扱いにおいてこれが有効であ
 ることが示される。

動詞句の先頭のbe動詞に続くものとしては、補語となる
 形容詞や名詞句、beとともに受動態を表わす他動詞の
 過去分詞形、進行形を表わす動詞の現在分詞形などが出現
 する。be動詞に出会ったとき、人間はこれらの可能性のう
 ち、その特定の場面に現われる要素を前もって予測できは
 しない。しかし、つぎの例の“What we call”のように、be
 動詞の後に理論上無制限の長さをもつような挿入が現われ
 得ることから、この段階で何らかの予測をしていることは
 明らかである。

He is what we call a walking dictionary.

be動詞を見て何の予測もせず、次に続く語と一緒に見て初
 めて構文の種類を予測すると仮定すると、この文がごく普
 通に理解できることが説明できない。おそらく、be動詞の
 出現した段階で、次にくることができよう要素を、それ
 らひとつひとつを強く意識はしないでおおまかに予測を
 しているのであろう。

このような予測をそのまま表現する手段として束予測を
 導入した。可能性のある予測の集合を新たな予測として独
 立に扱うこともできるが、動詞の後や、等位接続詞の後に
 出現し得るものの多様性を考えるとき、人間の行なう処理
 との対応はつきにくい。また、文法の記述としても、使わ
 れる記号やルールの増大を招き好ましくない。束予測を導

入すれば、可能性のある構文要素をひとまとまりにして予
 測することが、新たな記号を導入することなしに、しかも
 ひとつのルールでまとまって扱われる形でできる。例えば
 be動詞に続く要素をとらえるために、beが現われた時点で
 これに続く構文要素を細かく選択してしまわずに、後続語
 の情報を利用して構文構造の決定が行なえる。上の例でも、
 束予測の処理を一旦棚上げして、まず挿入された節の処理
 をしておいてから、先に出しておいた束予測を利用して残
 る名詞句を処理できる。beと補語の間にどんなに長い挿入
 がはいっても正しく捉えることができるわけである。

2. 3. 優先度(preference)

この節では、ルール間、および各単語の属する品詞間の
 相対的な優先度(好み; preference)について論じる。こ
 れらは、概念そのものが曖昧な為、今まであまり取り上げ
 られていない。自然言語とプログラミング言語などの人工
 言語との大きな違いのひとつに、ある単語の属する品詞の
 多様性があげられる。これが、時にはひとつの文に対して
 何十、何百という構文的曖昧さを生み出すこと的主要原因
 となっている。しかし、ある語の属する品詞の間にも、
 使われ易いものと使われにくいものがあることは確かであ
 る。また、一般には、ある時点で適用できるルール(ある
 時点で想定できる構造)は複数あるが、これらの間にも使
 われやすさの順序がある。プログラミング言語では、この
 いわゆる“conflict”は、通常一語の先読みによって解決さ
 れる[8]。また、Pascalなどのプログラミング言語では、
 “dangling else”[8]のような曖昧性が存在しこれは先読み
 を用いても解決できない。しかし、この曖昧性は天下り的
 に解釈規則を定めることにより解消される。自然言語に対
 しては、特に後者の方法によってこの問題を解決することは
 期待できない。

まず初めに、[9] から引用した次の文を細かく検討して
 この優先度の問題についての出発点とする。

We do not utilize outside art services directly.

この文章の解析をすると、[9] に述べられているように、
 “outside art services”の部分の次の3通りに解釈するこ
 とができる。

- (1) 名詞・名詞・名詞の列ととらえ、これが他動詞
 “utilize”の目的語となっていると見るもの。
- (2) 前置詞・名詞・名詞の列ととらえ、最後の名詞
 “services”が他動詞 “utilize”の目的語、“out-
 side art”は目的語の前に挿入された前置詞句
 と見るもの。
- (3) 名詞・名詞・動詞の列ととらえ、“outside”が
 “utilize”の目的語、“art services”が、“out-
 side”を修飾する形容詞節(関係代名詞省略)と
 見るもの。

人間がこの文章を読んだ時、最初に(2)や(3)の解釈をす
 るということは考えられない。この事実は、次のような仮
 定をすることによって、構文的な要素を考慮するだけで説
 明することができる。

- (1) “outside”という単語は、前置詞としてよりも名
 詞として使われることが多い。

- (2) "services"という単語は、動詞よりも名詞として使われることが多い。
- (3) 名詞句を後置修飾する形容詞節は、通常その始まりを示すなんらかの手掛かり、例えば、関係代名詞・冠詞などといったもの、を伴って出現する。

これらの仮定のうち、(1)と(2)は多品詞語の品詞間の優先度に対するものであり、(3)は、ある時点で適用されるルール間の優先度の問題である。[1]と[5]から引用した次の例は、(3)の仮定を間接的に支持するものである。

The cotton clothing made of grows in Mississippi.
The chocolate cakes are coated with tastes sweet.

これらは、いわゆる迷い道文 (garden path sentence) であり、通常の文と異なり、我々はその解釈の途中で後戻りをして文の構造を再構築させられる。上の(3)を仮定すれば、この事実を次のように説明することができる。通常あるはずの形容詞節の始まりを示す単語がないため、名詞の連続を前者が後者を修飾するものとして捉えてしまうからである。

(1)(2)のような同一語の品詞の優先度の存在の仮定は、構文的に曖昧な文として古典的に有名な次の文[6]を解析する時にも役立つ。

Time flies like an arrow.

この文に対しても、"time"、"flies"、"like"の3単語の品詞の曖昧さによって、少なくとも次の3種類の構文的解釈が考えられる。それらは3単語をそれぞれ、

- (1) 名詞・動詞・前置詞 (意味的に妥当な解釈)
(2) 名詞・名詞・動詞
(3) 動詞・名詞・前置詞

としてとらえるものである。しかし、次のような品詞間の優先度を考慮すれば、この文から(1)のような解釈を後戻りせずに得ることができる。

- (A) time 名詞>動詞
(B) flies 動詞>名詞
(C) like 動詞>前置詞>形容詞=接続詞

このような順序を考慮して、左から右に解析を進めて行けば、上記(1)のような解釈は容易に得ることができる。この過程はまた、先に述べたルール間の優先度、(この場合だと、普通の文章で命令形が使われることはまれであるといった形のもの)との関連を持っている。

人間の理解の過程を取り入れようとすると、この優先度 (preference)の問題は重要な要素のひとつとして扱うべきものである。この知識を導入すること、特に、各単語ごとに品詞間の優先度を考慮することは、膨大なデータを必要とすることから容易なものではない。しかし、地道な集積をする価値の十分にあるものと考えられる。現在、一応の優先度が付いている辞書が使用可能であり、これを使用して実験を行なっている。

2. 4. 特別な語の処理

英語においては、他の語と異なった種類の扱い方を要求する単語がある。これらをそのままプッシュダウンスタックの枠組みの中で扱おうとすると無理が生じる。現在は、'and'、'or'、','といった等位接続詞や特殊文字、及び、文の先頭に現われたり途中に挿入されたりする副詞、前置詞句、副詞節などを導く単語を特別処理の対象としている。これらの語に共通な点は次の2点である。

- (1) 通常陽に予測されない。
(2) 文のほとんどすべての場所に現われうる。

等位接続詞の処理は、[10]や[11]でのものとほぼ同じであるが、このためのスタックを用意し、束予測を用いることによって単純に処理できる。例として、"and"の現われうる位置を示す次の3個の句構造文法規則、及び、これらの規則で生成される次の3文について考える。

- (1) NP → NP and NP
(2) VP → VP and VP
(3) SE → SE and SE
- (1) Mary loves the dog and
the cat very much.
(2) Mary loves the dog and
washes him every day.
(3) Mary loves the dog and
she is always with him.

これらのルールは、先に任意予測を導入したときに述べた理由とまったく同様に、直接的にここで行なう文の解析には使えない。例えば、ある文の先頭で、その文が(3)のルールで示されているような等位な2つの節からなっているということは、無制限の先読みなしには予測不可能である。また、and それ自体及び既になされている予測を見ただけでは、次にくる構文単位の予測ができないという事実から、前に述べたような拡張だけではこの処理ができないことも明らかである。and の後に続く構文要素は、この前で完成された構文要素であり、上の3つの文では、それぞれ、名詞句(NP)・動詞句(VP)・節(S)がand の直前で完成されており、それらが再び現われている。これは、人間がただ予測すること以外に、直前で満たされた予測を記憶していることを示している。

この機能は、予測を積んでおくスタック (予測スタック) のほかに、もうひとつのスタック (and スタック) を、この記憶を保持するものに用いることで簡単に真似ることが

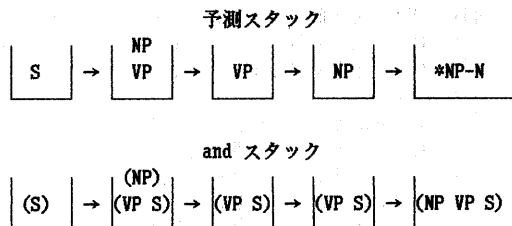


図1 予測スタックとand スタックの変化の関係

できる。人間もスタックのようなものを使っているかどうかかわからないが、機能的には似たものであると考える。この使用方法を上例で具体的に説明する。それぞれの文の処理は、当然、and まではまったく同じに行なわれる。ここにいたる処理の過程で、図1の上に示すように、S・VP・NP（主語ではなく、動詞の目的語の名詞句）の順に予測され、and の直前では、先に述べた名詞句の後ろに続き得る要素をとらえるための任意予測、“NP-N”が出される。

（ほかの予測も出されるが、ここでは省略する。）and スタックはこれと並行して図の下のように状態を変化させる。すなわち、and スタックは予測スタックと常に同じ深さをもっており、その各要素はそれと同じ位置にある予測スタックの要素が満たされることによって満たされる、それ以前に出された予測（その予測自身も含む）のリストとなっている。

このスタックによって、例えばこの場合のand の処理は、それが出現した段階でand スタックの要素のうち、予測スタックのNP-Nに対応する位置にあるものによって保持されている予測のリストを、束予測として予測スタックに積んでやることによって終了する。（なお、ここで、任意予測が、もうひとつの別の役目を果たしている。）後は、次に現われる単語によってその束の中から適当な予測を選択すればよい。

副詞や副詞節、文頭の前置詞句などは、通常必須な要素ではなくその出現は予測不可能であり、これらも特別な処理をする必要がある。前述の任意予測の要素として扱う方法も考えられるが、例えば、次のような副詞の出現ひとつひとつを陽に予測してルールで記述するのは、ルール的大幅な増大を招き好ましくない。

- (1) We do not utilize outside art services 'directly'.
- (2) We do not 'directly' utilize outside art services.
- (3) We do not utilize 'directly' outside art services.
- (4) There are substantial economic risks and 'generally' a lack of available data.
- (5) 'Generally' there are substantial economic risks and a lack of available data.

最初の3つの文は、副詞“directly”の位置が違うだけであり、いずれも許容される文である。(4)の“generally”の位置は、やや特殊な感じを受けるが、普通の文章に現われる形である。(5)のように、“generally”は文の先頭に現われることもできる。これらのすべての場合をルールで記述すると、解析のほとんどすべてのフェーズに副詞を陽に予測するようにルールを作らねばならない。これらは、まったく別の機構で扱うべきである。このような予測されない構文要素は、その先頭の語の登場によってその出現を感知し、（必要ならば）残りの部分に対する予測を、それ以前の予測で残っているものの上に積むことによって処理できる。上の副詞の例についてみると、(1)は、前に触れた動詞の目的語の後に出現する任意予測によりとらえられ、それ以外は、予測されていない副詞の出現により、後述する文法規則によって挿入処理の機構が働き、副詞の存在を示すだけでスタックの状態を変えずに処理を続行する。このほか、例えば、文頭に現われる前置詞句や副詞節などの

処理は、その始まりを示す確かな印となる前置詞や接続詞によって、前置詞に続く名詞句の予測や接続詞に続く節の予測を、初期状態で出されている節(S)の予測の上に積むことによって処理される。

3. 方策の実現とパーサの動作

本章では、前章で述べた方策をどのように文の認識に反映しているかをより細かく述べる。まず、パーサの入力となる、辞書引きされた後の各語に付く品詞の構造を示す。これはルールの記述にも関連する。続いて、パーサの基本的構成要素であるスタックの働きと、その動作を指定する文法ルールの記述方法について述べる。最後に、具体例により文法ルールとパーサの動作の関係をより細かく説明する。

3. 1. パーサへの入力データの形式

前節で述べたように、このパーサでは各単語が属するさまざまな品詞の間の使われやすさの違いを重要な要素として見ている。従って、単語ごとにそれが属する品詞が優先度付きで入力される必要がある。現在、この目的のために使える辞書があり、この辞書によって品詞付けを行なった結果を次のような形式のリストに変換してパーサの入力としている。

1. (study (vt1 fnt inf) noun (vil fnt inf))
2. (studying (vt1 ing))
3. (is (be fnt))
4. (important adj)

各語ごとのリストの先頭要素がその語形であり、続いてひとつ以上の、その語の属する品詞を表す記号のリスト（要素が1個の場合はかっこを省略してよい。）が、その優先度の順に並んでいる。各品詞のリストの先頭の要素を本体と呼ぶ。2番目以降の要素は、本質的には動詞の活用形を表すためのパラメータとして用いる。パラメータとして余分な要素がはいっていてもパーサの動作にはまったく影響がないので、実験ではここに品詞間の優先度を表わす数字を加え、ルールの選択に使っている。

品詞の分け方は普通の辞書などで用いられているものよりやや細かいものであり、現在約40の種類がある。特に、動詞は後続する要素のパターンによって7の種類に分け、それぞれを別の品詞として扱う。動詞の変化形は同じ品詞として扱うが、構文的な働きは大きく異なるので、それぞれを区別するためにパラメータを使う。原形(infinitive: inf)・定形(finite: fnt)・過去分詞形(past participle: ptp)・現在分詞形あるいは動名詞形(ing)の4種類である。例えば、2.の(vt1 ing)は直接目的語を取る他動詞 vt1の現在分詞あるいは動名詞形を表わす。なお、4.に現われる記号adjは形容詞を表わしている。

3. 2. パーサの構造と文法ルール

このパーサでは、3本のプッシュダウンスタックを用いて文の構造の認識を行なう。このスタックの状態の変化を記述するのが文法ルールであり、前章で述べた方策は、等位接続詞の処理と辞書の記述で表わされる品詞間の優先度を除いて、すべて、ルールに明示的に記述されるスタックの状態の変化により実現され、実際の処理に使われる。それぞれのスタックの役割及び名称は次のとおりである。

1. 予測の記憶に用いる予測スタック。
2. 等位接続詞の処理に用いる and スタック。
3. 関係節などの処理に用いる NP スタック。

このうちもっとも重要なのが予測スタックであり、ほとんどの文法ルールはこのスタックの変化のみを記述する。2. の and スタックは、前章で見たように、1. の変化に依存して状態を変える。3. の NP スタックは関係代名詞など、その構成要素のうち名詞句の欠けるものの処理を簡潔にするために導入したもの[12]である。様々なシステム[13 など]で類似した方法が使われているので、前章では特に触れなかった。これらのスタックの状態は、処理の進行とともに、ルールの記述に従って下に示すように変化する。

各スタック要素の最小の構成単位であり、また、ルールの重要な要素ともなっているのが予測記号である。それぞれは、ひとつの構文単位に対応しており、例えば、記号 S は節(文)に、NP は名詞句に対応する。品詞を表わす記号は予測記号としても使われる。従って、各予測記号にもパラメータにより細かな制限を加えることができる。これにより、たとえば、動詞句を表す予測記号は VP ひとつだけで済む。(VP inf) により原形動詞で始まる動詞句、(VP ptp trn) により他動詞の過去分詞形で始まる動詞句(目的語が取れているもの)といったように表わせる。現在、品詞を除いて約 15 の予測記号を使っている。品詞が予測記号として使われる場合を含めて、各予測記号の前に '*'、'e' あるいは '+' を付けることができる。*'、'e' は任意予測を表わし、'e' が 2.4. で述べた等位接続詞に続く要素を捉えるのに使われないことを除いて同じ働きをする。 '+' は NP スタックと関連して使われる。

文法ルールの記述は、前章で述べた方策を自然に表現できるように形となっている。各ルールは、下に示すように、品詞・現在の予測・語を進めるかどうかのフラッグ(シフトフラッグ)・新予測の 4 つの部分からなっている。

品詞	現在の予測	フラッグ	新予測
(1) aux	VP	t	(VP inf)
(2) nou	NP	t	*NP-N
(3) rel-pro	NP-N	nil	+ADJ-CL
(4) ?	S	nil	NP (VP fnt)
(5) pre	?	nil	PP @cma
(6) be	VP	t	
[bunch (ADJ) ((VP ing)) ((VP ptp trn)) (NP) (PP)]			

*VP-CMP

[注] すぐ上の 3 行が (6) の新予測である。

各文は、一語ずつ左から右に順にルールの適用の繰り返しにより処理される。品詞及び現在の予測の部分は、そのルールの適用の条件を表す。ある時点で適用可能なルールは一般には複数あるが、その中からあるルールが選択され適用されると、原則としてパーサは、予測スタックの最上位の要素を取り出して、そのルールの新予測を(それがあれば)スタックに積み、また、フラッグ部分が 't' となっている場合には、処理対象の語を指すポインタを次の語に進め以後の処理に備える。

あるルールの集合からひとつのルールを取ったとき、そのルールの適用可能性は、品詞及びスタックの状態を与えることによって、以下の 4 つを考慮して決定される。

- A. 与えられた品詞。
- B. 現在の予測、即ち、予測スタックの最上位要素。
- C. ルールの品詞部分。
- D. ルールの現在の予測部分。

ルールの品詞部分には、品詞または記号 '?' を、現在の予測部分には、予測記号または '?' を書く。あるルールについて、次の条件がすべて成り立つとき、そのルールが適用可能になる。

- (1) A が C の制約を満たしていること。
- (2) D が ? であるか B の本体が D と一致すること。
- (3) A が B の制約を満たしていること。

ここで、(1) 及び (3) の、X が Y の制約を満たすとは、Y が記号 ? であるか、あるいは、両方の本体(リストの先頭要素)が一致し、Y のパラメータ部分に現われる要素すべてを、X もパラメータとしてもっていることを意味する。(3) の条件は、例えば、現在の予測が (VP ing) のときに (vt1 fnt) の品詞がきても適用可能にしないようにする。これにより、原形・定型・ing 形のどれで始まる動詞句も 1 個のルールで処理することができる。

(5) のルールのように予測部分が ? のルールは、前章で述べた挿入処理を起動する。他に適用可能なルールがなく現在の単語が前置詞である場合このルールが適用され、前置詞句の予測を、すでにスタックにある予測の上に積み、その位置から始まる前置詞句をとらえる。このルールの新予測にある '@cma' は、文頭の前置詞の後に書かれることのあるコンマを処理するもので、品詞が予測記号として使われた任意予測である。

品詞部分が ? のルールによってルール数の増大を防ぐ。

(4) のルールにより、節の予測 S が予測スタックの最上位にあると無条件にこのルールが適用可能になり、その予測を、名詞句と動詞句の予測に置き換えてしまう。これにより、次のようなルールをいくつも用意する必要がない。

art	S	nil	NP (VP fnt)
adj	S	nil	NP (VP fnt)
nou	S	nil	NP (VP fnt)
(vt1 ing)	S	nil	NP (VP fnt)

ルールのシフトフラッグの部分は、現在の処理の対象となっている語位置を次に進めるかどうかを示す。't' というフラッグは、このルールの適用によってこの語についての処理が終了したことを示し、語のポインタを次に進めることを意味する。'nil' は、ポインタを進めない。このフラッグは、ルールの増大を防ぎ、また、今後加える予定の構文構造の明示的構築の為に存在する。

ルールの 4 番目の新予測の部分は、0 個以上の予測の列からなる。予測には、予測記号ひとつからなるもの、予測記号の集合からなる束予測、新たな予測スタックを特別につくるための処理をパーサに要請するものの 3 種類がある。束予測は、(6) のルールのように書かれる。この目的は前の章で述べた。等位接続詞に対するスタックの変更の処理は、第 3 の種類の予測によって起動される特別のプログラムによって行なわれる。この処理の詳細も前の章で述べた。

この新予測のひとつとして (3) のルールのように '+' がついた予測が現われたときは、NP スタックに名詞句の予測

(NP) をひとつ積み、後で名詞句が必要な場面でこれが作り出せないときに、ここから要素をポップアップして処理を続行する。人間が目的語の欠けた節・動詞句といったものを独立に扱っているとは考えられないので、このような処理は妥当なものであろう。

3. 3. 解析例

実験に用いている文章[14]からとった次の例文によって、これまで個別に述べてきたルールの記述とパーサの動作の関係を整理する。

- (1) As the example shows, STYLE output is in five parts.
- (2) First, we must throw away the printer's marks and formatting commands that litter the text in computer form.

図2に(1)の文のパーサの出力を示す。n という数字の横にあるのが先頭からn番目の位置にある単語の語形とその単語をもつ品詞の並びである。各品詞のリストの最後の要素はその品詞の優先度を表わす数字であり、小さい数字が優先度が高いことを示している。先に述べたように、これらの数字はルールの適用の可能性には影響を与えないが、適用可能ないくつかのルールから選択をするときのひとつの基準として用いている。品詞列の下に行から次の単語の間までが、その単語を処理するのに適用されたルールの列である。単語ごとのルール列の最後には、必ずシフトフラッグが't'のルールがある。

図3に予測スタックの変化の最初の部分を示す。この文では、先頭の'As'により予測されていない副詞節の始まりを感知し、予測部分が?のルールが適用され、これに続く節(S)とその終わりを示すコンマ(cma)をスタックにプッシュしている。コンマが任意予測になっているのは、文頭だけでなく文末の副詞節もこのルールによってとらえるためである。これらの予測は5語めまで満たされ、6語め以降でその下のS, prd, eos が満たされる。

(2)の例文でも、(1)の副詞節と同様に、最初のふたつの単語("First"と",")及び"away"が予測部分が?のルールによって、スタックの要素をポップすることなく処理される。この文の解析が"that"の所まで進むと、これを見て関係代名詞のルールが適用され、このときにNPスタックに名詞句(NP)がプッシュされる。このプッシュされたNPは、次の"litter"を処理しようとするときにポップされ、これによって"litter"は動詞として認識され処理が続けられる。

4. 実験

いくつかの文章をパーサに入力して、ここに提案した方法の有効性をテストしている。本章で、このうち比較的详细な検討を行なったものの結果を示す。本稿にあげた手法で構文解析をするとき、ある場面で適用可能ないくつかのルールの中からどれを選ぶかは非常に重要な問題である。まずこれについて現在採用している方法を示し、続いて、実験の結果とその分析により明らかになったこの構文解析手法の問題点をあげる。

4. 1. 適用するルールの選択

このパーサでは、単語単位の先読み(局所的な後戻り)しか許さないで、ひとつのルールの選択の誤りが致命的

```

*** sentence no. 29      12 words
(As the example shows , STYLE output is in five
 parts . eos)
1. (as (co2 4) (co3 4) (pre 4) (adv 122))
   ((co2 4) nil t S @cma)
2. (the (art 4))
   (? S nil NP (VP fnt))
   ((art 4) NP t NP-ADJ)
3. (example (noun 6))
   ((noun 6) NP-ADJ t *NP-N)
4. (shows (noun 6) (vt6 fnt trn 6) (vil fnt 12)
   (vt1 fnt trn 12))
   ((vil fnt 12) (VP fnt) t *VP-CMP)
5. (, (cma 6))
   ((cma 6) @cma t)
6. (STYLE (noun 4) (vt1 fnt inf trn 10)
   (vil fnt inf 142))
   (? S nil NP (VP fnt))
   ((noun 4) NP t *NP-N)
7. (output (noun 6))
   ((noun 6) *NP-N t *NP-N)
8. (is (be fnt 6))
   ((be fnt 6) (VP fnt) t @not
   [bunch ((VP ptp trn)) (ADJ *VP-CMP)
   ((VP ing)) (NP *VP-CMP)
   (TO-INF) (PP)1])
9. (in (pre 4) (adv 6))
   ((pre 4) PP t NP)
10. (five (noun 6))
   ((noun 6) NP t *NP-N)
11. (parts (noun 2) (vt1 fnt trn 4) (vil fnt 6))
   ((noun 2) *NP-N t *NP-N)
12. (. (prd 6))
   ((prd 6) prd t)
13. (eos (eos 6))
   ((eos 6) eos t)
*** success *** 12 words 102 /60 sec.

```

図2 解析例

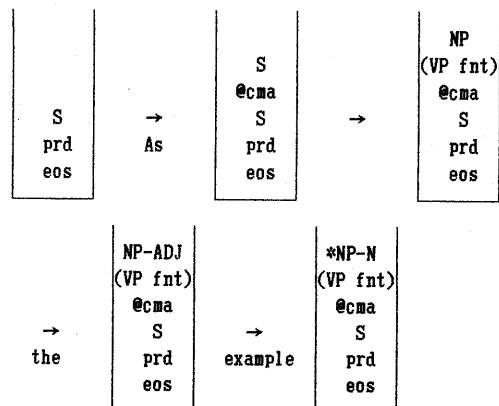


図3 予測スタックの変化

となる。許される先読みからできるだけ多くの情報を取り出して行なうべきだが、現在は、次のような方法によって次の語の情報を生かしている。

ルールの集合及び各単語の属する品詞には、いずれも、記述されている順番によって順序がついている。さらに、このように順序付けられているルールおよび品詞を、はいつている順序の間に境界線を引くことによって、優先度の高いものと低いもののA・Bふたつのグループに分ける。ルールでは、Bグループはほとんどが予測が?のルールであり、品詞の方では、普通にはあまり使われない品詞がBグループにはいる。これらを使って次のようにルールが選択され処理が進んでいく。なお、ここで用いている先読みによるチェックは、実用上は先読みの範囲内の局所的後戻りにより代用されている。

1. 現在の単語の属する品詞を優先度の高い順にひとつ選ぶ。
2. 選ばれた品詞及び現在の予測の条件を満たすルールを探す。
 - 2-1. そのようなルールがある場合は、ひとつをルール間の優先度に従って選び固定する。(これをRとする。)
 - 2-2. 該当するルールがない場合は、
 - 2-2-1. 現在の予測が任意予測ならば、それをスタックからポップし2.の先頭からもう一度試みる。
 - 2-2-2. 現在の予測が必須ならば、1.に戻り次の品詞で試す。この際、2-2-1.によりポップされた任意予測があるときは、これを戻してから行なう。他に残っている品詞の選択がない場合は、失敗となり停止する。
3. 選ばれたルールRの妥当性をチェックする。
 - 3-1. 妥当なルールならば、スタックの状態をルールRの記述に従って変える。ここで、このルールのシフトフラグが"t"ならば1.に戻り、"nil"ならば2.に戻る。
 - 3-2. そうでない場合は2.に戻り次のルールで試す。

ただし、3.の妥当性を満たすルールRとは、次の1.の条件と、2.あるいは3.の条件を満たすものである。

1. このまま進んでいっても、先読みの範囲では行き止まりにならない。
 2. RがAに属するならば次の語までAに属するルールだけで処理が行なわれる。(Bに属するならばこれは無条件に満たされる。)
 3. RはAグループに属し、次の語の処理を終えるまでにBグループのルールを使う必要がある。しかし、まだ試されていないルールの中で、上の条件2.を満たすものはない。
- 2.あるいは、3.の条件を満たすという制約は次のような文の解析に役にたつ。

It conforms to the stylistic rules.

動詞"conform"は、他動詞としてもよく使われ、使用した

辞書では自動詞より高い優先度がつけられていた。2.の制約がないと、次の前置詞句を処理するため、予測されていない句を処理するルール(予測部分が?のルール)を適用してしまい、解析に失敗する。この場合、この単語を自動詞とすると2.の条件を満たしてしまい、これを他動詞と取るルールは選択されない。なお、次のような例があるので、動詞の後に前置詞がきたからといって自動詞とは限らない。

He did adapt for the use of linguists an existing system of formalization.

4. 2. 実験結果

実験した文章のうち、計算機関係のマニュアル[12]とある雑誌の記事それぞれ86文ずつについては、比較的詳しい検討を行なったのでここにその結果をあげる。実験の方法は、まず基本的なルールを作り、それにもとに文をパーサに入力し、その結果を前の方の半分ほどの文章について検討しルールを修正することを何回か繰り返した。従って、対処できる文型はこれらの記事に依存している。

結果的に解析に成功した文は、マニュアルが65文、雑誌の方が70文であった。本稿の冒頭で述べたように、ここでいう解析の成功は一般的なものに比べてゆるやかな意味で用いており、前置詞句の修飾先などはきちんと決定していない。失敗した例には、解析手法自体の問題のほかに、処理するルールがないことによるもの(マニュアルの方では5文)もある。いたずらにルールを増やすと他の文の処理に悪影響を与えたりするので、ただ加えるわけにはいかない。その他の失敗の原因はルールの優先度の問題、品詞の優先度とルールの優先度との関係によるものの数が多く、ルールの優先度をうまく調整することによって解決されるものも多いと思われる。現在の文法ルール数は約100であり、細かな処理を十分に行なっていないこと、イディオムをまったくいれてなく、これを分解した形で捉えていること(例えば、"a number of ..."は、名詞句+前置詞句と捉える。)などの点を考えると、本稿で提案した方法の有効性が示されたものと考えられる。

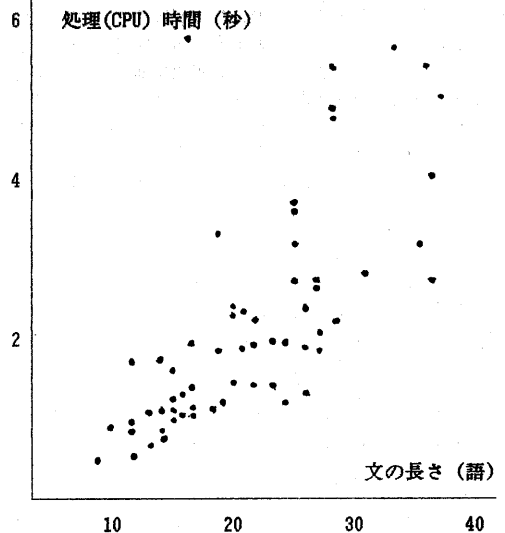


図4 処理時間と文の長さの関係

そのほか、解析時間と文の長さの関係及び先読みの長さと解析の成功・失敗との関係を調べてみた。図5にマニュアルの文のうち解析に成功したものの長さや時間の関係を示す。きちんとした検討はしていないが、両者はほぼ比例している。両方の文章について、先読みの長さを3語にして2語のときと結果の比較を行なった。長さを3語にしたことによって失敗が成功に変わったものは、両方の文章のうちともにわずかに1文ずつであった。一旦誤った方向に行ってしまうと、わずかな後戻り（先読み）だけでは、修正がきかないようである。

4. 3. 問題点とその改善

ここで解析に失敗した例を中心に、結果を見ながらこの解析手法の問題点を述べ改善の方法を検討する。

次のような文は、現在の枠組みではとらえることが不可能である。

Even using these broader definitions, simple sentences dominate many of the technical documents that have been tested.

分詞構文は、動詞句を完成して次の単語を見るまでは、動名詞を主語とする文と区別ができない。人間が動詞句を完成する前に、どちらの文型であるかを決定しているはずはないので、ここで採用している、文の最初の状態で節(S)を予測しているのは適当ではないと考えられる。おそらく、最初には何も予測しておらず、最初の単語によって呼び起こされる何らかの予測をきっかけにしてあるまじり（構文カテゴリ）を完成させ、これが本稿であげたルールの品詞部分のような役目を果たして、全体的な構文の予測を行なうのであろう。例えば、主語となる名詞句が完成され、次に（先読みとして）動詞がくるのを見て、初めて動詞句の出現を予測するのであろう。このような機構を表現する枠組みは非常に興味深いものであり、現在の方法をこのような方向に拡張することを検討している。

次の例のコンマで囲まれた挿入句は、途中にはいつている動詞句によって分離されている主語を説明しており、直前の情報を使うだけでは捉えることがむずかしい。ボトムアップな処理ができればこの部分が名詞句を構成していることは、簡単にわかるであろう。

Although many judgements of style are subjective, particularly those of word choice, there are some objective measures.

次のような単語列は、いずれも先頭の単語が次の語を修飾するものとして捉えられてしまった。

STYLE run on UNIX ...
sentences measure readability ...
STYLE reports on the number of ...

これらの解決方法としては、固有名詞や名詞の複数形が、後続する名詞を修飾することは少ないことを文法に取り入れることが考えられる。また、一般的に、動詞と名詞の両方の品詞を持つ単語が名詞の後にくるときは、動詞となる可能性が高いようである。こういったことを含めてより細かな処理をすれば、成功率はもっと向上すると思われるが、

システムとしてみた場合に、全体の見通しが悪くなることや、効率が低下することは避けられず、この方向での拡張は有効性をしっかり検討した上で行なうべきである。

5. 他の解析方法との比較

本章で、他の類似した特徴を持つ解析手法であるLR(k)パーサとMarcusによって設計された決定性パーサの2種類のパーサとの比較を行なう。

本論文で述べてきた方法は、先読みを単語単位に限ったことから、LR(k)パーサで同様の処理を記述できる。しかし、この枠組みによる実現は、必要になる文法ルール・記号の数を考えると実用的ではない。英語の各単語はいくつかの品詞に属するので、各品詞を終端記号として扱うことはできない。品詞の集合をひとつの終端記号としてみなす方法もありうるが、単語のもちうる品詞の多様性を考慮すると実用的ではない。この上、ここで述べた方法のように品詞間の優先度まで考慮に入れると、各単語一つ一つを終端記号とみなす場合と大差はないであろう。任意予測・束予測や特殊語の処理の機能も、人為的な記号やルールを増やすことにより、理論的にはこの枠組みで実現できる。しかし、例えば、各ルールに対して統一的に優先度をいれるなどは不可能であろう。

次に、Marcusにより提案され今もその拡張が行なわれている決定性パーサとの比較をする。ここで述べた手法とのもっとも本質的な違いは、その先読みの単位（語単位か構文要素単位か）にある。この違いが、両方のパーサのデータ構造に自然に現われている。Marcusのパーサでは、構文構造構築の為にスタックの他に、構文要素もはいることが許される先読みの為にキューが用いられている。ルールは、スタックの最上位要素とキューの最大3個までの要素によって条件を記述する、プロダクションルールの形となっている。それぞれのルールには優先度がついているが、これは、主に状態の変更に伴って別のルールを適用するために用いられており、ここでのものと使用目的が異なっている。ルールの使いやすさは考慮せず、先を見ることによって後戻りをしないで処理を行なっているといえよう。なお、彼のパーサでは、はっきりとした多品詞語の処理は行なっておらず問題として残されているが、ここではこの問題に対する処理をはっきりと打ちだしている。

ここで述べた方法では、陽に構文構造の構築を行なっていないので単純な比較はできないが、語単位の先読みしか行なわないという点によって、Marcusの方法に比べ単純な構造で実現している。ルールの形式はMarcusのパーサのルールの記述方法よりはるかに直観的で分かりやすく、変更も容易である。ここで述べた方法が予測に重きを置いているのに対して、Marcusの方法では、ボトムアップにある程度のまとまりを作っておいてから処理をする形式となっている。2.1.で触れたようにこの枠組みは強力であるが、人間の処理との対応はつきにくいように思われる。また、計算機による処理も、本稿のパーサの実現がプログラムに単純な局所の後戻りを組み込むだけで済むのに対して、かなり複雑になると思われる。

6. 結論及び展望

本稿では、人間の言語理解の過程のモデルに基づいた構文解析手法を提案した。この方法の主な特徴は、次の4点にまとめられる。

1. 限られた、語単位の先読みによって、人間が行なう構造の決定の遅延を反映する。
2. 任意予測・束予測により、人間の出す予測をできる限り忠実に反映する。
3. ルール及び各単語の属している品詞の間に使い易さの順序を考慮して優先度を入れ、人間がある場面である単語から受ける構文的な印象を反映させる。
4. 等位接続詞や、出現を予測できない副詞・前置詞句・副詞節等に対しては、単語をてがかりにそれらの出現を感知する別の機構を働かせて処理する。

また、ここでは、以上の方策を実際に計算機の処理に生かす枠組みとして、単純なプッシュダウンスタックを用いた実現法とその処理を記述する文法ルールの形式も提案した。以上の方策及び実現法は、実際にパーサを作成し多くの文章を解析することによってテストされており、それによりこの手法及び実現法の有効性も示された。

この方法は、その処理の対象を構文的な要素に限っているため、意味を用いずにどの程度まで正確に構文の把握を行なえるかという問いに対するひとつの答えを試みにもなっている。実験した範囲では、今の段階で8割程度の文の構造を捉えることに成功しており、意味を用いなくてもおおまかな構文構造の把握は可能であるという考え方を支持する結果が得られた。また、この方法が人間の処理の観察に基づくものであることから、実際に人間が内的にどのように文章を理解しているのかという問題との関連は非常に興味深いものである。

一方、人間が簡単に処理できるような構文でもこの枠組みの中では扱えないものもあり、この手法には本質的な部分で改善の余地がある。より深い人間の処理に対する観察に基づく新たな枠組みが必要であり、課題として残っている。

謝辞

指導教官である米澤明憲助教授には、この原稿に関して数多くの有益なコメントをいただきました。特に、この研究と意味を考慮した処理との関係を明確に述べることと、他の解析手法との違いを明確にすることの2点の指摘は、本稿の改善に大きな助けとなりました。ここに深く感謝いたします。

日頃ディスカッションをしていただく米澤研究室及び木村研究室の皆様にも本稿の草稿を読んでもいただき、様々な有効なコメントをいただきました。この中でも、特に次の方々からのコメントは、研究の進行に大きな助けとなりました。助手の柴山悦哉氏からのコメントは、解析のアルゴリズムの改善に役立ちました。現在、日本IBMの三ッ井欽一氏からは、彼の認知過程のモデル[15]との対比という観点から、大学院生の大澤一郎君には他の解析法との比較の面から、萩原馨君からは先読みの単位及び長さについて、それぞれ有益なコメントをいただきました。また、日本IBMの丸山宏氏からは、ルールの選択の方法に関して有益な示唆をうけました。皆様に感謝いたします。

参考文献

- [1] Marcus, M. P., A Theory of Syntactic Recognition for Natural Language, MIT Press, Cambridge, 1980.
- [2] S. M. Shieber, "Sentence Disambiguation by a

- Shift-Reduce Parsing Technique," Proceedings of 8th IJCAI, pp. 699-703, Karlsruhe, 1983.
- [3] R. C. Berwick, "A Deterministic Parser with Broad Coverage," Proceedings of 8th IJCAI, pp. 710-712, Karlsruhe, 1983.
- [4] A. W. Carter and M. J. Freiling, "Simplifying Deterministic Parsing," Proceedings of Coling 84, pp. 239-242, Stanford, California, 1984.
- [5] 橋田浩一・山田尚勇, "自然言語の認識過程の心理学的なモデル", 自然言語処理技術シンポジウム論文集, 情報処理学会, 1984.
- [6] S. Kuno, "The Predictive Analyzer and a Path Elimination Technique," Comm. ACM, Vol 8, No. 7, pp. 453-462, 1965.
- [7] 長尾真, 言語工学, 昭晃堂, 1983.
- [8] A. V. Aho and J. D. Ullman, The Theory of Parsing, Translation and Compiling, Prentice-Hall, Englewood Cliffs, N.J., 1972.
- [9] T. Fujisaki, "A Stochastic Approach to Sentence Parsing," Proceedings of Coling 84, pp. 16-19, Stanford, California, 1984.
- [10] T. Winograd, Understanding Natural Language, Academic Press, New York, 1972.
- [11] B. K. Boguraev, "Recognising Conjunctions within the ATN Framework," pp. 39-45, in Automatic Natural Language Parsing (K. S. Jones and Y. Wilks, eds.), Ellis Horwood Limited, 1983.
- [12] T. Winograd, Language as a Cognitive Process, Addison Wesley, 1983.
- [13] 今野聡・田中穂積, "ボトムアップ構文解析システム BUP での左外置変形の処理", 情報処理学会自然言語処理研究会資料44.
- [14] L. L. Cherry, "Writing Tools-STYLE and DICTION Programs," UNIX User's Manual.
- [15] 三ッ井欽一, An Object Oriented Approach for Natural Language Comprehension, 東京工業大学修士論文, 1985.