

第 5 世代の電子計算機に関する 調査研究報告書

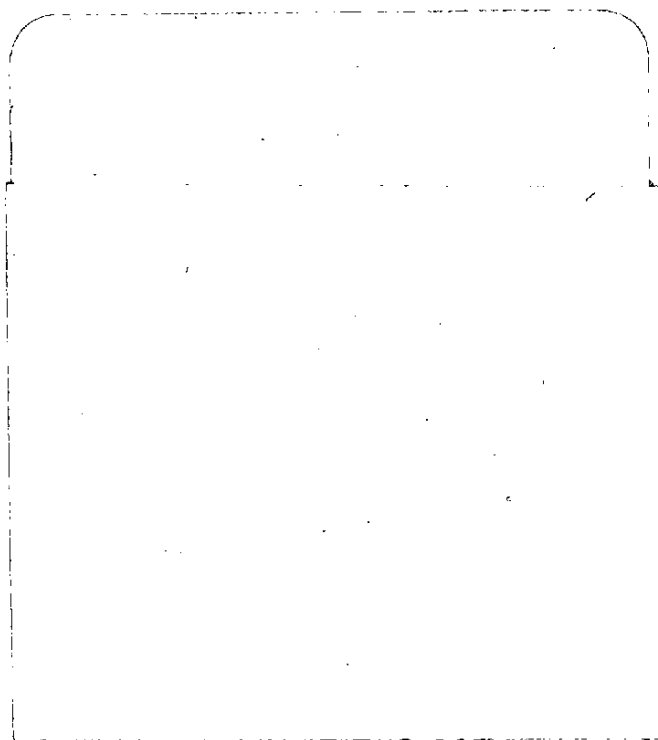
—— 言語の機械翻訳とその関連技術 ——

昭和 55 年 3 月



財団法人 日本情報処理開発協会

この報告書は，日本自転車振興会から競輪収益の一部である機械工業振興資金の補助を受けて昭和54年度に実施した「第5世代の電子計算機に関する調査研究」の成果をとりまとめたものです。





目 次

は じ め に	1
1. 機械翻訳システムの構造	3
1.1 機械翻訳システムのレベル	3
1.2 中間言語について	5
1.3 機械翻訳の方式	6
1.4 機械翻訳システムの今後の課題	10
2. 機械翻訳の歴史と現状	15
2.1 第1世代の機械翻訳	16
2.2 第2世代の機械翻訳	18
2.3 その他のシステム	20
2.4 対話型計算機支援翻訳	21
2.5 専門用語集	22
2.6 SYSTRAN の構造	23
2.7 機械翻訳システムの評価	29
2.8 EUOTRA 計画の概要	31
2.9 おわりに	34
3. 表題文の英和翻訳	37
3.1 はじめに	37
3.2 システム構成の基本的な考え方	37
3.3 プロトタイプ I のシステム構成	42
3.4 プロトタイプ I の問題点	50

3.5	プロトタイプⅡにおける改善点	54
3.6	プロトタイプⅡの評価	62
3.7	結 論	69
4.	計算機マニュアルの日英機械翻訳システム	75
4.1	システムの構成	75
4.2	日本語および英語の中間表現	78
4.3	日本語解析過程	97
4.4	構造変換過程(トランスファー過程)	129
4.5	英文生成過程	141
4.6	英文生成の実験結果と検討	157
4.7	今後の方向	159
5.	言語処理のためのソフトウェア	173
5.1	構文解析用のシステム	175
5.2	機械翻訳用プログラミング言語	183
5.3	おわりに	188
6.	辞書の役割とその構造	191
6.1	はじめに	191
6.2	国語辞典データベース	193
6.3	英和辞典データベース	202
	付 録	220

は　じ　め　に

機械翻訳は古くからの研究テーマであり、非常に難しい問題を多く含んでいるため、実現不可能な課題であると考えている人々も多い。しかし過去10年余りの間に言語理論が非常に発展したし、計算機システムの能力も格段の向上があったこと、また一方では、社会における情報流通の広汎・迅速・高度化に伴って、翻訳の機械化への要求が非常に大きくなって来たことなどから、最近ふたたび活発な研究が行われるようになって来た。ヨーロッパ共同体では、1984年には人間の適切な介入を前提とした多言語間自動翻訳を行うべく、その開発をスタートさせようとしている。

機械翻訳を実用レベルで考える場合には、これを総合システムとしてとらえ、システム設計をしなければならない。まず大量の翻訳すべきデータの入力の自動化の問題があるが、これはまだ試行的に行われている段階である。タイプ入力を前提とすれば、入力データの校正は省略できないから、そこで入力データの適切な前処理 (pre-editing) を行うことも考える必要が出て来る。翻訳に必要な辞書の内容をどのように作るかは非常に大切な問題であるが、辞書の記憶と参照の問題だけ取り上げても、データベースシステムとして十分検討すべき問題を含んでいる。

翻訳の過程においては、人間が種々の段階で機械と対話し、翻訳を助けることが必要となる。そのための対話システムと、翻訳システムむきのすぐれたテキストエディタが必要となる。翻訳の結果はテキストエディタで形式を整えた後、計算機タイプセッティングによって出力する必要がある。

要するに翻訳システムが実用的となるためには、大量のドキュメントの入力から、訳文の出力、使用者の手に入るまでについて、そのスループットの速度、翻訳の品質、コストパフォーマンスなどあらゆる点から十分な検討が必要である。このような配慮のもとに、人間の適切な援助を前提とした機械翻訳システムを作ることは、現在の技術レベルでも十分可能であると推定される。

1. 機械翻訳システムの構造

1 機械翻訳システムの構造

1.1 機械翻訳システムのレベル

機械翻訳のレベルとしては、図 1.1 に示すようにいくつかの段階が考えられる。1950年代に行われた語対語の翻訳はおくとして、1960年代には構文解析を中心とした翻訳が活発に研究された。1970年代になって、意味レベルの翻訳が試みられ、一部には文脈情報の利用、現実世界の知識の活用や、よりよい中間言語に対する努力もなされつつある。人間が発話する際に頭の中にもっている内容は、言語の種類に関係なく、人間すべてに共通の形で存在すると仮定し、それを表現する理想的な形式のことを中間言語と言っている。しかしそのような理想的な形式が存在するという保障はなく、これからの研究にまたねばならない。

機械翻訳システムの分類の仕方としては、(1)入力言語と出力言語が固定された場合の両言語のもつ特性に強く依存した2言語間翻訳システム、(2)多言語間翻訳を前提として、それらの言語間に共通なある種の中間的表現を仮定し、これを介して翻訳を行うもの、さらに、(3)意味の解析を中心としたユニバーサルな中間言語を想定して行うもの、などの区別を考えることもできる。(1)は1960年代前半までのものであり、第1世代の翻訳システム、(2)は1960年代後半から1970年代前半、さらには現在に至るまでのシステムで、第2世代のシステムと呼ばれている。(3)は現在のいくつかの野心的試みと、これからの問題である。

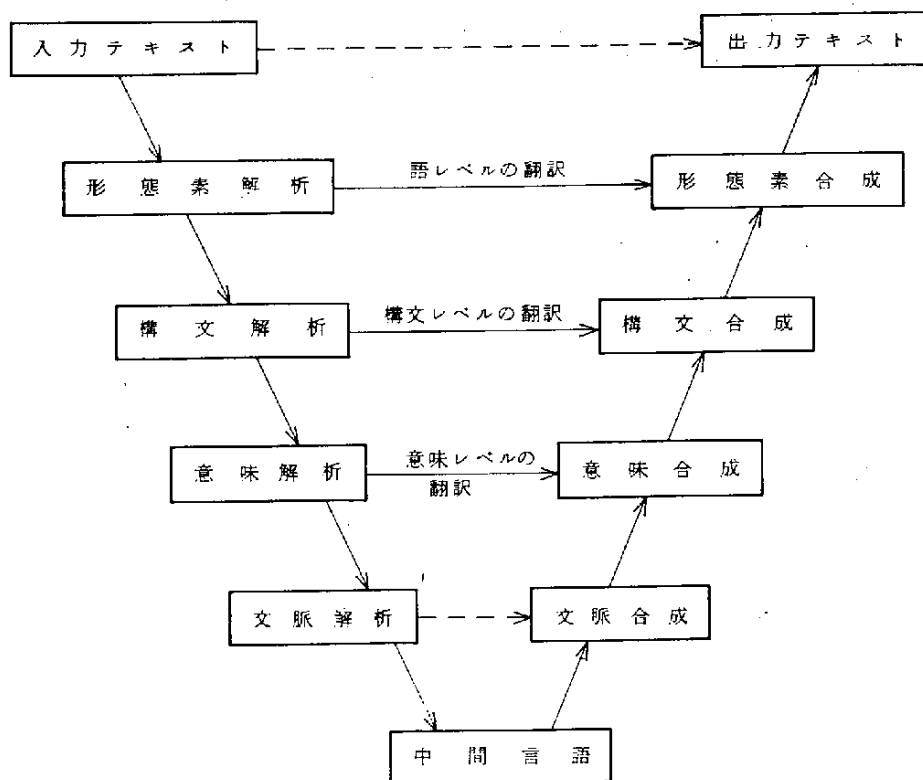


図 1.1 自動翻訳システムのレベル

機械翻訳の入力から出力までを完全に自動化することは、将来ともに非常に難しいことであり、どこかの段階で人間が介入せざるをえない。このような観点からすると、(1)入力文の前処理、出力文の後処理（修正）を中心に介入し、翻訳の途中の段階では介入せず、できるだけ高速の出力を得るようにする方式と、(2)翻訳のあらゆる段階で、機械が処理不能になったり、いくつかの選択の可能性について決定が下せなかった時などに、その状況を人間に呈示し、人間に教えを受けたり、選択を求めたりする方式の二つがある。いずれにしてもできるだけ介入を少なくすることができることが大切である。

翻訳の対象となる文章の範囲については、文学的な文章はもちろん除外する

として、一般の科学技術論文を対象とするものから、装置やシステムの仕様書、取扱い説明書などを扱うもの、天気予報の文章などのように特に限定された分野の文章など、いくつかの種類がある。これをさらに進めて、自然言語的ではあるが、非常に限定された用語と構文法則のもとに書かれた文章で、ほとんど一意的に扱えるものを対象とするシステムもある。

1.2 中間言語について

翻訳の橋わたしとなる理想的な中間言語を考えることは難しいので、比較的表面層に近いレベルのものがいろいろ考えられている。その場合には、中間言語の表現形式はそれぞれの言語にかなり依存した形のものとならざるをえない。これまでに広く用いられて来た中間表現としては、句構造表現、依存構造表現であり、より意味の世界に近づいたものとして、Schank の提唱する概念依存構造がある。また述語論理式による表現や、Montague 文法によるものなども試みられている。

句構造表現は機械翻訳の初期から広く文の解析に用いられて来たもので、単語列から句を作り、句から文を作ってゆくものである。文脈規定型句構造文法（変形文法）は、より単純明解な内部表現としての木構造をうるために、木対木の変換を行うものである。これはまた深層構造からの文の生成においても重要な役割を演じる。

依存構造文法は、D. G. Hays によって提案されたものであるが、その後 C. Fillmore の格文法（case grammar）の考え方によって、意味的にさらに明確化された文法である。格文法では、述語に対して他の句がどのような役割で存在するかについて考える。また述語に対してどうしても存在しなければならない格（必須格）と、存在してもしなくてもよい格（自由格）とを区別する。Fillmore 以後、格に対する種々の考え方が提案されている。

句構造文法が本質的に語のならば方を解析のよりどころとしているのに対し、

格文法では、句の述語に対する役割を知ることが中心となるから、語順や語形もさることながら、語や句のもつ意味がかなり明確に表現できないと解析が難しいという難点がある。しかしながら、本質的には語順によらない方法であるために、日本語のような言語の解析には非常に適したものであると言える。

述語論理による表現は、非常にしっかりした枠組の内での表現であり、種々の操作に対する保証があること、ユニバーサルな中間言語として考えることができることなどに特色がある。しかし一方では、述語論理表現は表層表現からただちに得られるものではなく、多くの場合、格文法による解析をへて得られるものであり、また述語論理表現に変換する際に、言語表現がもっていた多くの情報が落とされて、骨組だけの情報が残され、目的言語の世界での言語表現が非常に貧しいものになって、元の文のもっていたニュアンスその他多くのものを表現できなくなってしまうことなどに問題がある。

Montague 文法は、句構造文法的一种である範疇文法 (categorial grammar) によって、解析された文構造の各部分に対して、機械的に内包論理による記号論理表現を対応させ、これを整理・簡単化することによって、文内容の述語論理表現を与えることができるものである。最近この考え方にもとづく機械翻訳の研究が現れて来た。

Schank の概念依存構造や、Minsky のフレーム理論を中間言語あるいは翻訳のための橋わたしとする考え方もあるが、あまり一般的ではない。

1.3 機械翻訳の方式

機械翻訳システムは次のようないくつかの部分から構成される。

- (1) 辞書
- (2) 前処理
- (3) 形態素解析
- (4) 構文解析

- (5) 意味解析
- (6) 文脈処理・一般的知識の利用
- (7) 言語対応・移行
- (8) 文の合成
- (9) 後修正

以下にこれらの諸項目について簡単に説明する。

(1) 辞書

辞書をしっかりとしたものにすることは翻訳のための基本である。特に翻訳の対象となる文書が属している分野固有の専門用語を十分に集めることが大切である。その他に、文法関係の辞書も各種のものが必要である。辞書の構成の仕方としては、全体を1つのものとして扱うものと、翻訳システムの部分に応じて分割して使用するものとが考えられる。後者の内容としては次のようなものが考えられる。

(a) 解析のための辞書

.....	{	形態素解析, 各種文法辞書 単語とその意味, 高頻度語辞書, 不規則活用形, 慣用句 など
-------	---	--

(b) 言語間対照辞書

(c) 文合成のための文法辞書

(2) 前処理

翻訳すべきテキストを計算機に入れるとき、人手によって簡単な処理を行っておくことによって構文解析が非常に簡単になる可能性がある。翻訳すべきテキストには種々の異種の要素が混在しているからである。例えば、数式、化学式、数値、見出し部分などは、あらかじめその性質をマークしておくことによって、構文解析プログラムが取扱いやすいようにすることが考えられ

る。また長い連続した漢字列で、切れるところは入手で切っておくとか、並列句の範囲を指示したり、文節の適当なところにコンマを入れて、係り受け関係の解析を簡単にしたりすることが含まれる。

(3) 形態素解析

辞書の見出し語として、あらゆる活用形を入れておくことも一つの方法であるが、基本形だけを入れておき、その変化形は形態素解析によって認識するという方法が一般にとられている。形態素解析はそれほど難しいものではないが、100%解析が確実に行えるという保証はない。辞書にない単語が現れたときどうするかという問題や、慣用句や複合語の取扱いの問題、同形異義語の処置などの問題がある。日本語の場合には、特に長い漢字連続の取扱いが問題となる。

(4) 構文解析

構文解析はこれまでに非常に多くの方法が研究されて来た。文法として現在よく用いられているものは次のようなものである。

- (a) 句構造文法
- (b) 依存構造文法
- (c) 格文法
- (d) 拡大遷移文法

変形文法もある程度使われる場合があるが、体系的に大規模に用いるのはあまりにも複雑で、解析のプロセス全体の見通しを悪くする可能性がある。

構造解析の結果の出力としては、

- (a) あらゆる可能な構造を全部出力する。
- (b) 何らかの条件（普通意味的条件）を附加して、最も確からしい構造のみを出力する（最も確からしい構造から順位をつけていくつか出力する）。
の2つがある。

構造解析のプロセスとしては、

- (a) 左から右へ（右から左へ）あらゆる構造的可能性を調べながら、1パス

で解析を行う方法。

- (b) 文法をいくつかのグループに分け、その適用順序を規定し、何度も文を左右に走査して解析する方法。

があり、また別の観点からは、

- (a) ボトムアップ解析

- (b) トップダウン解析

の2つの方法が考えられる。

(5) 意味解析

文の解析に意味を持ち込む必要が生じる理由は種々あるが、構文解析の結果、非常に多くの可能な構造が発生するため、その中から正しい構造を選ぶためにまず必要となる。1つの単語が多品詞語であることはよくあり、同じ単語が異ったいくつかの意味を持っていることが多く、多様な構文解釈を取りうる原因となる。ある文節がどの述語に係ってゆくかについても多くの可能性が発生する。従って、ある語句とある語句とが、修飾・被修飾の関係をもちうるかどうかの検定を、それぞれの単語、語句のもつ意味的整合性によって行うことになる。日本語に対して比較的素直な方法は、いわゆる格文法を採用し、その枠組に合う意味要素を設定することが必要となる。

(6) 文脈処理・一般的知識の利用

単語や語句の意味がわかって、その係り受け関係が明らかになっても、なお残るあいまいさがある。例えば、省略された語句や代名詞が何を指すかといったことが明確にならねばならない。これを一般に文脈処理と称している。現在解析している文の前に現れた2、3の文との関係で文脈を考える場合と、さらに広く一般的な社会的知識を用いねばならない場合とがある。前者は主として代名詞の指すものを決定することである。後者は意味ネットワークと推論のプロセスに関係するが、まだほとんど解明されていない問題である。このような問題は埋め込み文と本文との関係においても問題となることがある。

(7) 言語対応・移行

構文解析・意味解析された文の深層情報構造が目的言語の文の深層情報構造と同じであればよいが（これが中間言語の考え方である）、一般的にはそうはならないので、このレベルでの入力言語と目的言語との間の対照文法が必要で、その対応関係をもとに入力文の深層情報構造を目的言語の深層情報構造に移す移行の段階が必要となる。

西欧諸語の間では、この対応関係は比較的単純であるが、日本語と西欧諸語との間では、かなりかけ離れていて、各動詞の格構造について、すべての対応関係を調べ上げて、辞書に持たせておく必要があるだろう。

(8) 文の合成

文の合成は通常、句構造文法を作り、次に適当な変形文法による構造変換を行い、最後に形態素合成を行うという段階を踏む。日本語と西欧諸語との間の翻訳においては、文の読みやすさ、スタイルについてはいまだにほとんど研究がなされていないため、どのような問題があるかさえまだ明確でない。動詞を名詞形に訳した方がよい場合とか、主語部分が長すぎて後置するなどといった多くのことが存在する。

(9) 後 修 正

ポストエディティング（後修正）は現在の機械翻訳システムではほとんどの場合必要であると考えられている。形式を整え、パラグラフを明確にし、数式や化学式等の文章でない部文を挿入したりすることの他に、読みにくさや、誤訳、多重の可能性の中からの選択といった仕事を人間が行うことになる。そのための柔軟な対語システム（ハードウェア、ソフトウェア）が必要となる。

1.4 機械翻訳システムの今後の課題

機械翻訳を完全自動で行うことは当分の間無理であることは既に述べた。そ

ここで、人間がいろんなレベルで介入し、適切な判断が下しながら翻訳を行うシステムを作ることになるが、システム全体の良否は翻訳すべきドキュメントが与えられてから、翻訳結果がタイプないし印刷された形で得られるまでについてのトータルなコストパフォーマンスを調べる必要がある。このような実験・研究はまだほとんど行われていない。ここでは、このようなシステムを作っていく上での技術的な問題のいくつかを列举することにしよう。

(1) 専門用語集

現在、日本では各種の分野における用語はほとんど整備されていない状態といつてよい。学術用語集も代表的な用語のみを収録しているだけで、網羅的に集めることはやっていない。まして多言語間対応で辞書を作る努力がない専門用語集としては、用語のコントロールを行い、スコープノート（定義範囲）をつけ、実際の用例を添えることが必要である。広く集めるという意味では、多言語間シソーラスとし、複合語、同義語、類義語、同形（音）異義語、総称語などまでを含む必要がある。

(2) 形態素レベル

形態素解析については、その方法がほぼ明らかになっているのであるから、標準的なプログラムを作り、誰でもがこれを活用できるようにする必要がある。また各種の辞書についても、それらの辞書とそれらを活用するための辞書引きプログラムも移行可能な形にして広く一般に公開する努力をすべきだろう。

(3) 文字読取機

EURODICAUTOMとTEAMではデータの入力にOCR（光学的文字読取装置）を使っているという。この種の入力装置の実用化、特に漢字の自動読取りの実用化は重要な問題である。漢字データの入力、校正に必要な費用は膨大なものである。もしこのような自動読取装置を使用できず、人手によって入力するという場合には、適当な前処理（pre-editing）を加えることは全体のコストからいうと低廉で、効果のあるものであると考えられるから、

どのような前処理を行うかはシステム全体の中で十分検討すべき要素である。

(4) 文の構造解析

文の構造解析については、これまで数多くの研究があり、あまり問題はない。大きく分けると、書きかえ規則による方法と遷移文法の考え方とがある。問題はむしろ、言語データ・文法法則と、それをどのように適用していつて解析を実際に行うかというアルゴリズムの明確な分離であり、これを支えるソフトウェアシステムの整備をうまくすることである。このソフトウェアはどの言語の解析・合成をもできるような一般性をそなえていなければならない。

(5) 文の意味解析

この部分はまだ確立されていない。今後の研究に待つ必要がある。単語の多義の処理、句のかかり方の多様性、対応する目的言語における単語、構造の選択の基準などについて、意味の面からどこまでうまく規定することができるかである。

(6) 文脈解析

連続して並んだ文の間の相互関係、その中での省略句、代名詞の推定などの方法の研究を必要とする。人工知能研究でいろいろな試みがあるが、翻訳の対象となるドキュメントの世界ははるかに広く、このような広い世界に対してよく意味ネットワーク、フレームやスクリプトを作ることができるかという疑問がある。

(7) 制限文法・制限意味

TAUM METEOやTITUSのように扱う文の構造や用語の範囲をかなり狭く、しかも一意的なものに限定すれば翻訳は楽になる。一般のユーザに不便を感じさせず、なおかつ適切な制限文法、制限意味をも受ける方法に関する研究はぜひとも必要である。

(8) ソフトウェアの問題

言語データと解析アルゴリズム・辞書操作のプログラムの完全な分離の必

要性は既に述べた。多言語間翻訳を考える場合には、優れたソフトウェアシステムが必要となる。今後さらによりシステムの開発を行わねばならない。

(9) 機械翻訳のためのテキストエディタと端末装置の開発

ゼロックス・パロアルト研究所で開発した Alto machine より以上の優れた操作性をもつ端末装置を開発し、機械翻訳に適したエディタを作ることとは絶対に必要である。

辞書内容の表示（複数辞書内容の同時表示）

文法情報（形態素法則，構文法則，など）の参照

例文の検索と表示

入力原文と訳文との同時表示，その修正印刷出力のためのフォーマッティング

など各種の機能を持つ必要がある。

2. 機械翻訳の歴史と現状

2 機械翻訳の歴史と現状

機械翻訳の研究が行われている所については、Brudererの調査報告書に詳しく出ている（少し古く、調査不足のところがあるが）。現在積極的に研究を行っている代表的な所は次のような所である。

1. Brigham Young University (Utah)
2. TAUM, University of Montreal (Montreal)
3. Grenoble University (France)
4. European Community

現在使われているシステムとしてはBrudererによれば9つあるが、本格的なものとしては

1. SYSTRAN
2. TAUM METEO

が代表的なものである。これらの他にもいくつかのものがあるが、小規模であったり、人間の修正に多くをたよったシステムである。

1960年前後には完全な自動翻訳を目ざすと称する研究が多かったが、現在では次のようなステップで考えられている。

1. 専門用語集などを計算機に入れておき、その助けを借りて人間が翻訳するシステム。(Machine-aided Human Translation)
2. 機械翻訳のプロセスの中に人間の介入する部分を積極的に作るが、できるだけ機械に翻訳させるシステム。(Human-aided Machine Translation)
3. 完全に自動的な機械翻訳。(Full-automatic Machine Translation)

これまで広く実用されて来たのは1の段階であり、これはドイツの翻訳局やECなどで既に数年以上の実績があり、100万語を超える専門用語を整備し、ECはEURODICAUTOMという専門用語集とシンラスに関するセンターを作ってよりよい辞書作成の仕事をしている。

2の段階は TAUM METEO, TAUM AVIATION, Brigham Young 大学のシステム, 香港大学のシステム, EC など多くのシステムがある。人間の介入にも種々のものがあり, pre-edit, post-edit, 翻訳プロセスの任意のところで介入するシステム, 翻訳できなかった文のみを人間が翻訳するシステムなどがある。pre-edit だけを行うシステムの極端なものとしては, 使用する用語や文体の範囲を制限し, 解釈に一意性をもたせた TITUS System がある。いずれにしても, この種のシステムの成功は人間の介入の容易さ, cost の安さにあり, 優れた翻訳用の editor の開発が必要になるだろう。

現在の多くの研究者はこの第2の段階において, 人間の翻訳と較べて cost が安く質のよい優れたシステムを作り, 実用させようということを目指しており, 1960 年前後の頃と比べると非常に謙虚な態度で着実な進歩をしようとして努力している。

以下に機械翻訳システムの代表的なものを歴史的にたどり, 現状について説明しよう。

2.1 第1世代の機械翻訳

第1世代の機械翻訳は入力言語と目的言語との対が固定していて, 両言語についての経験的な情報がもり込まれたものであって, 1966年までのものはほとんどすべてこれである。その代表例を以下にあげる。

(1) Mark II

ロシア語から英語への翻訳システムで, 1964年にUS Air Force の Foreign Technology Division に導入され, 1970 年まで使われたものである。システムはあらゆる可能な訳を出し後修正をするシステムであるが, 翻訳の質は悪かった。

(2) Georgetown システム

ロシア語から英語への翻訳システムで, 1964年にOak Ridge の Atomic

Energy Commission に導入され、現在に至るまで使われている。また Ispra (Italy) にあるヨーロッパ原子力共同体 (EURATOM) に導入され、1976 年まで使用され、その後 SYSTRAN に入れかえられた。翻訳に対する評価としては次のような数字が示されている。

良質／受け入れられる	9 2 %
情報は十分に伝えられている	8 2 %
読みうる	7 1 %
忠実度	8 7 %
使用することを推薦する人	9 6 %

(3) SYSTRAN

Georgetown システムを改良して売り出されたもので、まずロシア語から英語への翻訳システムが作られた。次いでヨーロッパ共同体 (EC) に導入され、ロシア語から英語への翻訳の他に、英語からフランス語への翻訳その他のシステムが追加されている。SYSTRAN の翻訳に対する評価として次のような数値がある。

(ロシア語から英語への翻訳)	SYSTRAN	MARK II
翻訳されなかった単語	2.3 %	1.2 %
単語の多重訳	5.3 %	6.3 %
後修正	3.5 %	
(英語からフランス語への翻訳) 後修正なし 後修正あり 人間による翻訳		
了解度	44~47 % 66 %	87~96 % 92~97 % 99 %
	(辞書を改良したとき)	(改良版)
忠実度	86~95 %	61~80 %

(4) その他のシステム

(a) LOGOS 英語からベトナム語への翻訳システムで後修正を行う。

(b) LOGOS III 英語からロシア語への翻訳システム

英語からフランス語への翻訳システム

(c) Xonics ロシア語から英語への翻訳システム

その他いくつかの商用システム，実験システムがある。これらは Bruderer の調査報告書に詳しい。

2.2 第2世代の機械翻訳

第1世代の機械翻訳が入力言語と目的言語の対を決めて、それら言語の特質を活用したシステムを作っていたのに対し、翻訳の対象となる言語が増えてゆくにつれて、 nC_2 でその対が増加してゆくという難しい問題が生じて来た。そこで1つの中間言語を仮定し、入力言語を中間に変換し、また中間言語から目的言語に合成するというプロセスを作れば、対象言語が増えても、その数だけの解析・合成のプログラムを付け加えるだけですむという考え方が出て来た。さらにシステム全体の見通しをよくし、言語学者がよりよい文法法則、辞書を書くことができるようにという目的で、言語データと翻訳のプログラムとを完全に分離するという方向に向って来た。1965年以降のシステムには、このような考え方で作られるものが出て来た。その代表的なものを以下に記す。

(1) CETA システム (グルノーブル大学)

フランスのグルノーブル大学のシステムで、フランス語、ロシア語、ドイツ語および日本語を対象として、中間言語 (syntactic interlingua, pivot language) を仮定したシステムである。言語データとプログラムとを完全に分離した点でも新しい概念のシステムとなっている。中間言語としては一種の依存構造文法がとられている。ロシア語からフランス語への翻訳の評価としては次のものがある。

正しい翻訳 42% 入力エラー，計算機容量の制限から

了解度 61% プログラムのエラー

正しい翻訳の率が低いのは、翻訳の多くの段階のどこかで誤りや解析不能に落ちいと、その文は翻訳不能として捨てるというシステムのかたさによつ

ている。

(2) TAUM システム

TAUM はモントリオール大学の機械翻訳グループの開発したシステムで、1962 年よりカナダ政府が援助して研究を進めている。英語からフランス語への翻訳システムで、言語データとプログラムとを完全に分離している。プログラムは Q システムと称する言語で、string, tree, string of trees を操作できる。このシステムで開発された実用システムとして、TAUM-METEO と名付けた天気予報の英文のフランス語訳のシステムがある。これは 1976 年始めから実用されていて、毎日 1500~2000 の比較的短い天気予報の文を翻訳している。語彙は制限され、天気予報特有の文体構造を subgrammar と称する文法体系にまとめ上げている。形態素解析は特にしていない。20% 程度の文が翻訳できないが、これは人間が直接翻訳する。翻訳のできた文については後修正を行わない。翻訳できない原因としては、入力文は前処理をなしで、タイプやスペクトル、句読点などに含まれる誤りによるものが主である。

(3) GETA システム

グルノーブル大学の CETA システムがあまりにも理想的に中間言語の形を追求した結果、種々の困難な点が現れて来た結果、システムを少し作りかえ、深層構造レベルで言語対による変換を行うシステムを作り上げた。これが言語対応・移行の部分である。GETA のシステムは任意のヨーロッパ言語間での翻訳を行うことが目的であり、現在進行中である。ここでも言語データとソフトウェアとは完全に分離されている。

(4) その他のシステム

(a) POLA カリフォルニア大学バークレイで研究されたもので中国語から英語へのシステムである。文法法則をサブグループに分け、ルールの改良を局所的に行える。

(b) SLC II EURATOM (Ispra) で作られた Georgetown システムの改

良版であるが、現在は主として自動インデクシングに使われている。

- (c) Wilks の翻訳システム Preference semantics と称して、意味を導入してあいまいさを解明して解析結果を出し、文の合成を行うシステムで英語とフランス語の間の翻訳が試みられた。

2.3 その他のシステム

現在使われている機械翻訳システムとしては、既に述べた第1世代、第2世代に属するものでいくつかあり、SYSTRAN, LOGOS, TAUM METEO, CULT システムなどがあり、研究として行われている代表的なものにはGETA のシステム、Brigham Young 大学のシステムなどがある。これら以外であげるべきものとしては次のようなものがある。

(1) TAUM-AVIATION

モントリオール大学のTAUMグループが政府の依頼で行っているもので、一般に科学技術関係の文書の英仏翻訳を目標としているが、その第1歩としてヨーロッパで開発されたエアバスオーロラの技術マニュアルの翻訳を手がけている。このマニュアルは9000万語におよぶもので、5万語の専門用語の辞書が用意されているという。現在試験的な訳文が出る段階となっており、さらに改良されてゆくだろう。

(2) LEIBNIZ グループ

グルノーブル大学のGETA、ナンシー大学、ブラジルのカンピナス大学、そして西ドイツのザールブリュッケン大学がゆるいグループを構成し、ロシア語、フランス語、英語、ドイツ語、ポルトガル語についての相互機械翻訳を進めてゆく計画である。

(3) TITUS システム

TITUS は Textile Information Treatment User's Service の略称でヨーロッパの繊維関係に関する情報システムである。この分野の文献・資料は

専門家の手で制限された文法と意味の範囲で、かなり人工的表現のテキストに書き直おされ、これがシステムに入れられる。システムはこれをデータベースに格納し、検索要求に応じて自然言語に近い文体の文章として出力する。入力、出力ともヨーロッパの数ヶ国語のいずれでもよいという特徴をもつ。入力形式はあまりにも人工的なので、これをより自由な自然言語に近いものにする改良が計画されている。

2.4 対話型計算機支援翻訳

計算機に自動的に翻訳をやらせることには種々の困難を伴ない、人間が介入しても能率のよいシステムとはならないという考え方がある。また以前から人間が行う翻訳作業に計算機ができるだけ援助を与えるというシステムが作られ、実際に使われて来た。このようなシステムでは人間が主体で計算機は従であるため、*machine-aided (human) translation* と呼ばれている。この分類に入るいくつかのシステムを紹介する。

(1) MIND システム

10 年程前に研究がやめられたものであるが、同形異義語の選択、代名詞のさすものの決定、構文のあいまいさの解消などを、計算機からの質問に答える形で、人間が対話的に翻訳プロセスの任意の時点で行うシステムであった。

(2) Brigham Young 大学のシステム

詳細は明らかでないが、*Junction Grammar* と称する文法に基づいたシステムであり、人間の介入にかなり重点の置かれたシステムといわれている。1980 年半ばにはコストパフォーマンスを測定し、人間による翻訳と比較して、よりよいものになるよう努力しているという。

(3) CULT システム

Chinese University of Hong Kong で作られたもので、中国語から英

語への翻訳システムであり、1975年以来、Acta Mathematica Siniicaの翻訳に使われている。文の前処理をかなりの程度行う。冠詞の挿入、動詞の時制・態、構文的・意味的あいまいさの解消などは人間が行うものである。計算機がうまく翻訳できなかった文については人間が翻訳を行う。

2.5 専門用語集

人間にしろ、計算機にしろ、翻訳を行う場合に大切なものの1つに辞書がある。特に人間の翻訳において専門用語の訳語を探したり、決定したりするために要する時間は大きな比率を占めるといわれている。そこで専門用語を体系的に集め計算機で管理し、端末装置やバッチプロセスで用語の訳語を翻訳者に与えるシステムが作られている。このような場合の典型的な辞書内容としては次のようなものがある。

- { 単語の意味、定義的な意味、使用法の範囲.
- { 専門用語の正規の定義.
- { 例.

特にこのような辞書は多言語間の Terminology data bank という形で、欧州を中心に作られて来ている。それらは次の通りである。

EURODICAUTOM (ヨーロッパ共同体)

TERMIUM (モントリオール大学)

LEXIS (西ドイツ) 1966年 11万語 30%ヒット率

1976年 90万語 80%ヒット率

1978年 120万語

TEAM (シーメンス) 70万語

INFOTERM (ウィーン)

2.6 SYSTRANの構造

SYSTRANはPeter TomaがGoergetownシステムを改良して商用のシステムとして売出したものである。SYSTRANは1970年から今日に至るまで、US Air Forceでロシア語から英語への翻訳に使われ、またOak Ridge National Laboratoryでも使われている。ECは1975年に導入し、種々改良のすえ、英語からフランス語、フランス語から英語、英語からイタリア語への翻訳システムを現在動かしている。

SYSTRANのもっている内部構造はあまり明確ではないが、わかっていることだけでも、比較のおもしろい内容であるので、ここに少し紹介しておこう。

(1) IBM 360/370, 400KB memory, 3330 disk が必要。

約100,000命令のシステム

(2) ・単語辞書、語尾変化の部分を除いた不変化の部分に登録。

・慣用的表現は1単語であるかのように扱う。(構文的取扱いやすさのため) on account of, as far as, give rise to, carry out light green, heat denatured, medium term dwell chamber, seal ring

・イディオム(1語ごとでは意味が出てこない)

on the one hand, for many a long year to come

・特別な意味となる表現(名詞)

blast furnace, developing nation, kitchen garden

・特別な意味となるイディオム(名詞), 個々の単語の意味はわからないが、構文的取扱いとして1つと考えるもの。

sulphuric acid (high sulphuric acid content)

cooling temperature

civil engineering (civil engineering and chemistry)

・文脈によって特別な意味となる表現

agree	—	{	(active) être d'accord
		{	(passive) convenir
another	—	{	encore (数表現の前)
		{	un autre (その他)
apparent	—	{	apparent (property という意味の名詞を修飾するとき)
		{	visible (その他)
custom	—	{	douane (複数, 他の名詞を修飾する場合)
		{	costumes (その他)

- (3) 単語辞書はアクセスの便利さのため, 単語の長さ (short, medium, long) によって, 3つの部分に分けてある。

高頻度の単語辞書: 主記憶におく

- 高頻度の単語辞書, 慣用的表現, イディオム, 特別な意味となるイディオムは文が入力されると辞書引きされる。
- 低頻度の単語辞書は品詞の多義性決定の前に辞書引きされる。
- 特別な意味となる表現は品詞の多義性決定後, 辞書引きされる。
- 文脈により特別な意味となる表現は, 文解析の後, トランスファーのレベルで調べられる。

(4) 単語辞書の内容

- 基本形
- 活用
- 多品詞の場合の選択優先度
- 品詞 (45 品詞)
- 品詞細分類 (108)

nouns (commun, propre, titre, sigle, ...)

adjectifs (simple, comparatif, superlatif, ...)

- 文法情報 (性, 数, 人称, 時制)
- 文字の大小 (大文字ばかり, 第1文字大, 小文字のみ)

- 格（主格，目的語，…主として代名詞）
- 多品詞情報（LIGHT：verb—conjugate, —infinitive, noun, adjective）
 - 8 3 の組合せがある（ROSE：verb, noun, adjective）
- 文法コード
 - ・動詞のとり前置詞 fight against, take from
 - ・動詞の型 generally transitive (take)
generally intransitive (decay)
possibly govern infinitive (continue)
possibly have predicative adjective (appear)
- 構文・意味マーカー（名詞に対して）
 - abstract, concrete, …,
- 形容詞の文法情報
 - govern infinitive (easy)
 - 名詞に続く形容詞が修飾句を作る。
(warning IMPLICIT in his remark)
- 副詞に対する文法情報
 - 動詞を修飾 (slowly)
 - 形容詞，副詞を修飾 (very)
 - 程度 (hardly)
 - 方向 (below)
- 意味コード
 - 食料品，職業，国……

(5) 訳語の工夫

- 前置詞の訳語は関係する単語に関連して訳す。
cover with de (not avec)
- 多義語

一番広い意味の訳語をとるが、前後に来る語によって熟語として取扱うことも多い。

station (poste, station, gare, centre, base, arrêt)

railway station (gare)

power station (centrale)

一方、テキストの分野によって語の意味が決まることがある。

飛行機: station → base aérienne

鉄 道: station → gare

(6) 翻訳のプロセス

- テキスト入力 (カード, 磁気テープ) とコード変換
- 文の形式を調べる (表題, パラグラフ, ...)
- 高頻度辞書, イディオム辞書 (主記憶) の照合
- ハイフンで結ばれた語の処理
- 未照合の単語をアルファベット順に並べ, disc から block 単位に取り出して来た辞書と照合。
- 未照合の単語

数, 省略形, 固有名詞, 語尾が語尾表と一致したらその文法情報を書く。

(7) 文の解析

- 左から右へ scan, 多品詞語を見つける。
- 左右の単語をみて品詞決定
- 品詞決定ができないときは, 優先度情報によって決定

he will EAT soon.

I watched Mary EAT.

they will never EAT again.

we like to watch him EAT.

} 不定形

- 特別な意味をもつ表現を辞書によって検出。それらの表現に文法的, 意味的情報をつける。

potato plant (food), treatment plant (device)

[pass 0]

- 文節の句切になるところを検出

relative pronoun, subordinate conjunction, punctuation sign, ...

- subordinate phrase の型を決め (1 1 の型)

relative, comparative, conditional, ...

- 関係代名詞より右の最初の活用動詞を検出, 続いてその右にコンマがあるかを調べる。
- なければ関係代名詞の左を見て, 活用動詞を捜す。
- なければ関係代名詞の右を見て, 2 つめの活用動詞を捜す。
- 見つければ関係代名詞の左のその動詞の主語になる語句を捜す。
- そこで関係代名詞から第 2 の動詞の手前までを 1 つの句として区切る。

[pass 1]

- 文を右から左へ scan し, 明確な構文グループをまとめる (10 種類)

形容詞 + 名詞, 名詞と名詞を修飾する名詞, 動詞 + 目的語, 関係代名詞とそれが指す名詞, ……。

- 名詞として働いている語を検出: 数, 形容詞なども
- 時, 人称, 数などの情報の検出
- MORE, MOST に関する情報の検出, これに関係する形容詞に comparative, superlatif の情報を与える。

[pass 2]

- 文を右から左へ scan し, AND, OR, , で句の区切りでないものを検出。
- これらで結合されている単語がすべて同一品詞かどうかを調べる。
- 少し広い範囲を見ることによって, あいまいさを解消する。

smog and pollution control are pressing issue.

smog and pollution control is pressing issue.

[pass 3]

- 区切られた文節の中を調べ述語を取出す。
- その主語を捜す。

それまでの解析で残された名詞，代名詞で述語の左にあるもの

述語の左に来る不定詞，現在分詞，there

疑問文のとき述語の右に来る名詞

- 主語がないとき，命令形とみなす。
- 前置詞を検出，それを支配する単語を捜す。

(8) 他言語への変換

- 前置詞の訳を与える。支配する単語の情報による。
- 意味の多義性の解消，隣接する語との関係を調べる辞書にたよる。意味マーカーによる。

(9) 文の合成

- 文を左から右へ scan し，各単語の訳語を書き出す。
- 文法的処理

人称，数，性，時制，モードを決める。

否定のとき ne …… pas を挿入

合成時制に対し être, avoir のいずれかの選択

現在分詞，過去分詞の文法的一致

正しい変化形

- 並べかえ

副詞，形容詞，名詞等の位置の変化

以上に見るように，SYSTRANは各種の膨大な言語データの集積の上に作られたシステムであるが，構文解析や辞書知識の使い方が非常に ad hoc であり，言語情報とプログラムとがごちゃごちゃに混在したもので，理論的にすっきりした形からはほど遠いものとなっている。ただ，翻訳速度を上げる工夫はしているので，かなりの速度で翻訳できるという利点はある。

これに対して理論的にすっきりとしており、プログラムと言語情報とを分離し、しかも多言語間翻訳を目ざしたシステムで最も進んだものとして、グルノーブル大学の翻訳システムがある。このシステムについては、情報処理学会CL研究会（54年7月）で解説された。

2.7 機械翻訳システムの評価

機械翻訳されたテキストはそのまま、あるいは post-editing されて印刷され、使用者に配布される。翻訳の結果について、それをどのように評価するかという問題は非常に難しく実際に評価された例はほとんどない。これは実用システムがほとんどないから当然のことと言えるかもしれない。しかし、一方では翻訳の結果の良さはかなり主観によるものであるという問題もある。TAUM METEO では、毎日約 6000 語の文について 80% 程度の正確さで訳されていると言われているが、詳しいデータは不明である。

比較的詳しく評価したものとしては EC の SYSTRAN がある。

(1) 速度

機械翻訳	IBM 370/145	2000 語/分 (0.05 分/100 語)
	IBM 370/158	6000 語/分

人手による翻訳

修正なしの翻訳..... 22 分/100 語

修正の速度..... 7 分/100 語

機械翻訳の人手による修正

翻訳専門家..... 22 分/100 語

非翻訳専門家の技術者..... 8 分/100 語

(2) 了解度

原文	99 %
人手による翻訳・修正済み	98 %

機械翻訳	78%
機械翻訳・人手による修正	98%
(3) 忠実度	73%
(4) 修正率	
人手による翻訳の修正	12%
機械翻訳の修正	
翻訳専門家	36%
非翻訳専門家の技術者	31%
(5) 修正の対象となった誤りの原因	
原文	5%
入力	1%
機械翻訳システム	80%
単語	50%
解析	16%
合成	10%
その他	4%
文体	14%
(6) 利用者の読む速度	
原文	3700語/時間
機械翻訳(修正なし)	3200語/時間
人手による翻訳(修正)	5000語/時間
機械翻訳(修正)	4300語/時間
(7) 機械翻訳の受け入れられ方	
限られた所でなら使える	88%
自分が使うとき機械翻訳(修正なし)でも使う	71%
(8) 費用(フランス・フラン)	
入力	15F/100語
機械翻訳	5F/100語

あと修正

45 F / 100 語

タイプ

6 F / 100 語

従って、既に磁気テープにテキストがある場合、あと修正なしの機械翻訳を使う場合、Text editor が十分よくてタイプしなくてよい場合など、種々の使用の場合を考えると、5 F / 100 語 ~ 7 F / 100 語の範囲となる。

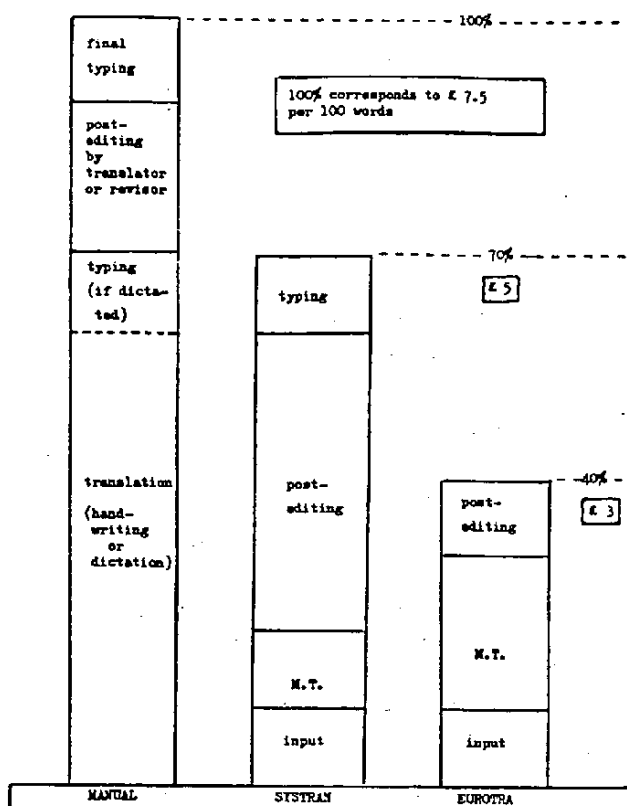


図 2 - 1. 翻訳に要する費用の比較

2.8 EUROTRA 計画の概要

EC は何らかの意味で多言語問題に予算の半分以上を使っているという。現在 EC の公用言語は 6 ケ国語であり、すべての公式文書は他の 5 ケ国語に訳さ

れねばならない。そのために現在1300人の言語専門家をやとい、1976年だけで538,000頁の文書を翻訳したという。そこで、1976年には SYSTRANの英語-仏語版を導入し、現在は仏語-英語、英語-伊語の版も動き出した。そして前節に述べたように、SYSTRANはcost的にも人間に対応できることが明らかにされた。今後はEURONETを通じてこのシステムを加盟各国で使用する計画である。しかし今後6ヶ国語全部、さらに今後予定されている9ヶ国語相互間の翻訳となると、1言語対1言語の翻訳しかできない SYSTRANでは対応できないことは明らかである。そこで、ECは次のような4つの項目について開発を行うことになった。

- (1) 多言語間シンソーラスとそれを支えるソフトウェア
- (2) 翻訳者を助ける専門用語データベースの整備
- (3) 現在の需要に応えるため、現在の機械翻訳システムを使用し改良すること。
- (4) これまでの経験を生かして、新しい高度な機械翻訳システムを作ること。

これらのうち、最後の新しい機械翻訳システムは、(1)多言語間機械翻訳システムで、(2)言語理論や情報科学の研究成果の上に立ったSYSTRANより質のよい機械翻訳システムを作る計画となっている。その中心となるのはGrenoble 大学(フランス)、Saarbrücken 大学(西ドイツ)、Pisa 大学(イタリア)、Manchester 大学(イギリス)であり、これらの他に、オランダ、デンマークの大学も参加する。このシステムは第2世代のシステムとして、Grenoble 大学などが開発して来たものを中心とするもののようで、ECの計画では1980年初頭からはじめて、1982年には実験的システムを完成すること、さらに1984年には少なくとも4ヶ国語間での完全な実用システムとすることを目ざしている。そのシステムをSYSTRANと比較したものを表2-1、図2-1に示す。システムは辞書と言語システム、およびソフトウェアは互いに独立であるようにし、多言語間翻訳は辞書と言語データを付け加えるだけでよいように設計し、かつ異種の計算機でも使えるよう完全なportabilityを目ざしている。そしてtotal cost は現在の半分以上を目ざしている。

多言語間翻訳を各国が共同して作るというとき、ソフトウェアは1つの共通なものであるが、辞書と言語情報とをシステムに合った形で整備しなければならない。各国は自国語の解析と合成の部分を受持ち、言語間の対照・変換の部分は対応する2ヶ国の研究者の共同作業になるという。仕事量の予測は次のようになっている。(言語Aから言語Bへの翻訳)

文法：解析文法	12人年
生成文法	6人年
対照文法(トランスファー)	3人年
辞書：単一言語辞書	10人年
2言語間辞書	3人年
計	34人年

なお、言語Aから言語Cへの翻訳システムをAからBへの翻訳システムに付加するための仕事量の予測は次の通りである。

目的言語の単一言語辞書	5人年
2言語間辞書	3人年
計	8人年

表 2 - 1 : E C における翻訳システムの概要

	SYSTRAN	EUROTRA
development period	(E.C.) 1975 - 80	1980 - 84
developed by	World Translation Center (Dr. P. Toma)	C.E.C. + six European universities
quality objectives	intelligibility > 90% (now 78%) revision rate < 20% (now 33%)	intelligibility 99% revision rate 10%
fields of application	unedited: information retrieval edited: industrial brochures	unedited: internat. communication edited: publications
languages (to be) covered by EC development	E → F, F → E, E → I,	E, F, D, I, NL, DK in all combinations
D.P. support	IBM and SIEMENS	full portability
dictionary formats	compatible	

2.9 おわりに

機械翻訳のシステムは、

第 1 世代 言語データとプログラムの非分離、あまり言語理論的でないシステム、1 言語対 1 言語の翻訳システム

第 2 世代 言語データとプログラムのシステムの分離、言語理論にあった見通しのよいシステム、多言語間翻訳

第 3 世代 意味を取り入れたシステム

のように区分することが行われている。おそらく EC のシステムは第 2 世代のシステムの最後、完成とみなすべきものとなるだろう。

機械翻訳研究は世界各地で再び盛んになって来た。これらすべてを調査することは、FID/LD の 1 つの仕事にしようということになっている。今後は、オフィス・オートメーション、ドキュメンテーション・システム、データベース

ス・システム、計算機との対話システムなどと助け合いながら発展してゆくものと考えられる。とにかくECのEUROTRA計画が1984年に出来あがって、それがどのような評価を受けるかということは機械翻訳の将来にとって最も大きな要素であり、逆にそのために、ヨーロッパのこの計画に関係している研究者は非常に慎重な態度をとっている。1960年初めの失敗があるから、今度は絶対に失敗できないというわけである。彼らはしかし、TPOをよく考えれば必ず成功するものと信じて進めているのである。

参 考 文 献

- (1) H.E. Bruderer : Handbook of machine translation and machine-aided translation, Amsterdam, North-Holland, 1978.
- (2) G. Van Slype, I. Pigott : Description du Systeme de Traduction Automatique SYSTRAN de la Commission des Communautés Europeennes, Bureau Marcel van Dijk, report to EC.
- (3) G. Van Slype, I. Pigott : Evaluation du Systeme de Traduction Automatique de la Commission des Communautés Europeennes, Bureau Marcel van Dijk, in Proc. 3ème Congrès National Français sur l'Information et la Documentation, mars, 1979.
- (4) L. N. Rolling : The Second Birth of Machine Translation, A Timely Event for Data Base Suppliers and Users, Commission of the European Communities, CETIL/149/79.
- (5) L. N. Rolling : The Facts about Automatic Translation, Commission of the European Communities, CETIL/67/78.
- (6) B. Vauquois : ヨーロッパ共同体における自動翻訳計画, 情報処理学会 CL研究会 CL18-1, 1979. 7. 11.
- (7) B. Vauquois : 自動翻訳のためのソフトウェア・サポート, 情報処理学会 CL研究会 CL18-2, 1979. 7. 11.
- (8) 長尾 真 : 情報科学と自然言語, 情報処理 Vol. 20, No. 3, 1979.

3. 表題文の英和翻訳

3 表題文の英和翻訳

3.1 はじめに

機械翻訳システムのように、非常に多くの未解決の問題を持っているシステムを作成してゆく場合には、まず現在の技術の範囲内で比較的簡単に作成できるプロトタイプシステムを作り、そのシステムでの経験から、残された問題と、もしそれが解決したとしたら、どの程度のパフォーマンス向上が得られるかを整理し、次に解決すべき問題点を明確にしてゆくべきであると思われる。特に工学的立場からは、あまりに詳細な言語現象についての解明を行なっても、それが全体に及ぼすパフォーマンス向上につながらない場合もあることを考えると、プロトタイプシステムの作成と、その問題点の明確化は重要な作業になる。

我々は、このような観点から、1つの機械翻訳システムのプロトタイプモデルを作成し、これによって実際に翻訳実験を行なった（昭和53年度）。この実験の結果をフィードバックし、さらに次のプロトタイプモデルの作成とその結果の検討も行なった（昭和54年度）。ここでは、この2つのプロトタイプモデルについて述べる。機械翻訳の対象としたのは、科学技術論文の表題（J I C S T 発行の文献速報磁気テープ、電気工学編10,000文献）であり、翻訳の方向は英語から日本語とした。

3.2 システム構成の基本的な考え方

自動翻訳は一般に次のようなステップをへる。

- (i) 原文入力
- (ii) 原文の形態素解析

- (iii) 原文の構造解析
- (iv) 原文の意味解析
- (v) 原文情報から訳文情報への変換
- (vi) 訳文の構造合成
- (vii) 訳文の形態素合成
- (viii) 出力プリント

これらのうち(ii)の形態素解析 (vii)の訳文の形態素合成の部分は、過去にしっかりした研究もあり、やればほぼ出来る範囲のものである。ここでは研究の焦点を(iii)と(v)にあて、(iv)についてある程度の処理を行なうという立場をとった。その理由は次の通りである。

- (1) 意味の問題にはいくつかのアプローチがあるが、せまく限られた明確な世界について適用はできても、一般の分野に適用できるような確立された手法はいまだにない。
- (2) 意味を十分にあつかうためには、現実世界に関する知識も採用しなければならない。
- (3) 技術論文の表題の翻訳というような実際的な分野の場合に、(1)(2)のような意味を積極的にあつかうのではなく、逆にどこまで意味のことを考えずに実用的なところまでゆけるかを試みる方が实际的である。
- (4) 計算機はほう大なデータの処理にむいているから、複雑でこった方法を考えるよりも、種々の場合をなるべく多く集めて、それによって表題の解析と翻訳を考える方がよいのではないか。

このように、論文表題の翻訳を単純なプログラムとできるだけ多くの場合をおおったデータ、法則とで行なってみようというところに研究の焦点をあてた。したがって、入力文、合成文ともに形態素の処理は行なっていない、構文と変換の部分のみのテスト結果がわかるようになっている。なお、辞書としては、1万件の英文表題にあらわれた9,800語(形態素処理は行なっていないので、morphological variantもすべて独立した1語として登録されている)に

ついて、品詞、日本語処理がつけられている。

論文表題は、機械翻訳システムの対象としてみると、次のような特徴を持っている。

- (1) 表題文中の単語は専門用語がほとんどで、他の分野に比べて多義語の問題を排けることができる。

もちろん、専門用語であっても各専門分野ごとに別個の訳語がつけられている場合もあるが、各訳語ごとに分野コードを指定する等、簡単な措置で多義語を排けることができる。ただし、訳語と分野コードの指定は現在のシステムでは行っていない。

- (2) 表題文固有の文体構造がある。

特に、Experiment on ———, Application of ——— to ——— 等のように、表題文全体の構造を決める英語における特有の言い回し方は、それに対応する日本語訳のパターンを用意することにより、前置詞の訳出方法など、従来困難と考えられていた問題をある程度排けるのに有効である。我々は、この種の特定の単語に固有な文型パターンをFlexible Idion (以下FI)と呼び、各単語の辞書に登録している。

- (3) 専門用語は、必ずしも1語単位ではなく、句に対応することが多い。

例えば、Large Scale Integrated Circuit (大規模集積回路) のように、専門用語では、複数個の単語で1つの概念を表わすことが多い。この場合、訳出の過程で個々の単語の訳語から全体の訳語を合成していたのでは、ほとんど理解不可能な訳文が生じる。人間の専門家が専門用語の訳語を決定する場合に、個々の英語の単語の訳から合成するのではなく、その句全体が表現している概念を自然な日本語で表現することによって行なっていることを考えると、すなわち、専門用語の日本語訳を決定する際に人間の創意が入っている以上、この過程を計算機に行なわせることはほとんど不可能であり、これらの句も1つの単語と考える必要がある。そこで我々はこれらの句に相当する専門用語も1つの独立した辞書項目として設定することにした。実際

には、その句中に出ている単語の辞書記述中にそれらの句を登録している。

この他、各分野でよく使われる固有の表現 (Idiom 的な表現) ——
time varying (時間とともに変化する), Settling time (整理時間)
——あるいは、英語全体にみられる Idiom 的な表現 (based on ——, perpendicular to —— 等の群前置) もまた、辞書中に記述され、辞書引きの段階で単語と同じように取扱われる。これを (2) の FI に対して Rigid Idiom (RI) と呼ぶ。

一般に、単語の単位を大きくとればとるほど、辞書の規模は大きくなるが、逆に後での処理が簡単になり、訳文の質も向上する。特に、ある特定の専門分野を対象を固定した場合には、この手法は有効であろう。この RI を選定する過程で、学会発行の学術用語集等を参考にしたが、用語数が少なく、ほとんど役に立たなかった。現実的な機械翻訳システム作成のためには、現在使われている用語を網羅的に収集した Terminology Data Bank 的なものを用意する必要がある。

- (4) 表題文においては、連続した名詞、形容詞によって名詞句を構成する電文調の表現が多くあらわれる。

通常の文体では、前置詞を使って結合される名詞が、表題文においては、単に連続するだけで結合され、名詞句を構成することが多い (例 —— High Resolution Bragg Reflection Method, Pulse Compression Distance Measuring Equipment System etc)。

これらの名詞間、形容詞一名詞間の結合関係を同定し、訳文合成の際に適切な語順変換と、助詞の挿入を行なうことはもちろん理想的ではあるが、これを行なうためには、かなり深い意味的な解析を必要とし、現在の時点ではほとんど不可能である。

一方、日本語においても、表題文では、かなり長い漢字連続で英語と同様な表現をとることが可能であり (上記例の場合には、「高分解ブラック反射手法」、「パルス圧縮距離測定装置システム」となる)、しかもその中での英語と日本

語の語順はほとんど変化しない。

したがって、現在の我々のシステムにおいては、このような長い名詞連続内での単語間の詳細な係り受けの関係は同定せず、単にそのような長い名詞句があったということだけを簡単な Transition Network (以下TN)文法で認識することにした。現在のシステムでは日本語と英語の語順が変化するのは、前置詞、—— ed 形、—— ing 形等の単語があらわれた場合だけであると考え、形容詞を含む名詞連続では語順がかわらないと考えている。したがって、ここで使われるTNはこのような語順の切れ目を示す語があらわれるまでを、読みとばして1つの名詞句としてまとめてゆく(ただし、{冠詞}+{—— ed 形}の—— ing 形のように明らかに形容詞的に使われていることがわかる。—— ing 形、—— ed 形の場合には、語順を変換する必要がないので、TNによってそのまま読みとばされる)。

(5) 表題文にあらわれる文構造は、書き換え規則で処理する必要があるほど多様ではない。

言語分析において、書き換え規則に代表される規則の体系を使用する大きな理由は、言語表現の持つ「無限」の可能性を、有限個の規則で処理したいためである。しかしながら、表題文においては、表題文全体の長さが制限されていること、埋込み文等の複雑な文体がほとんどあらわれないこと等の理由によって、すくなくとも品詞並びに還元してみる限り、規則で処理しなければならないほどの無限の可能性があるとは考えられない。しかも、前述の(3)の項で述べたように、句で表現される専門用語を1つの辞書項目と考え、また、(4)で述べたように長い名詞句連続をTNによって1つの名詞に縮退する操作を行なうと、可能な品詞の並びはさらに局限される。

このような考え方から、我々は表題文の品詞列を処理する際に書き換え規則等の規則による処理は行なわずに、表題文にあらわれる可能な品詞並びをそのまま列挙しておき、これと入力表題文の品詞並びとのパターンマッチングを行なうことにより、表題文全体の訳出語順を決定することにした。入力表題

文から、種々の操作を経て得られた縮退した品詞並びをその入力表題文の骨格パターン、システムによってあらかじめ用意されている品詞並びを文型パターンと呼ぶ。したがって、入力骨格パターンとシステムに登録されている文型パターンのパターン・マッチングによって、訳出語順が形定される。

一般に規則の数を少なくするためには、局所的な言語の構造を処理するための規則を用意し、その組み合わせで文の大域的な構造を把握することになる。同じように、解析の結果を訳出する場合にも、逆方向の規則によって、局所的な構造を一部分づつ線形表現に変換してゆくことになり、全体として訳のバランスや、より大域的な観点から決めるべき語順の決定において困難がある。

以上のような理由からも、表題文のように文体に強い制限がある場合には、我々のとった文型パターンによる訳出がすぐれていると思われる。もちろん、後述する文型パターンに付随した意味処理の指定等が、本来的には1つの統語規則について宣言されるべきものが、同じ部分構造を持ついくつもの文型パターンについて宣言されなければならないという欠点がある。

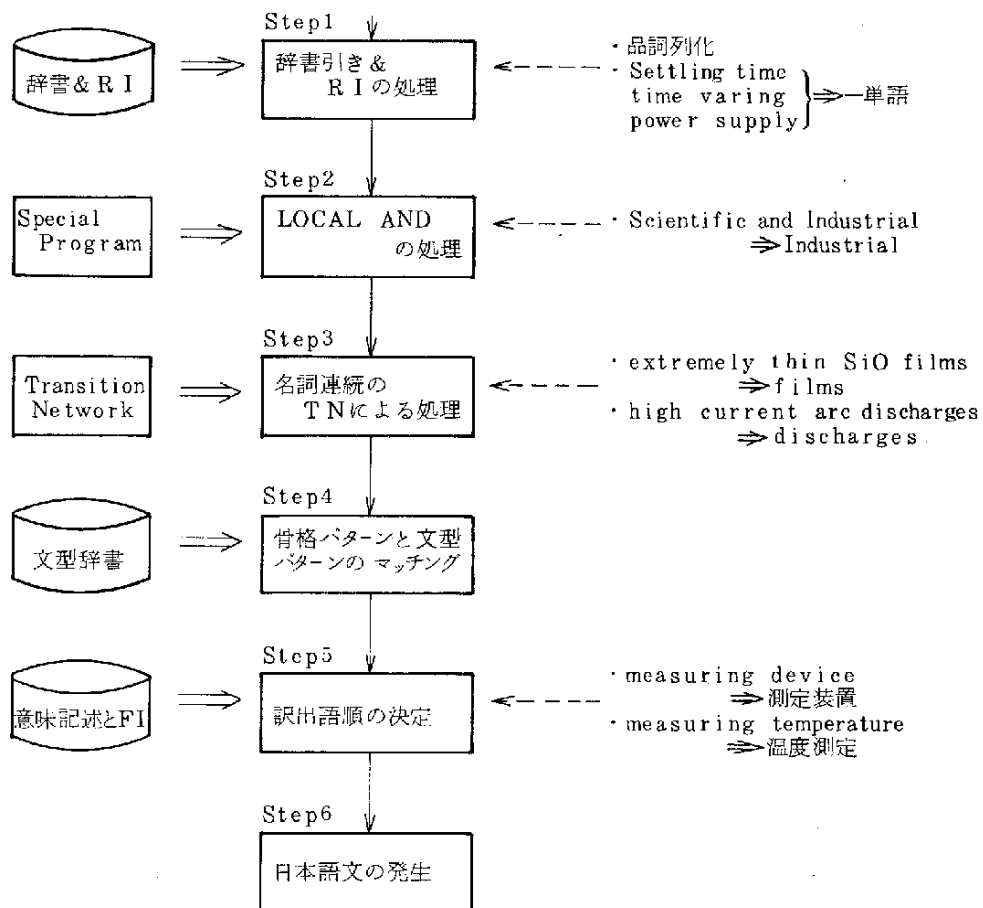
以上が、表題文翻訳システムに対して、我々がとった基本方針であるが、このうち(5)の骨格パターンに対する考え方は、53年度に作成されたプロトタイプⅠのシステムに強く反映され、54年度に作られたプロトタイプⅡにおいては、いく分弱められ、プロトタイプⅠでは文型パターンによって処理されていた多くの構文を局所的な解析によって処理することにより、文型パターンの数を減らすことになった。

3.3 プロトタイプⅠのシステム構成

プロトタイプⅠでの翻訳は、概略次の各ステップによって行なわれる(図3-1に処理の流れと、実例を示す)。

Step1 : 辞書引きおよびRIの処理

Step2 : LOCAL ANDの処理



翻訳ステップの例

入 力 : Industrial and Scientific Techniques for Measuring field Effect Mobility

Step 1 : Industrial and Scientific Techniques for Measuring Effect Mobility

Step 2 : Scientific Techniques for Measuring Effect Mobility

Step 3 : Techniques for Measuring Mobility

Step 4 : <文型パターン>

n	prep	ing	n
---	------	-----	---

訳出結果 : 電界効果移動度測定のための工業的及び科学的手法

図 3-1 表題文英和翻訳システムの構成とその実例

Step3 : 名詞連続の処理

Step4 : 骨格パターンと文型パターンとのパターン・マッチング処理

Step5 : 簡単な意味処理とFIを使った訳出語順の決定

Step6 : 日本語文の合成

以下では、表題文の各ステップの処理の概要について述べる。

[1] Step1——辞書引きおよび Rigid Idiom の処理

現在のシステムでは、英語の形態素処理を全く行なわないために、morphological variantもすべて辞書項目として含まれている。もちろん、実用化の際には規則化可能な部分を規則化することによって、辞書項目数を減少させる必要がある。

ただし、同じ‘V + ing’の形式で作られた語であっても、accounting, bonding, scattering, engineering 等はほとんどいつも名詞的に、superconducting は形容詞的に、using はもっぱら目的語をとってその直前の名前の名詞を修飾して前置詞的に、determining は他動詞的に使われる。一般に‘V + ing’形はこれらのすべての使われ方をする可能性があるが、これらの語については、分野を固定するとほとんどある一定の使われ方をする。したがって、辞書記述の中にこの固定的な使われ方をするということを書き記しておくことにより、後の段階での複雑な処理を排けることができる。また、time varing, based on のように、RI に組込まれているものもある。英語の形態素処理を一般的なルールで行なうことが意味をもつためには、それに対応する日本語の形態素発生がうまくルール化できて、しかも、英語の解析、語順決定等がその形態素で変形をうけた元の単語とは無関係に実行できることが必要であるが、上述のように元の動詞が何であったかによって、‘V + ing’形の果す役割がかわる場合には、結局単語ごとに ing 形の付いた場合の記述が必要となり、個々の V + ing 形を辞書項目として立てることとほとんど大差がないことになり、いたずらに処理を複雑にするだけで、利点がないことになる。

[II] Step2 — LOCAL AND の処理

ここでは、and だけでなく、and・or・but のような等位接続詞によって作られる並列句一般の処理を行なう。実際には、対象とした表題文中には、等位接続詞としては and だけしかあらわれなかった（1万表題中、and を含む表題 722 件、内 2 つ以上の and を含むもの 51 件）。

等位接続詞を含む句の解釈には、よく知られているように、scope ambiguity がある。例えば、Controlability and Stability of the system を「システムの可制御性と安定性」と訳すべきか、「可制御性とシステムの安定性」と訳すべきかは、結局 Controlability, Stability, System の 3 語間の意味的關係によって決定せざるを得ず、意味処理の問題と密接に関連する。これを「システムの安定性と可制御性」のように、並列されている句の順序を入れ替えて、日本語においても同じ曖昧さを持つ表現に置き換えることもできるが、本質的な解決にはならない。現在の我々のシステムでは、このような曖昧さを持つ等位接続詞の解析は未解決な問題として残されている。

我々のシステムでは、文全体の大域的な構造を参照しなくても、局所的な品詞列だけから決定できる「形容詞 and 形容詞」等の連続だけを処理する（図 3-2）。

マッチすべきパターン →		結 果
(top) det adv ing ed v prep	adj and adj n →	(top) det adv ing ed v prep
ing and ing →		ing

図 3-2 局所的な AND の処理

〔Ⅲ〕 Step 3 —— 名詞連続の TN による処理

ここまでの Step でつくられた品詞列は、R I と簡単な構造の並列句を処理しただけで、入力表題の品詞列とほぼ同じものである。従って、このままでは入力表題の数だけ異なった品詞列があり得る。この Step 3 では、語順の変換を必要としない。' n + n ' やそれに形容詞が挿入されたような品詞列を、1つの名詞句に縮退する。これを行なうための TN を図 3-3 に示す。

前に注意したように、この TN はあくまで Acceptor 型の TN で、入力品詞列に対する構造を作り出す transducer ではない。

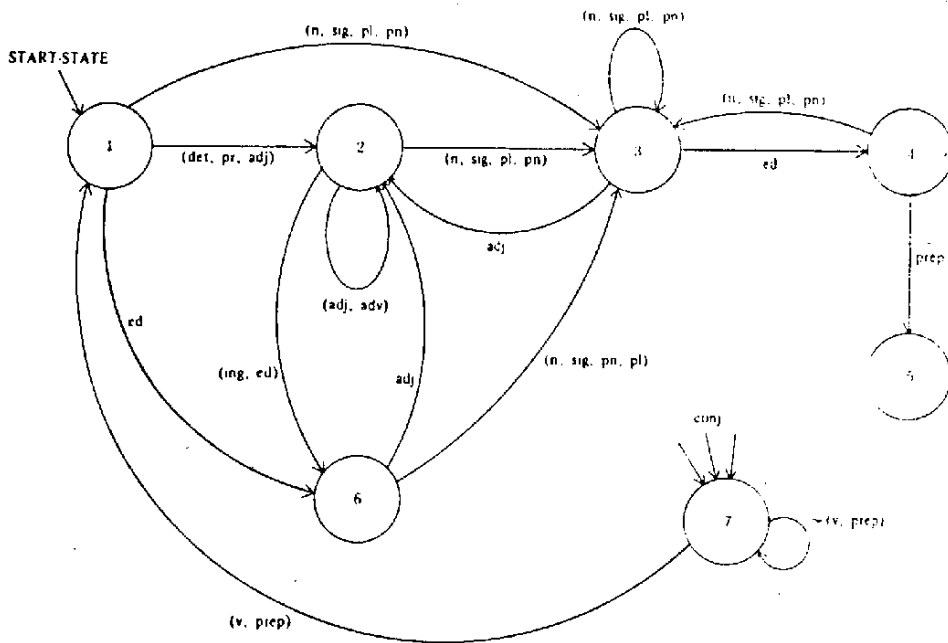


図 3-3 名詞連続を処理する TN

この Step によって、次のような句が 1 つの単語に縮退される。

(例) This film waveguides → waveguides

High current arc discharge → discharge

Pressure control system → system

High quality phosphor screens → screens

Transmission Line Power Flows → Flows

An Automated General Purpose Test System → System

Eddy current method → method

Time varying strong magnetic fields → fields

The refractive index profile → profile

stochastically disturbed industrial plants → plants

IV) Step 4 — 骨格パターンと文型パターンとのパターン・マッチング処理

以上の Step で、入力表題文に対応して、図 3-4 に示すような骨格パタ

(1) Experimental Method of Measuring the Anisotropy Field in Small Samples

Skelton Pattern: Method of Measuring Field in Samples

品詞列 : n prep ing n prep n

(2) A Laser Doppler Technique for Measuring Flow Velocities in High Current Arc Discharge

Skelton Pattern: Technique for Measuring Velocities in Discharge

品詞列 : n prep ing n prep n

(3) Voltage Time Integral Method for Measuring Machine Inductance

Skelton Pattern: Method for Measuring Inductance

品詞列 : n prep ing n

(4) Diffuse Field Technique for Measuring Directivity Index

Skelton Pattern: Technique for Measuring Index

品詞列 : n prep ing n

(5) Apparatus for Thermopower Measurements on Organic Conductors

Skelton Pattern: Apparatus for Measurements on Conductors

品詞列 : n prep n prep n

(6) The Design of a Pressure Control System for the Pepsius Spectrometers

Skelton Pattern: Design of System for Spectrometers

品詞列 : n prep n prep n

(7) Evaluation of High Quality Phosphor Screens for Image Tubes

Skelton Pattern: Evaluation of Screens for Tubes

品詞列 : n prep n prep n

図 3-4 入力表題文とその骨格パターン

ーンが抽出される。Step 4 では、これに対してシステムに用意された文型パターンとのマッチングが行なわれる。

[V] Step5 —— 訳出語順の決定

Step 4 で入力文の骨格パターンが見つけれられるが、一般に 1 つの文型パターンには複数個の訳出語順があるので、この Step では単語の意味記述と FI を用いて訳出の語順が決定される。

FI は、ある特定の単語に固有な表題文の文体を列挙しているもので、これに合致する場合には、その訳出語順及び前置詞句等の訳は、これが優先される。もし、FI を持つような単語がなかったり、もしあったとしても、入力表題文の骨格パターンがそれに合致しない場合には、システムの用意する一般の文型パターンに対する訳が採用される。図 3-5 にその模式図を示す。

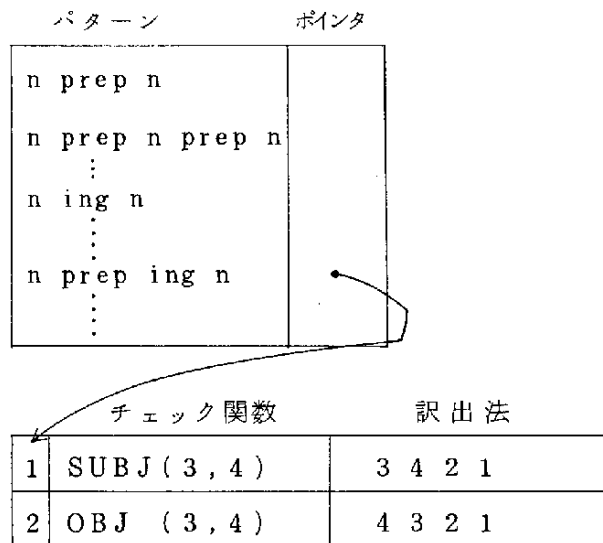


図 3-5 骨格パターンと訳出順序の決定

図に示すように、各訳出語順には、その訳出語順を意味的に支持するチェック関数が用意されている。すなわち、我々のシステムにおいては、必ずしも意味的な記述が完全に与えられることを仮定していないので、Wilksの preference semantics の手法と同様に、あくまで意味処理は補助手段になっている。意味処理関数をはたらかせた結果、意味的には製合がとれていることが判った場合には、その訳出語順の優先度が上がることになるだけで、意味的な整合が確認されなかったとしても、必ずしもその訳出語順が棄却されたことにはならない。

訳出語順の決定においては、——ing形の存在がもっとも大きな影響を与える。すなわち、——ing形があった場合には、measuring device, measuring temperatureのように、日本語訳において他の単語との結びつき方によって語順が変化するためである。これを処理するために、我々は各動詞に関して、その動詞が主語、目的語としてどのような意味カテゴリーの名詞をとるかを示し、これに対応して、名詞の意味カテゴリーに分類した。その1例を図3-6に示す。

道具	probe, instrument, equipment system, set, machine unit
理論	technique, principle, method approach
観点	velocity, displacement acceleration, position, energy inductance, resistance capacitance, conductance temperature mobility, permittivity intensity level parameter coefficient, time, angle
物体	laser, beam, light solenoid, coil film, tape, disc metal MOS, device, waveguide diode, transistor oil, water, liquid, solid air, earth star, stone

図3-6 名詞の意味カテゴリーの例

名詞の意味カテゴリー：動詞の意味記述は、このような簡単なものではないが、現在どの程度の意味記述が実際に質の良い訳文を得るために必要かの検討を行なっている。

ここでいう訳出語順の決定は、実際には骨格パターン中の係り受け関係の同定になっているが、この過程で問題になるのは前置詞句の修飾先である。英語では、前置詞句はそれが修飾する語よりも後に置かれ、日本語では全くその逆になるので、英語の語順を完全に逆転させておけば、英語において前置詞句の修飾先が曖昧なら同じように日本語においても曖昧な表現が得られ、プロトタイプの翻訳システムとしては十分である。しかしながら、考えている英語の前置詞句が動詞（——ing形、——ed形を含む）を修飾しているか、名詞を修飾しているかによって、対応する日本語での修飾句の語尾変化が変わる（連用修飾句か連体修飾句か）ため、このStep 5における係り受け関係の決定は、より質の高い訳文を得るために重要である。現在のシステムでは、次にくるStep 6の日本語の形態素発生の処理を行っていないため、このStep 5で動詞にかかるとされた前置詞句もすべて連体修飾語として生成しており、これが訳出結果の質を落す原因にもなっている。このStep 5の訳出語順の決定をうまく形態素発生に結びつけるStep 6を開発する必要がある（現在のシステムでは、なるべく動詞的概念も名詞化して表現しているため、これによる不自然さはいくぶん緩和している。——（例）A Method of measuring velocity by doppler effect
——→「ドブラー効果による速度測定手法」（現システム）
——→「ドブラー効果によって速度を測定する方法」）。

3.4 プロトタイプⅠの問題点

ここでは、プロトタイプⅠの翻訳実験の結果を検討し、プロトタイプⅡへの移行を考える。以下に、プロトタイプⅠの問題点を列挙する。

(1) ANDを含む表題文

ANDを含む表題文は、全体（1万表題）の約2割存在する。プロトタイプIでは次のような局所的な文法関係から係り受け関係が容易に決定できる修飾語の接続のみを処理している。

(例) Industrial and scientific techniques for measuring mobility.

しかし、このような例はANDを含む表題文中10%程度にすぎない。ANDを処理せずにそのまま置いておくことは、必要な文型パターン数を膨大なものにすることになるため、残りの90%のANDを含む表題文をどのようにして処理するかが大きな問題である。

(2) —ingを含む表題文

プロトタイプIでは、—ingは次のような特殊な場合以外に縮退せず骨格パターンにまで残る。

(例) Optical measuring techniques as a diagnostic aid in circuits breaker development.

したがって、このために必要となる文型パターン数はかなり多くなる。例えば、前置詞を2つ含む表題文として次のパターンがある。

N prep N prep N

図3-7に示すように、前置詞に囲まれたNは、N, N ing, ing, ing N, ing ing, N ing N, ……と換わることができる。例にあげたパターンのNをここに示した6個の要素で置き換えることによってできるパターンは、 $6^3 = 216$ 個である。前置詞が3つになれば $6^4 = 1296$ 個のパターンが必要となる。これらの文型パターンを全て用意することは果して有効であろうか。もし—ingを文型パターンとマッチングを行なう以前に処理することができれば文型パターン数は極端に減少すると思われる。

N prep N prep N
 N prep N prep ing
 N prep N prep N ing
 N prep N prep ing N
 .
 .
 .
 N prep ing prep N
 N prep ing prep ing
 N prep ing prep N ing
 N prep ing prep ing N
 .
 .
 .
 N ing prep N prep N
 .
 .
 .

図 3-7 ing による文型パターンの増大

(3) 過去分詞及び不定詞

過去分詞の中には修飾語を伴い直前の名詞を修飾するものがある。また、不定詞の形容詞的用法もやはり前の名詞を修飾して名詞句を生成する。このような名詞句を文型パターンとのマッチング以前に処理していないシステムでは、これらの文のために多数の文型パターンを用意しなければならない。なぜならば、(2)で述べた——ing 同様、この名詞句も文型パターン中の多くの N と置き換わる可能性があるからである。

(4) 数 字

プロトタイプ I では数字を単語としては認識しておらず単語への分解のプ

ロセスで切り捨てている。このため、次のような表題文では文法的にも誤った品詞列が割り当てられ、文型パターンとはマッチしない。

(例) Divided a digital signal by any digit from 1 to 9.

(5) 動詞を含む表題文

プロトタイプ I では助動詞，受身などの動詞に関する処理がないため，文型パターンとしては N V N などに帰着する表題文も訳出不可能となっている。

ここでこれらの原因のために訳出不可能となる表題文の概数を知るために，

表 3-1 プロトタイプ I による処理結果

	訳出可能	訳出不可能
建部氏の骨格パターンとマッチしない表題文		707
(1) 接続詞を含む文	393	
AND	383	
OR	10	
(2) ing を含む文	91	
(3) 過去分詞，不定詞	60	
過去分詞	40	
不定詞	20	
(4) 数字を含む文	2	
(5) 動詞を含む文	43	
助動詞，受動態	5	
その他	38	
(6) 辞書のミス	26	
(7) 表題文のパンチミス	14	
(8) その他でマッチしない文	56	

10,000表題文中の任意の2,000の表題文を例として調べた。その結果を表3-1に示す。この表に示された数値は、表題文2,000を状態遷移網、of-NOUNの処理（以下に述べる）を行なった後の品詞列を調べた結果である。したがって、表中(9)で示したプロトタイプIに用意した文型パターンにマッチするものは翻訳可能なはずである。しかし、実際に訳出されたのは、約1100の表題文である。このことから、断定はできないが約300程度の文がof-NOUNの処理によって、翻訳可能となると思われる。

3.5 プロトタイプⅡにおける改善点

本節では、3.4で述べたプロトタイプIの問題点に対しプロトタイプⅡではどのように改善したかについて述べる（プロトタイプⅡの全体の流れは、プロトタイプIとほぼ同じである）。

〔I〕ANDの処理

ANDを含む文を正確に翻訳することは、人間にとっても難しいことであり、文全体の意味を把握することが必要となる。現在使用している辞書の意味記述は、この処理に利用するには不十分であり、また文全体の意味の把握も困難である。そこで、本システムでは、品詞列のみを調べて処理して行く方針を取らざるを得ない。

まずANDの含まれた表題文を調べて見ると、次の5つの並列の仕方が存在する。

1. adj & adj, v & v, adv & adv
2. prep N & prep N
3. prep adj N & (det) (adj) N prep
4. N prep N & N prep N
5. 重文 その他

1の形はANDの前後の品詞が同一であり、単語どうしがANDによって接

続されている。この形をまず優先的に処理する。この処理はプロトタイプⅠでも形容詞の場合などに限って行なわれている。2の形は、前置詞句どうしが並列している場合である。この形ではANDの直後が前置詞となっており、容易に発見できる。

(例) Study of the electric conductivity of plasma of combustion products with seeding in the u02 MHD generator channel and on a laboratory installation.

3の形は最も多く表題文中にあらわれる形である。3と4は品詞列からだけでは判断することはできない。しかし、4の形が非常に少数であることを考え、本システムでは、3の形としてこのANDを処理する。次の文を例として、その処理結果を図3-8に示す。

(例) Effects of spectral cross relaxation and collisional dephasing on the absorption of light by ...

5の形についてはプロトタイプⅡの検討の項で説明する。現在は重文の処理は行っていない。

effect	of	spectral	cross	relaxation	on	the	absorption	of	light
1	2	3	4	5	9	10	11	12	13
n	prep	adj	n	n	prep	det	n	prep	n
0	0	0	0	0	0	0	0	0	0

relaxation	and	collisional	dephasing
5	6	7	8
n	conj	adj	n

図 3-8 ANDの処理例

〔Ⅱ〕 ing の処理

(1) 表題文での ing の用法

ingは現在分詞、動名詞の2通りの用法が考えられる。また、動名詞にはさらに名詞化してしまっているものもある。本システムの辞書では、名詞化したing(ある対象分野をかぎると名詞としてしか使われないing形)は名詞として登録する方針をとり、ここでは対象としない。

現在分詞は形容詞として働き、名詞を修飾するが、前置詞をこえた名詞を修飾することはない。動名詞は名詞と同等であるが目的語を取ることがある。しかし、この場合も前置詞をこえた名詞を目的語として取ることはない。以上の理由からingを前置詞および動詞を区切りとして考え、そのパターンについて調べると、次のような分布をしている。

a. (prep) N ing (prep)	55
b. (prep) ing N (prep)	113
c. (prep) N ing N (prep)	72
d. (prep) ing (prep)	10
e. (prep) det (adj) ing N (prep)	
f. ing N ing, ing ing N	9

eは先行する冠詞があるため状態遷移網によってすでに形容詞と同様の扱いを受け、処理されている。したがって、ingの処理ではa～dまでの形を処理している。上の数字は20000 文献中にあらわれたパターンの数である。どのパターンにおいても、現在分詞・動名詞の2つの用法は可能であるが、以下で述べるように実際には各々のパターンには特徴があり、用法に偏りが見られる。各々の解析には、この種の用法の偏りを考慮し、文法的チェックを考え次に意味的处理を行なっている。

(2) N ing

ingの後が動詞または文の終了の時には、ingは動名詞として用いられていることは明らかである。したがって、このingは品詞テーブル中で名詞に置き換えられる。しかし、ingの後に前置詞が置かれている場合は、文法的な関係だけからは決定できない。次に2つの例を示す。

例1 Editor facilities for small laboratory automation
computers operating in a virtual machine environ-
ment.

例2 Impedance matiching with lossy components.

例1では、in以下の前置詞句が副詞句としてoperatingを修飾し、operatingは現在分詞としてcomputersを修飾する。例2では、matchingは動名詞であり、with以下の前置詞句は形容詞句としてmatchingを修飾する。現在のシステムでは例2の用例が実際の標題文中には多いことから、これを標準的解釈としているが、意味関係をチェックするSUBJのチェック関数の実行により、Nがingの主語となっていると判断された時には例1の処理を行なう。

(3) ing N

この形はingの用法として最も頻度の高いパターンである。Nの先頭が冠詞がある場合には、Nはingの目的語となっている。

(例1) Modeling the bipolar transistor.

この場合ingは当然動名詞である。このような例は113例のうち18例あった。残りの表題文においてもNを目的語として取り動名詞として用いられているingは77個である。現在分詞として、形容詞的に用いられているのは10例である。残る8例はingが名詞化されていると考えられ、これらは辞書記述自体を変更すべきものである。そこで、現在分詞として使用されているingを見つけ出すために、プロトタイプIでは文型パターンとのマッチングの際に行なっていた意味的なチェックをここで行なう。ここで一例を挙げると、

(例2) measuring techniques

(例3) measuring temperature

使用するチェック関数は、SUBJとOBJである。この2つのチェック関数の実行の結果、後ろのNを主語として認識した場合には、ingは現在分詞とみなされ、語順はそのまま、ingを品詞テーブル中で名詞に置き換える。目的語として認識した場合は、語順を入れ換えてからingを名詞に置き換える。しかし、辞書の意味記述がまだ十分ではないため、どちらも認識しないことがある。この時には、標準的解釈として多数である例

3 の解釈を取る。

(4) N_1 ing N_2

この場合の ing は N_2 を目的語に取る時には現在分詞として N_1 を修飾することが普通であり、動名詞となることはまずない。実際、2000 文献中には存在しなかった。ing が動名詞ならば、それは ing だけで名詞化されている。したがって、次の 3 つの場合が考えられる。

1. ing が N_1 を目的語にとり、 N_2 を修飾する。

訳出順 N_1 ing N_2

(例) Product contamination monitoring technique.

2. ing が N_2 を目的語にとり、 N_2 を修飾する。

訳出順 N_2 ing N_1

(例) Integrals involving the Gaussian distribution function.

3. ing が名詞化している。

訳出順 N_1 ing N_2

(例) Upper bound on signal quatizing noise ratio for television d.p.c.m.

1 と 3 は判断をつけにくい、訳出順が同じであるため、2 の形のものを見つけることが問題となる。しかし、2 の形の ing は、数が少なく、また having, involving, using などは常に目的語を伴い、2 の形で用いられる。このような ing は前置詞と同等と考えられる。そこで、これらを前置詞として辞書に登録することにより、本システムでは、1 を標準的解釈とした。しかしながら、2 の形を取る ing でも having などのように、常に 2 の形として用いられるものばかりではない。これらをうまく処理するには (3) の場合と同様に意味的關係を参照する必要がある。

(5) ing

前置詞に挟まれた ing は文法的に明らかに動名詞と考えられるので、品

詞テーブル中で名詞に置き換えられる。

以上が現在行なっている ing の処理であるが、ing は元来動詞的な性格も持っているため、後ろの前置詞句は、副詞句として ing にかかることもある。〔Ⅶ〕でも述べるが、現在 ing の訳は名詞化して与えられているために（例 measuring 測定）、後ろの前置詞句は形容詞句として処理しても、訳出された日本文を見ると、ing に副詞的にかかっているように取ることができる。

以上5つのパターンにおいて ing を処理した後、品詞列を再び状態遷移網に通す。

〔Ⅲ〕 of - Noun の処理

文型パターンを簡単化するには、不必要な前置詞を減らすことが必要である。N₁ of N₂ という形において、N₂（状態遷移網によってまとめられた名詞句）が N₁ 以外にかかることはない。そこで、これを of を含む名詞句として N₁ に置き換える。図3-9に次の例文における of - Noun の処理を示す。

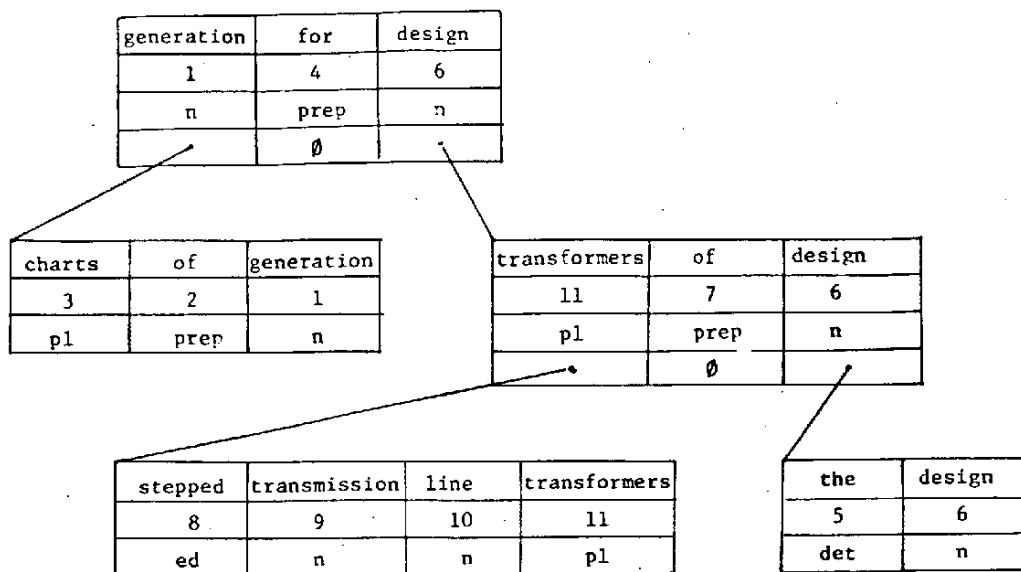


図3-9 of - Noun の処理例

(例) Generation of charts for the design of stepped
transmission line transformers.

of は頻繁に表題文に用いられているため、この処理によって表題文の文型パターンがかなり簡単化されることになる。

[IV] 動詞の処理

(1) 不定詞の処理

N_1 prep V N_2 という品詞のつながりを品詞列中に見い出した時に前置詞が "to" であれば、不定詞句として処理する。前置詞が "to" 以外であれば、この動詞は名詞として用いられていると考え、名詞に置き換える。本システムではこの不定詞を形容詞的用法と考え、prep V を前置詞に置き換えている。ここでは同時に日本語訳の処理も行ない、 N_2 を目的語として取るために V に助詞「を」を付加し、"to" を連体形の「ための」とする。不定詞にはこの他に名詞的用法、副詞的用法があるが、表題文ではほとんど用いられない。

(2) 助動詞

助動詞としては主に can を対象として処理を行なった。このためにまだいくつかの問題点が残っている。助動詞は動詞の次に訳出されるものとして処理するが、日本語文の生成のとき、can と might では動詞の語尾は変化するため、動詞の語尾「する」の扱いをどのようにするかが問題となる。

(3) 受動態

V ed prep という品詞の並びが存在する時にこれを受動態として処理する。

(例) Calculator parts are integrated on a single glass
substrate.

この時の on 以下の前置詞句は副詞句として動詞にかかる。そこで on に

は、副詞句を示すコードが与えられる。システムの処理は V (are) を取り除き ed を V に変え、生成用訳語の語尾「される」を付加する。

〔V〕 過去分詞

過去分詞の用法としては受動態、完了形などの動詞として使われる場合と、形容詞的に使われる場合がある。ここでは、形容詞的に用いられている過去分詞について述べる。

(1) 状態遷移網で処理される過去分詞

前置詞を伴わず、ed が一つの形容詞と全く同じと考えられる場合である。次にその例を示す。

(例) Noise measurements on the floating diffusion input
for charge coupled devices.

(2) 前置詞句を伴う過去分詞

次に示す文を例として考える。

(例) Minimum prediction residual principle applied to
speech recognition.

この例文中の ed は状態遷移網では処理されない。ここで対象とするのは、このような前置詞句を伴う過去分詞である。この例文に対する処理を図 3-10 に示す。この過去分詞は前置詞句を伴い、直前の名詞を修飾すると考え、ed prep を 1 つの前置詞に置き換える。過去分詞の伴う前置詞句は副詞句である。そこで to には副詞句を示すコードが与えられる。to の辞書記述には連体修飾、連用修飾の 2 つの場合に応じて「への」、「へ」という 2 つの訳が与えられており、このコードによって訳出時に「へ」という訳を選択する。

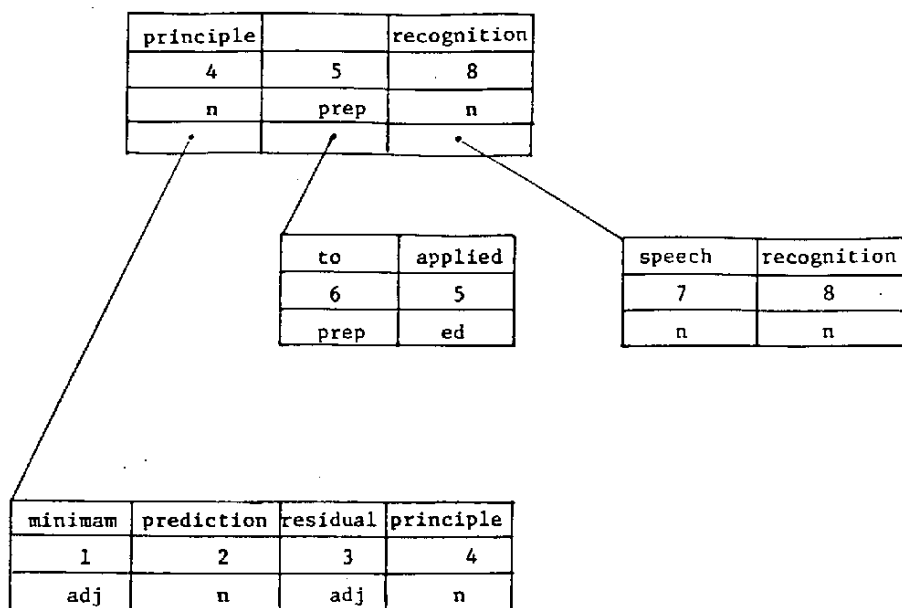


図 3 - 1 0 前置詞句を伴う過去分詞

〔VD〕 冠 詞

冠詞の持つ情報を利用するために、この処理を行なう。1つは、動詞と名詞をもつ多品詞語のチェックである。det V という形では、明らかに名詞となるべきである。この時には、Vに名詞であることを示すコードを与え、訳出時に名詞の訳を用いる。

3.6 プロトタイプⅡの評価

本節では10,000文献の任意の2000の表題文をプロトタイプⅡによって訳出した結果について検討する。訳出不可能であった表題文、訳出されたが誤訳あるいは意味の不明である表題文、意味は理解できるが日本語としてなじまないもの、の3段階に分け、以下順に説明する。分析結果を表3-2に示す。

表 3 - 2 プロトタイプⅡによる処理結果

訳出不可能					
	システムの能力外	60		93	
	ing の処理不可能	17			
	多品詞語	8			
	辞書記述の誤り	8			
誤記及び意味不明な文					
	解析誤り				240
		AND	21	99	
		Ing	24		
		文への分解	16		
		その他	38		
	多義語				
		前置詞	29	71	
		名詞・その他	42		
	辞書の問題				
		記述の誤り	38	70	
		熟語の未登録	32		
日本語生成の問題					
	日本語の語尾生成	51		66	
	専門分野による訳	15			

〔I〕 訳出不可能な表題文

- a. 現在のシステムでは対応する処理を用意していないために翻訳の不可能な文。関係詞を含む文，骨格パターンにまで副詞の残る文，などが挙げられる。

(例) Test socket for use where electrical interferences

is critical.

b. ing の処理ができなかった表題文

ing の処理で処理できない ing は骨格パターンまで残ってしまうため、用意した文型パターンにマッチせず、訳出されない。

(例) A system for ultrasonically imaging the human heart
in three dimensions.

c. 多品詞語を含む表題文

主な多品詞語を表 3-3 に示す。多品詞語を根本的に解決するには、文全体の意味を知らなければできないが、文法的制約によって処理できる場合もある。本システムでは動詞と名詞を持つ increase などを対象にこの処理を考えた。しかし、本質的な多品詞語（同じような割合でいくつかの品詞として用いられるもの）は、文法的チェックだけでは処理できない。

d. 辞書記述の誤り

辞書に登録されている品詞が誤っているため、文法的に誤りのある骨格

表 3-3 主な多品詞語の例

use	n,v
controle	n,v
supply	n,v
increase	n,v
test	n,v
aid	n,v
near	prep,adj
inside	prep,adj
below	prep,adj
current	n,adj

パターンとなってしまう表題文がある。

e. 表題文のパンチミス

表題文自身に文法的な誤り、またはパンチミスがあり骨格パターンにマッチしない文。

〔II〕 誤訳および意味不明な訳

a. 解析の誤り

ANDの処理、ingの処理システムの解釈が誤りであるために、訳文が誤訳または意味不明となってしまった表題文

a.1 ANDの解析が間違っている訳。これについては、〔IV〕項で説明する。

a.2 ingの解析誤り。これは辞書中に意味的チェックが有効に働くためのデータが不足しているために、ingの標準解釈が行なわれたが、それが誤りであった表題文である。

(例) Comment on a ceramic monopole sound source for
measuring purposes.

この種の誤りは名詞の意味的属性を辞書に記述して行くことによって、解消される。

a.3 文の分解における誤り

(例) Laser resonators with polarizing element engenstates and engenvalues of polarization.

上の例では、elementとengenstatesの間で文を分解することができず、誤った訳が出されている。

b. 多義語を含む表題文

b.1 前置詞

英語の前置詞は本質的に多義語であり、1つの日本語を対応させることは困難である。しかし、本システムが対象としている技術論文の表題文では用法に偏りがあることを考慮し、1つの訳を与えている。このた

め意味不明な訳出が出力されることがある。ここでは with について考えて見る。with には次の3つの用法を表題文中に見ることができる。

- ① An ultrasonic echo pattern with a mechanical control for the alternate echoes.
- ② A learning system with the ability to grasp its situation.
- ③ Method for measuring magnetic moments with precision.

①の訳は「による」であり、手段を示す。辞書にはこの訳が記述されている。②は所有を示す with で「をもつ」と訳されるべきだろう。③は、正確「な」と訳されるが、この例では with は副詞句を形成し、measuring を修飾する。

①と②の訳し分けは意味処理を必要とするが、③は名詞の属性を参照することによって可能となるのではないかと思われる。

with 以外では、on が「について」、「上の」と2つの訳が表題文では考えられる。

b.2 名 詞

名詞にも多義語は存在する。主な多義語を表3-4に示す。この種が多義語を選択するには、やはり意味解析が必要となる。現在、本システムでは多義語にはきるべく一般的であると思われる訳を与え、その他は、なるべく広い範囲を参照するイディオムとして処理していく方針をとっている。

b.3 その他

以上の他に辞書引きのためのインデックステーブルがすべて小文字に統一されているため、前置詞としての as と化学記号の AS 等の区別がつかない。これらは単語としては異なるがスペルが同じために、多義、多品詞語となってしまう。

c. 辞書の問題

単純な辞書記述の誤りとイディオムの未登録がある。イディオムとして専門用語が登録されていないために、意味が不明な訳となることは当然考えられる。また訳の向上にもイディオムの登録は必要であろう。

表 3-4 主な多義語の例(名詞)

with	～による, ～をもつ
on	～についての, ～上の
solution	解, 溶液
function	関数, 機能
plasma	プラズマ, (人の)血しょう
field	場, 分野
factor	率, 要因
discharge	放電, 放出
no	ない, ナンバー, 一酸化窒素
as	～としての, AS
physical	物理的, 生理学的
cutting	切断, 減少

(III) 日本語の生成の問題

a. 専門分野によって訳をかえる必要のあるもの

文全体から考えて単語の持つ意味以上に, その分野に相当と思われる訳をつけなければならない場合がある。イディオムとして登録できる例もあるが, 次のような例においてはイディオムとしての登録はできない。

(例) Facsimile compatibility with different resolutions.
resolution には「分解」という訳が与えられているが, この例では「分解能」と訳されるべきである。

b. 日本語の単語の接続の問題

日本語の単語間に、助詞を置かなければならない場合、接続する語によっては訳を省略したり変更する必要がある場合などが考えられる。

(例) Food crushing sound

食品(を)つぶす音(計算機出力では「を」が欠落する)

(例) Computation of co laser radiation absorption in air
boundary layers.

空気境界層での一酸化炭素レーザー(放射)吸収の計算

[IV] ANDを含む表題文

この節ではANDを含む表題文だけを取り上げ、プロトタイプIIによって、どの程度まで訳すことができたかを、ANDの訳し方に注目し調べた結果を説明する。結果を表3-5に示す。

a. 3.5の[I]で述べたAND形は、本システムでは3の形として処理されるため、誤った訳出がされる。その例を次に示す。

(例) Power generating waterways in the German federal
Repubric and navigation.

b. 動詞は含んでいないが、重文的構造を持ちANDで切れる文がある。普通は次の例のように、代名詞がANDの直後に置かれている。

(例) Side looking radar systems and their potential app-
lication tv earth resources surveys.

c. ing がANDで並列している名詞句に含まれている場合

(例) Optimizing system and equipment design.

d. 前置詞句を伴う過去分詞が並列している名詞句に含まれている場合

(例) X ray ionization cross sections, and ionization
equilibrium equations modified by auger transitions.

e. both ~ and ~ などのような、flexibleな熟語を含む文

(例) Nomal mode program with both the discrete and bran-

ch line contributions.

以上の5つの誤った解析が頻度の高かったものである。これらはいずれも品詞列からだけでは判断できず意味解析が必要となる。

本システムにおける翻訳結果を付録にまとめた。

表 3-5 ANDを含む表題文の処理結果

a. 前置詞を含む名詞句の並列		12
b. 重文的なもの		17
代名詞を伴う	13	
その他	4	
c. ingを名詞句に含んでいる文		5
d. 過去分詞を含んでいる文		3
e. flexibleな熟語と考えられる文		2
f. その他		7
計		46
g. 訳出不可能だった文		50
計		96

対象としたANDを含む表題文の総数

492

3.7 結 論

プロトタイプⅠでは、なるべくすべての現象を文型パターンのレベルで処理するという方針をとり、逆にプロトタイプⅡでは of-NOUN 処理, ing 処理等局所的に処理できる構造は文型パターンに至るまでに解消するという方針に従った。この結果、プロトタイプⅡにおいては、表題文の約9割を12個の文型パターンで訳出することができた。しかし、訳の質を考えて見ると、プロトタイプⅡにおいても約2割の訳が誤訳または意味不明な訳となっている。これ

らの表題文を、どのようにして正しく訳出するかが、まず第1の問題点である。結果の分析からも明らかなように、最も有効な手段として、専門的イディオムの登録が考えられる。この作業は簡単ではないが、データの収集を行なうことによって可能である。同様に意味的なチェックのための名詞の属性を辞書に登録することも必要であろう。多義語、多品詞語に関しては、相当に深い意味処理が必要であり、現在、有効な方法は考えられず、研究の行なわれるべき問題である。

本研究では試験的にANDの処理を行ない、その結果をまとめたが、表題文におけるANDの処理としてはかなり良い結果になっている。

第2の問題点は日本語生成である。本システムでは系統的な日本語の生成を行っていない。日本語文法にそった、語尾の処理や数詞の処理を行なうことによって、訳文の読み易さを飛躍的に向上させることが可能であろう。

第3に訳出不可能な表題文を考えるが、辞書のミスや表題文のパンチミスを別として、非常に複雑な原因による場合が多い。したがって、これらの少数の表題文のために、多くの処理を考えなければならない。その中でも、特に必要と考えられる処理として次に挙げるものがある。

- ・ 副詞の処理
- ・ 関係詞節の処理
- ・ 重文の処理

これらはいずれも、自然言語処理には当然必要であるが、本システムが表題文という限定した分野を対象としていることから考えると、これらの処理は放棄してPost - editing 等の人間の介在にまかせるのも1つの方法である。現在のシステムは、プロトタイプⅠ・プロトタイプⅡともに実験システムとしての性質が強く、今後pre - editing, post - editing を含めた人間の介在をどうするか、また専門分野に限った場合に、どのようにして系統的にその分野における専門用語あるいは専門的なイディオムを収集してゆくか等の問題が残されている。

付録：表題文の翻訳結果

- 1001
REAL TIME UPDATING OF MAXIMUM USABLE FREQUENCY PREDICTIONS FOR HF RADIO COMMUNICATION.
N. ING
H F無線通信のための最大使用周波数予測の実時間最新
- 1002
UNIMOMENT METHOD OF SOLVING ANTENNA AND SCATTERING PROBLEMS.
アンテナ及び散乱問題解決のユニモーメント手法
- 1003
RADIO AND LONG HAUL TRANSMISSION.
無線及び長距離伝送
- 1004
FURTHER STABILITY RESULTS OF A DELAY DIFFERENTIAL SYSTEMS FOR DIGITAL NETWORKS.
ディジタルネットワークのための遅延微分システムのより進んだ安定性結果
- 1005
TONELESS CARRIER RECOVERY.
無音キャリア回復
- 1006
MODEM WRAP TEST APPARATUS AND METHOD.
モデム巻き試験装置及び手法
- 1007
A SIGNALING SCHEME FOR 4 PSK TRANSMISSIONS WITH SEQUENTIAL DECISION FEEDBACK.
順序決定フィードバックによる4PSK伝送のための信号方式
- 1008
ADAPTIVE QUANTIZING IN SPEECH CODING SYSTEMS.
音声符号化システムでの順応する量子化
- 1009
E TELEMETRY:
E 遠隔
- 1009
INSIDE STORY OF CENTRALIZED MAINTENANCE.
YAKUZI 4.51: 3: 保守の話
- 1010
SOME RESULTS CONCERNING THE PERCEPTION OF MUSICAL DISTORTION IN MISTUNED SINGLE SIDEBAND LINKS. NOT MATCHING N N
PREP N
- 1011
AN APPROACH TO ESTIMATING LAND MOBILE RADIO COVERAGE.
推定陸移動無線区域へのアプローチ
- 1012
DESIGN TECHNIQUE FOR A NEAR FIELD COMMUNICATION SYSTEM.
近くの場通信システムのための設計技術
- 1013
OUTAGE OF THE L 4 SYSTEM AND THE GEOMAGNETIC DISTURBANCES OF 4 AUGUST 1972.
4 月 19 日 72 の L 4 システム及び地磁気じょう乱の停電
- 1014
RESULTS OF SUBURBAN BASE STATION SPATIAL DIVERSITY MEASUREMENTS IN THE UHF BAND.
UHF帯域での郊外ベース基地空間ダイバーシティ測定の結果
- 1015

SPACE DIVERSITY ENGINEERING.

N I N G

空間ダイバーシティ工学

1016

PROPERTIES OF MOBILE RADIO PROPAGATION ABOVE 400 MHZ.

400メガヘルツ以上の移動無線伝搬の性質

1017

OPTIMAL CHANNELIZATION IN FDMA COMMUNICATIONS.

FDMA通信での最適チャネル化

1018

MILITARY SATELLITE COMMUNICATIONS USING SMALL EARTH TERMINALS.

小さい大地端末を用いた軍事衛星通信

1019

PCM TDMA SATELLITE COMMUNICATION EQUIPMENT.

PCM TDMA衛星通信装置

1020

REDUNDANCY CONFIGURATION AND SOFTWARE FOR PCM-TDMA SATELLITE COMMUNICATION SYSTEM.

PCM TDMA衛星通信システムのための冗長構成及びソフトウェア

1021

REFLEXION PHENOMENA IN COAXIAL DIGITAL LINE LINKS.

同軸デジタル送電線リンクでの反射現象

1022

RURAL SUBSCRIBER LOOPS VIA CARRIER AND RADIO.

農村加入者ループVIAキャリア及び無線

1023

CANTAT 2 SUBMARINE CABLE SYSTEM:

CANTAT 2 海底ケーブルシステム

1023

OVERALL PROJECT.

一般計画

1024

CANTAT 2 SUBMARINE CABLE SYSTEM:

CANTAT 2 海底ケーブルシステム

1024

SUBMERGED REPEATER MONITORING EQUIPMENT.

水中中継器監視装置

1025

CANTAT 2 SUBMARINE CABLE SYSTEM:

CANTAT 2 海底ケーブルシステム

1025

POWER FEEDING EQUIPMENT.

電力給電装置

1026

EXCHANGE AREA AND LOCAL LOOP TRANSMISSION.

交換面積及び局所ループ伝送

1027

L 5 SYSTEM:

L 5 システム

1027

OVERALL DESCRIPTION AND SYSTEM DESIGN.

一般記述及びシステム設計

1028

LS SYSTEM:

LS システム

1028

JUMBOGROUP MULTIPLEX TERMINAL.

ジャンボグループ多重化端末

1029

ENHANCEMENT OF BANDWIDTH AND ADDITIONAL LOSS CAUSED BY CONTROLLED MODE COUPLING IN QUARTZ FIBRES.

N ING

石英繊維での制御モード結合による帯域幅及び添加損失ひきおこされるの増大

1030

RESOLUTION OF GAUSSIAN IMAGES ON A PHOTOELECTRICALLY EMISSIVE SURFACE.
光電放射表面についてのガウスイメージの分解

1031

A THEORY OF OPTIMUM PHOTON COUNTING DETECTORS.

最適光子計数検出器の理論

1032

AN APPROACH TO DIGITAL RECORDING BY LOW COST AUDIO CASSETTE.

N ING

低コストオーディオカセットによるディジタル記録へのアプローチ

1033

COMPANDING CODE.

符号圧伸

1034

DELAY TRAJECTORY CONTROLLED SWITCHING.

遅延軌道制御切替

1035

BUILD COMPACT MODEMS INTO DIGITAL EQUIPMENT.

ディジタル装置へのコンパクトモデムを製作する

1035

AN LSI CIRCUIT DOES MOST OF THE SIGNAL HANDLING, EXCEPT FOR A FEW EXTERNAL ANALOG CIRCUITS. RI FOUND

11

NOT MATCHING N V ADJ PREP N PREP N

1036

HYBRID ROUTING TECHNIQUE FOR PACKET SWITCHED NETWORKS.

パケット切替ネットワークのためのハイブリッドルーティング技術

1037

EXPECTED CROSSTALK PERFORMANCE OF ANALOG MULTICHANNEL SUBSCRIBER CARRIER SYSTEMS

アナログ多重チャネル加入者キャリアシステムの期待漏話性能

1038

ELECTRICAL DESIGN OF METACONTACT EXCHANGES.

メタコンタクト交換機の電気設計

1039

THE NETWORK:

ネットワーク

1039

FORGING NATIONWIDE TELEPHONE LINKS.

鍛造全国電話リンク

● 4. 計算機マニュアルの
日英機械翻訳システム

●

4 計算機マニュアルの日英機械翻訳システム

前章では、科学技術論文の表題というかなり限定の強い対象を選んで機械翻訳システムを構成し、そこでの問題点を整理した。本章では、さらに本訳対象をひろげて一般の文章を対象とした場合に、どのような問題点があり、現在の技術でどの程度の翻訳システムが構成できるかを、実際に構成してきたシステムを基礎に明らかにしてゆこうと思う。

我々は現在、本格的な日英機械翻訳システムの実現を目指して、システムの作成を行なっている。現在の翻訳対象は、「計算機マニュアル」(富士通OS IV「データセットユーティリティ」)という比較的限定された語彙と曖昧さの少ない分野のテキストを選んでいますが、このような限定された対象においても、機械翻訳システムを実用するためには解決しておかなければならない種々の問題がある。とくに、日本語と英語という、本来語族が全く異なる言語を対象としているために、これまでヨーロッパ等で行なわれてきた英語・仏語といった語族として近い言語間の機械翻訳にはみられなかった問題も数多く生じることになる。

4.1 システムの構成

現在我々が開発中の機械翻訳システムの全体的な構成を図4-1に示す。

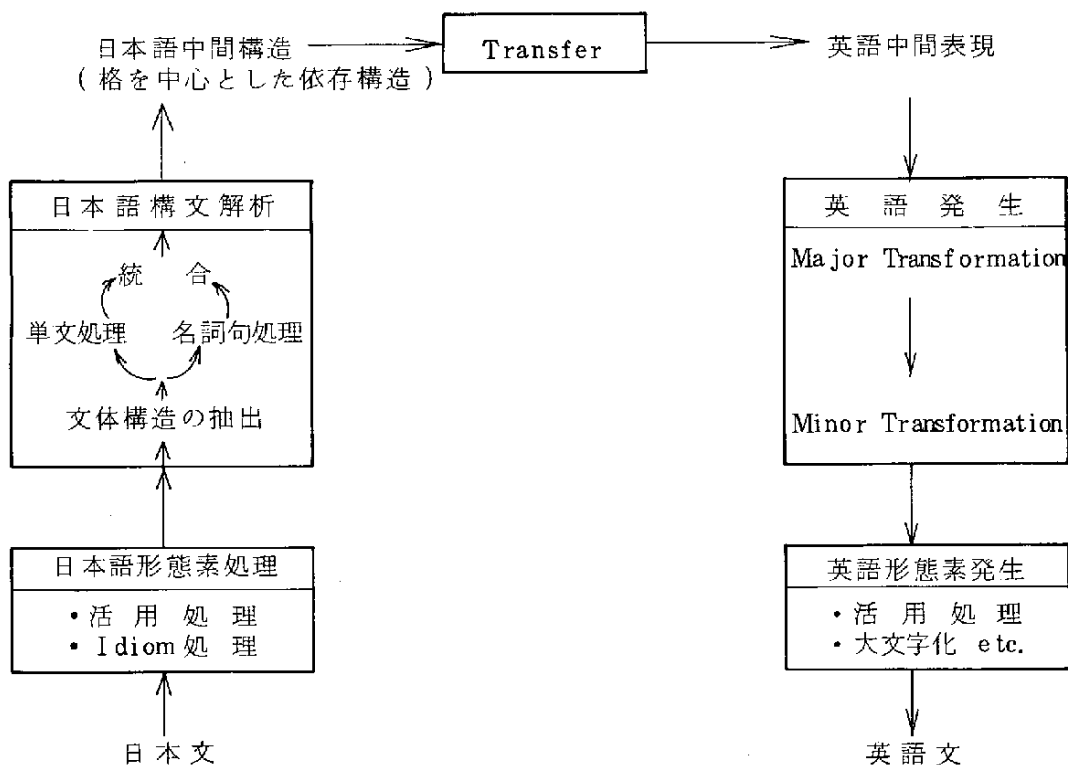


図 4-1 機械翻訳システムの構成

この図に示されるように、我々のシステムはカナダの TAUM やフランスの GETA のグループが主張する Transfer アプローチに従った機械翻訳システムである。したがって、機械翻訳のプロセスは、

日本語文 → 日本語中間表現 → 英語中間表現 → 英語文
 ① (格構造を中心とした依存構造表現) ② (句構造表現) ③

という 4 つの構造間の変換プロセスになっている。①が、原語 (Source Language - S L) の解析、②が両言語間の Transfer、③が目的語 (Target Language - T L) の発生の各過程である。①、③は、それぞれ形態素解析 (発生) と統語 (あるいは意味も含めた) 解析 (発生) の段階に細分できるとになる。

一般的に言って、このシステム構成において、S L、T L 両語間の語族が近

い場合には、②の Transfer 過程は単なる語彙挿入 (bi - lingual dictionary を参照して、S L の単語を対応する T L の単語に置き換える) と、それに伴う若干の統語構造の変換を行なうだけで十分である。Transfer アプローチにおいては、それぞれの言語に対する中間表現がその言語の特徴を強く残していることを前提としていること、また英語と日本語の言語としての構造が大きくく違っていることから、我々の日英機械翻訳システムにおいては、この Transfer の過程で、かなり大きな構造変換を行なわなければならないことになる。

日英機械翻訳システムを Transfer アプローチで構成する場合には、一般的に次のような問題が生じる。

① 両言語の距離が遠いために、Transfer がかなり深いレベルの中間表現形に対して実行されることになる。このため、S L 解析、T L 発生は、それぞれ表層表現からかなり離れた中間表現への解析、あるいは、中間表現からの発生を行なわなければならない。より具体的には、T L 発生において、passivization, relativization のような大きな構造変換を含む各種の変形規則をステップワイズにうまく適用していかなければならない等の問題が生じる。

② S L 解析の結果、得られる中間表現のレベルが、意味的に深くなればなるほど、Transfer アプローチから Interlingual アプローチへ近づいていくことになるが、どこまでの処理を S L 解析での処理とし、どこから以降を Transfer Component あるいは S L - Synthesis Component の役割とすべきかに任意性が生じる (すなわち、中間表現形にどの程度 S L 言語自体の言語的特徴を残し、逆にどの程度 Universal にするか)。同様の問題は Transfer Component と T L - Synthesis の役割分担の際にも生じる。

以上、2 点はかなり一般的な問題点の指摘であるが、以下の各節において、より詳細にこれらの問題を考えることにする。

4.2 日本語および英語の中間表現

本節では、我々のシステムで採用している日本語中間表現および英語中間表現の仕様を述べ、中間表現による日本語文、英文の表現例を示し、その問題点について考察する。

4.2.1 中間表現の形式

機械翻訳に限らず、計算機による自然言語処理一般について、文の構造や意味をどの様に表現するかは重要な問題である。特に機械翻訳では、データベースの自然言語応答システム等の応用に比べるとはるかに幅の広い言語表現を対象とするため、各処理過程で適切な表現形式をとらなければ、システムに入力できる S L や、システムから出力できる T L の範囲が限定されたり、処理が困難になったりする。

言語学の分野では、各種の文法に応じて、次の様な、文の構造の記述体系が提唱されている。

- (1) 句構造表現 (変形文法, 生成意味論)
- (2) 依存関係表現 (依存関係文法)
- (3) 論理式表現 (モンテギュー文法)

句構造表現は、単語を名詞句 (N P) , 動詞句 (V P) といった高次のまとまりに整理してゆくことを基本にしており、英語や仏語の様に語順に関して比較的強い制約のある言語の構造を表現するのに適している。依存関係表現は、単語の支配、被支配の関係を構造化する表現であり、日本語のように比較的語順に関する制約の少ない言語の構造を表現するのに適している。論理式表現は、各単語を論理式における述語 (Predicate) としてとらえて記述するものである。これらの表現は、いずれも翻訳に必要な言語情報の記述という点で必ずしも十分でないため、各種の機械翻訳システムは、このいずれかを基本としながらも、それぞれ独自の表現を使っている。トランスファー方式の利点は、S L

と T L の中間表現を設計する際に、一応相手側の表現形式とは独立して、その言語に適した中間表現を選べることである。我々の対象とする日本語と英語を比較すると、

- (1) 英語においては省略することが文法的に不可能であるような要素が日本語では頻繁に省略される。
- (2) 日本語では、連用修飾句間の、あるいは連体修飾句間の語順の入れかえが比較的自由である。そのかわりに、係り受け関係を明示する格助詞等が表層表現上にあらわれる。
- (3) 時制・様態・アスペクトの表現形態が両言語において全く異なる。

などの言語体系としての違いが大きい。このため、我々は日本語中間表現としては格関係を中心にした依存関係表現、英語中間表現としては英文生成過程において適用される変形規則の定式化が容易な句構造表現をそれぞれ基本として設計を行なった。この両中間表現間の相違は、中間表現間の変換 (Transfer) の段階で調整されることになる。

4.2.2 日本語中間表現

本システムでは、日本語中間表現として依存関係表現を用いているが、単なる単語間の依存関係を構造化するだけでは、翻訳を目的とした中間表現としていくつかの点で不十分である。以下では、依存関係表現の問題点を示し、我々の行なった表現の拡張について述べる。なお、以下からは日本文に現われる単語と日本語中間表現に現われる語を区別するため、前者を単語、後者を中間語と呼ぶ。

(1) 格関係の表示

翻訳を目的とした場合、依存関係表現では不十分である例を次に示す。

(図 4-2)

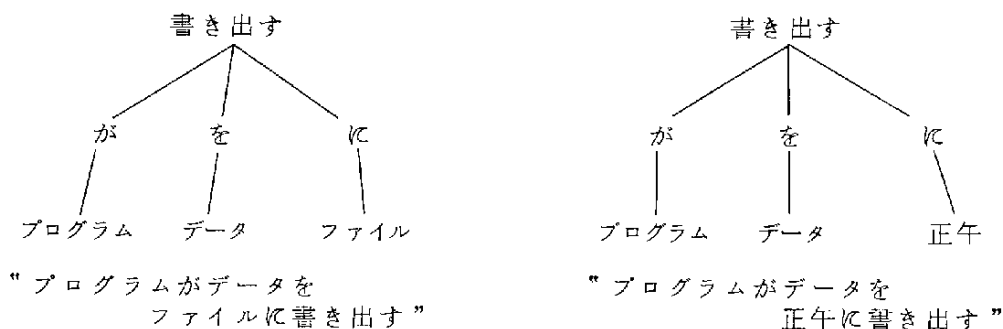


図 4-2 依存構造表現の例

(例文) a. プログラムがファイルにデータを書き出す。

b. プログラムが正午にデータを書き出す。

a, b の両方に同じ格助詞 'に' が現われているが, a と b ではその働きが異なっている。a では, 'に' は書き出すという動作の目標を示しているが, b では, 動作の起こる時間を示している。この動詞に対する名詞の関係を格と言うが, 依存関係表現では, 名詞の格を区別して表現することができない。

C. J. Fillmore は, 格関係を整理し, 深層格と表層上の文法関係の概念を用いて例文の様な言語現象を説明する格文法 (Case Grammar) を提唱した。格文法によれば, '～に' は a と b では表層表現として同じであるが, 深層格関係では異なっており, 区別される。

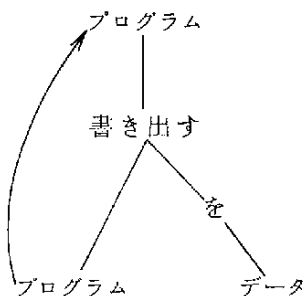
我々の日本語中間表現では, 例文 a, b は図 4-3 に示す構造になる。動詞の必須格は木構造中の位置で, 任意格は格の種類を示すマーカを付加して表現する。格文法では, 全ての格にマーカを付けて示すが, 我々は, 全ての動詞の格をカバーするような統一的な格関係の集合を規定することが困難であり, 各動詞が固有の意味的關係で名詞を統合するという立場から必須格には格マーカを付加していない。また, 格関係を導入したことは, 日本語中間表現がより意味的表現となっていることに対応し, 構文解析はより深くなっていると言える。



図 4-3 日本語中間表現の例

(2) 関係記述語の導入

依存関係表現は、基本的に日本文の表層に現われた単語だけを使った構造の表現であるため、単語として現われない連体修飾関係や連用中止法を表現する場合に非常に不自然な形になる。例えば、“データを書き出すプログラム”という連帯修飾節を含む名詞句は、図 4-4 に示す構造になる。我々のシステムでは、両言語間の構造変換を、日英両言語間の構造対応を記述した二言語間辞書（以下では構造変換辞書と呼ぶ）を手掛りに行なう。このために、日英両言語で異なった表現形態をとるものについては、日本語中間表現中に節点として明示的に表現され、この節点につけられた中間語についての構造変換辞書中に変換用の規則が記述されているという構成が望ましい。したがって、我々は「埋め込み」、「連用中止」、「名詞修飾」等、日本語の単語としては現われないが、構造変換の段階では重要な役割を果たす文法的関係を、関係記述語として明示的に中間表現中に表現することにした。したがって図 4-4 の依存関係表現は、我々のシステムでは、図 4-5 のように表示される。



‘データを書き出すプログラム’

図 4-4 埋込み文の依存関係表現

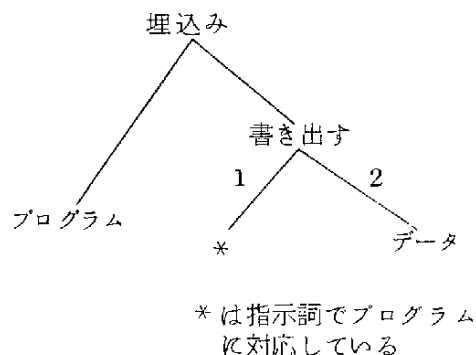


図 4-5 関係記述子を用いた
日本語中間表現

(3) 単語の細分化および慣用句

自然言語一般に、1つの単語が使われ方により異なった意味を表わす場合が少なくない。このような意味の違いは、構文解析過程で明らかになる。日本語中間表現では、表層で同じ単語も、構文解析過程で異なった意味に解釈されれば異なった中間語として表現する。また、日本語の特徴として形容詞、形容動詞などは、活用形によって単語の働きが変わる。例えば“立派だ” “立派に” “立派な” は、述語、連用修飾語、連体修飾語とそれぞれ働きが異なる。

逆に翻訳の立場からは、複数の単語を1つのまとまりとして処理した方がよい場合もある。例えば、“ことができる” “なければならない” などは意味解釈が一意に決まり、中間表現においては1つの中間語で表現する（図 4-6）

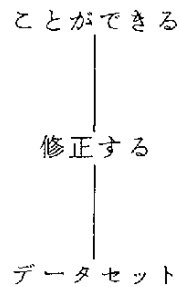
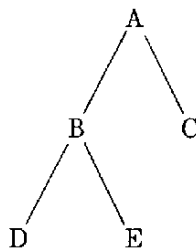


図 4-6 モーダルの表現の中間表現

(4) 日本語中間語の分類

中間語は、木構造の位置関係から支配語あるいは被支配語のいずれかになる。支配語とは、日本語中間構造を木で表わした場合に、下位に中間語を持つ語であり、被支配語とは、下位に中間語を持たない語である（図 4-7）。

我々は、中間語を、対応する単語の国文法における品詞、依存関係、意味関係を考慮して7つのカテゴリーに分類した（表 4-1）。しかしながら、この分類はあくまで便宜上のものであり、表中のカテゴリーは、さらに細かく分類されている。以下では、日本語の中間表現について各論的に説明してゆく。



支配語：A B

被支配語：C D E

図 4-7 支配語と被支配語

表 4 - 1 中 間 語 の 分 類

大分類	日本語の文法カテゴリとの対応	位置パラメタ	中分類	中間語の例		
ME	名詞, 代名詞等	NOGOV	NET	NOUN ENTITY	DATASET, SHOSAI, RIYOSHA	/
			PET	PROPOSITION ENTITY	KOFE, DORE, NANI	
SU	数 詞	NOGOV	NUM	NUMBER	0, 1, 2...	
JU	動詞, 形容詞, 形容動詞等で文の後となる語	GOV	PRD	PREDICATE	SHUSEISURU, NOKORU, DEARU HAYAI5	S
JO	助動詞等の助述表現をつかさどる語	GOV	KOS	KIND-OF-SENTENCE	GIMON, HITEI	S
			MOD	MODAL	DEKIRU, TEYOI	S
			ASP	ASPECT	KIGEN, SHIDO	S
			TNS	TENSE	MIRAI, KANRYO	S
SZ	接続詞等の接続関係を表わす語	GOV	S		TAME5, TOKI5, SHIKASHI	S
			CNJ	CONJUNCTION	OYOBI, MATAWA, NO-	
SH	形容詞, 形容動詞, 副詞等の他の語を修飾する語	NOGOV	EMD	ENTITY-MODIFIER	ATARASHII2, KIREI2	/
			PMD	PREDICATE-MODIFIER	HAYAI3	
			SMD	SENTENCE-MODIFIER	TOKIDOKI	
			MMD	MODIFIER-MODIFIER	IJO, IKA, HIJO	
KJ	日本文の表層には表われないが各種の言い回し, 用法を示す語	GOV	USG	USAGE	QUALIFY, QUANTIFY, PART UMEKOMI1~3, RENYOCHUSHI SHIEKI, UKEMI	
			CSS	CASE-SPECIFIER	LOC1~LOC3	
		NOGOV	IDN	IDENTIFIER	*	

(1) カテゴリー ME・SU

このカテゴリーの中間語は全て被支配語であり、MEは国文法における名詞、代名詞に対応する。SUは数字であり、数を表わす中間語‘数’または序数を表わす中間語‘序数’のいずれかによって支配される。

(2) カテゴリー JU

このカテゴリーの中間語は全て支配語であり、国文法における動詞、形容詞、形容動詞および‘である’‘だ’などがある。このカテゴリーの中間語は、各々固有の必須格を持っている。必須格は中間構造の位置に対応づけられているため、必須格が表層に表われていない場合には、空であることを示す中間語‘NIL’を入れる(図4-8)。また、カテゴリーJUの中間語は、任意格要素を支配する場合もある。任意格を示す中間語を表4-2に示す。

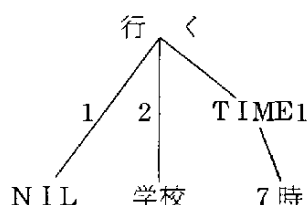


図4-8 スロットにNILのはいる例

(3) カテゴリー JO

このカテゴリーの中間語は全て支配語であり、文の種類、様態およびアスペクトを表現する。これらは、国文法の助動詞等に対応する。様態以外のカテゴリーJOの中間語は、単語のつづりを使わないで意味を表わすつづりにしている。特に、時制、アスペクトに関しては、金田一春彦編「日本語のアスペクト」を基本に分類している。

表 4 - 2 任意格を示す中間語

格の名前	中間語	例 文	格を表わす慣用表現
道 具	INST	カメラで美人を撮影する	を用いて
目 的	PURPOSE	彼に面会に <u>来</u> た 友達と遊びに <u>出</u> かける	た め
原因・理由	REASON	病気で休むかもしれない 運転手の不注意 <u>から</u> 惨事になる この作品で芥川賞を <u>獲</u> 得した	た め
手 段 方 法	WAY	飛行機で <u>旅</u> に出る	によって
目 標	ACCUS	父に本を <u>買</u> う	
場 所	LOC 1	日本で万博が <u>行</u> なわれた	
起 点 場 所	LOC 2	学校 <u>から</u> まだ帰らない	
終 点 場 所	LOC 3	京都に <u>着</u> いた 家へ <u>帰</u> る	に対して
経 過 場 所	LOC 4	窓 <u>から</u> ゴミを捨ててはいけない	
時 間	TIME 1	封鎖を十二日 <u>に</u> 解除した	
起 点 時 間	TIME 2	午後 1 時 <u>から</u> 会議を行なう	
終 点 時 間	TIME 3	六時 <u>まで</u> 待つ	
経 過 時 間	TIME 4	1 日を日曜大工で <u>過</u> ごす	
付 随	ACCOM	父親も日曜日には子供と <u>遊</u> ぶ	と 共 に

(4) カテゴリー S Z

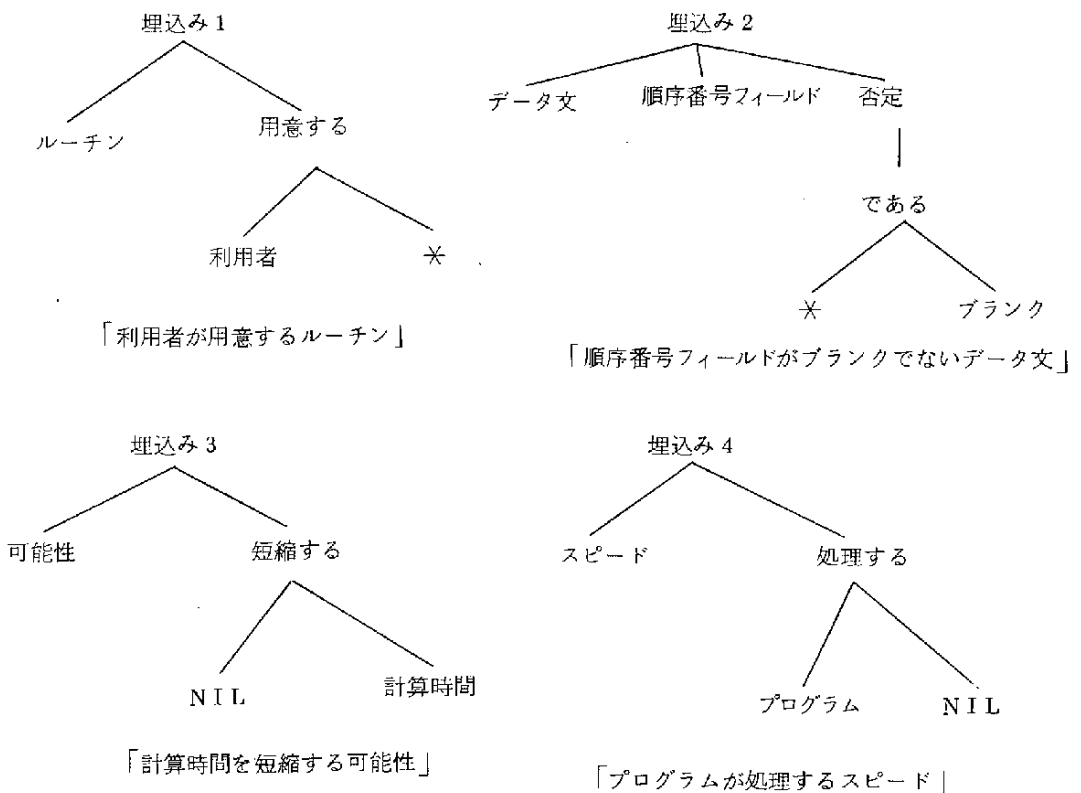
このカテゴリーの中間語は全て支配語であり、国文法における接続詞、接続助詞および条件節を形成する。‘時’ ‘場合’ などの一部の名詞および格助詞 ‘の’ などがある。特に名詞句中の格助詞 ‘の’ は、名詞間の接続関係によって細分類化された関係記述語に対応づけられている。

(5) カテゴリー S H

このカテゴリーの中間語は全て被支配語であり、他の中間語を修飾する働きをする。国文法における形容詞、形容動詞、副詞および ‘以上’ ‘各’ 等の名詞句内の付加的表現をつかさどる単語に対応する。

(6) カテゴリー K J

このカテゴリーの中間語は全て被支配語であり、単語として表現に表われない関係記述語である。これらの例を表 4-3 および図 4-9 に示す。



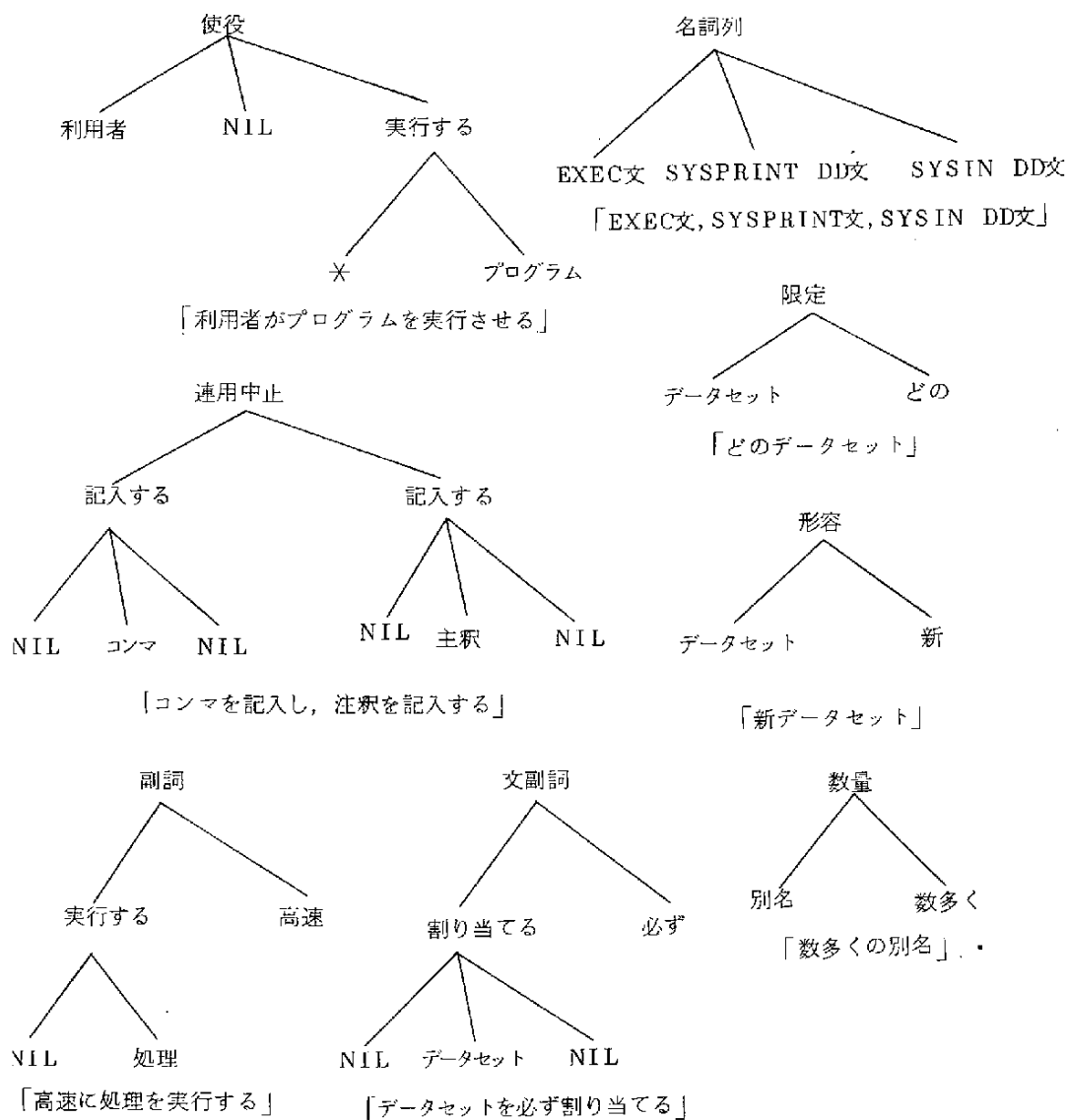


図 4 - 9 カテゴリー K J の中間語を用いた日本語中間構造
(例文は表 4 - 3 にもとづいている)

表 4 - 3 カテゴリー K J の 中 間 語

名 前	中 間 語	説 明
埋 込 み 1	UMEKOMI 1	修飾される名詞が修飾する文（動詞）の格要素になっている。 例：利用者が用意するルーチン
埋 込 み 2	UMEKOMI 2	修飾される名詞の部分、属性を表わす名詞が修飾文中の格要素になっている。 例：順序番号フィールドがブランクでないデータ文
埋 込 み 3	UMEKOMI 3	修飾される名詞の具体的内容の説明として、修飾する文がかかっている（同格表現）。 例：計算時間を短縮する可能性
埋 込 み 4	UMEKOMI 4	修飾される名詞が修飾する文で表現されている事象と何らかの意味で関係している。 例：プログラムが処理するスピード
使 役	SHIEKI	使役助動詞「させる」があることを示す。 例：利用者がプログラムを実行させる
名 詞 連 続	MEISHIRETSU	名詞が連続していることを示す。 例：必要な制御文は、EXEC 文、SYSPRINT DD 文、SYSIN DD 文である。
連 用 中 止	RENYOCHUSHI	述語が連用中止されていることを示す。 例：コンマを記入し、注釈を記入する
名詞限定修飾	DET	名詞が限定的に修飾されていることを示す。 例：どのデータセット
名詞形容修飾	QUALIFY	名詞が様態的に修飾されていることを示す。 例：新データセット、各論理レコード
名詞数量修飾	QUANTIFY	名詞が数量的に修飾されていることを示す。 例：数多くの別名、いくつかのデータ
副 詞 修 飾	V-MOD	述語を副詞的に修飾していることを示す。 例：処理を高速に実行する
文 副 子 修 飾	S-MOD	文を副詞的に修飾していることを示す。 例：データセットを必ず割り当てる

4.2.3 日本語中間表現の問題点

日本語中間構造は、構文解析の結果として与えられるものであり、構造変換の出発点である。構文解析が強力であればある程、日本語中間構造を意味的な表現にすることができる。逆に構文解析が、それ程強力でない場合は、日本語の表層上の表現形態が、英語中間構造においても強く反映されることになる。日本語中間表現は、構文解析と構造変換の部分が、どの程度までの処理を行なうかによって変化する。例えば、次の例を考える。

(例文)

c. プログラムがデータを変化させる

d. 彼が犬を走らせる。

c, d は共に自動詞に使役の助動詞 'させる' が接続している。'変化する' に 'change', '走る' に 'run' を対応させるとし、使役の関係を直接英文に反映すると、c, d はそれぞれ次の様な英文となる。

c'. The program makes the data change.

d'. He makes the dog run.

この場合、d' は理解可能であるが、c' は理解可能ではない。c' の訳としては

c''. The program changes the data.

という英文を生成すべきである。c'' の英文を生成するには日本語中間構造が、図 4-10 a, b のいずれかの構造をしていても良い。a の場合は、構文解析の段階で、'変化させる' を 1 つの中間語としている。構造変換では、'変化させる' をそのまま 'change' に対応させれば C'' が生成できる。一方、b の場合は、構造変換過程で、日本語中間語「変化する」に対応する英語表現 'change' には、他動詞と自動詞とがあることを認識して、使役の形 c' ではなく、他動詞を使った C'' を生成するように変換しなければならない。以上の様に、解析過程と変換過程での処理の分担の度合いに応じて日本語中間構造は変化することになる。この他、計算機マニュアルに現われた、日本語中間表現で記述困難な例を表 4-4 に示す。

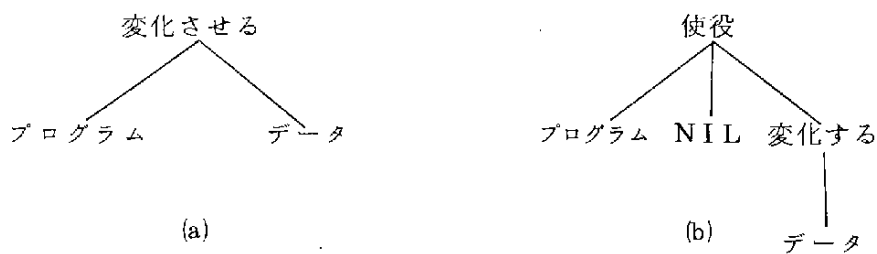


図 4-10 「プログラムがデータを変化させる」の2つの日本語中間表現

表 4-4 日本語中間表現で表現が難しい例

日本語表層	説 明
元のまま	例文 'データセットは変更されず元のままである' 国文法に従って'元'と'まま'に分割すると意味を成さないため'元のまま'で1つのまとまりと考える。この場合、'元のままだ'を述語とするか'である'を述語とするには任意性がある。
この場合	例文 { この場合 このような場合 } データを修正する この特別な場合 この例では、'場合'を接続を示す語と考えることは、生成の立場から適切でない。何らかの形で{ }内の構造を表現する必要がある。
文につく副助詞	例文 データを入れる <u>だけ</u> の十分な領域 修正を行なう <u>のみ</u> で、実行はしない 'だけ' 'のみ'だけでは意味が定まらない。構文解析と密接に関係しているため、処理をどの時点で行なうかが大きな問題となる。
～, 例えば～	日本語中間表現では、日本語の語順に関する情報を、基本的に失ってしまう。このため、文章中に任意に挿入された句などをどの様に表現するかが問題となる。
比較に関する表現	例文 システム領域より大きい利用者領域 'より大きい'を1つの中間語と見るのは、'よりずっと大きい'などの表現があり实际的でない。比較を表わす関係述語を導入してゆく事が考えられる。

4.2.4 英語中間表現

4.2.1で述べたように、本システムの英語中間構造は、句構造表現で記述する。句構造表現では、木の終端に単語が位置し、非終端に単語や句の文法カテゴリを示す記号が位置する。この記号を以後カテゴリ記号と呼ぶ、カテゴリ記号がどのような単語あるいはカテゴリ記号を支配するかは、書き換え規則(rewriting rule)の表記を用いて示すことができる(図4-11)。この書き換え規則の設定、すなわち句構造表現の設定により説明できる言語現象(機械翻訳の立場からは、生成できる英文の範囲)が規定される。現在のところ、言語学者の間でも、句構造の構成をどのようにするか定まった体系は存在しない。我々は、変形生成文法に関するいくつかの文献を参照し、英語中間表現を設定した(この句構造表現は、変形生成文法の始発句構造表現(語彙挿入後の)に対応している)。英語中間表現の特徴としては次のことがあげられる。

- (1) 変形文法では扱っていない副詞句(AVP)を導入している。
- (2) 英語中間構造に現われるカテゴリ記号に任意のノード情報を付加することができる。

英語中間構造は日本語中間構造から構造変換によって生成されるという性質上、受動態、様態等の情報は英語中間構造の構造としてでなく、カテゴリ記号にノード情報として付加している。これらのノード情報は、構造変形過程で解釈され、英語表層上に構造化される。

S	→ NP VP		
NP	→ { [DET] [QP] [AP] { N } [PP...PP] } { CMP S } { CMP VP } { NP RPP }		
DET	→ { a } { the } { one's } { NP }		
QP	→ [AVP] [QM] QUT	VP	→ { V } [NP] [AP] [PP...PP] { VG }
AVP	→ AVP ADV [NP]	V	→ <動詞>
ADV	→ <adverb>	VG	→ V { VGA } { VGP }
QM	→ [QME...QME] QME	VGA	→ <adverb>
QME	→ { more } { than } { less } { or }	VGP	→ <pveposition>
QUT	→ { 数詞 } { some } { many } { any }		
AP	→ [AVP] ADJ [NP]		
ADJ	→ { <adjective> } { <動詞の過去分詞> } { <動詞の現在分詞> }		
N	→ <noun>		
NC	→ [NE...NE] NE		
NE	→ <noun element>		
PP	→ { PRP } NP { GPRP }		
PRP	→ <preposition>		
CMP	→ { that } { to } { ING } { than }		
RPP	→ RP S		
RP	→ { <relative pronoun> } { RAV }		
RAV	→ <relative adverb>		

図 4 - 1 1 英語中間表現のシンタックス

以下では、英語中間表現のカテゴリー記号 S, NP および ノード情報について例を挙げ簡単に説明する。

(1) カテゴリー記号 S

S は文を示すカテゴリー記号である。古典的な英語文法では、英語文は基本 5 文型に分類される。5 文型の句構造表現例を図 4-12 に示す。句構造表現では、主語 (subject), 目的語 (object), 補語 (complement) の情報は句構造中の名詞句 (NP) の位置に明示的に示される。また、英語中間構造では、主語の位置に何もなくても許される。この様な非文法的な構造は、構造変形過程で処理され、受け身化する等の変形により文法的に正しい英文となる。

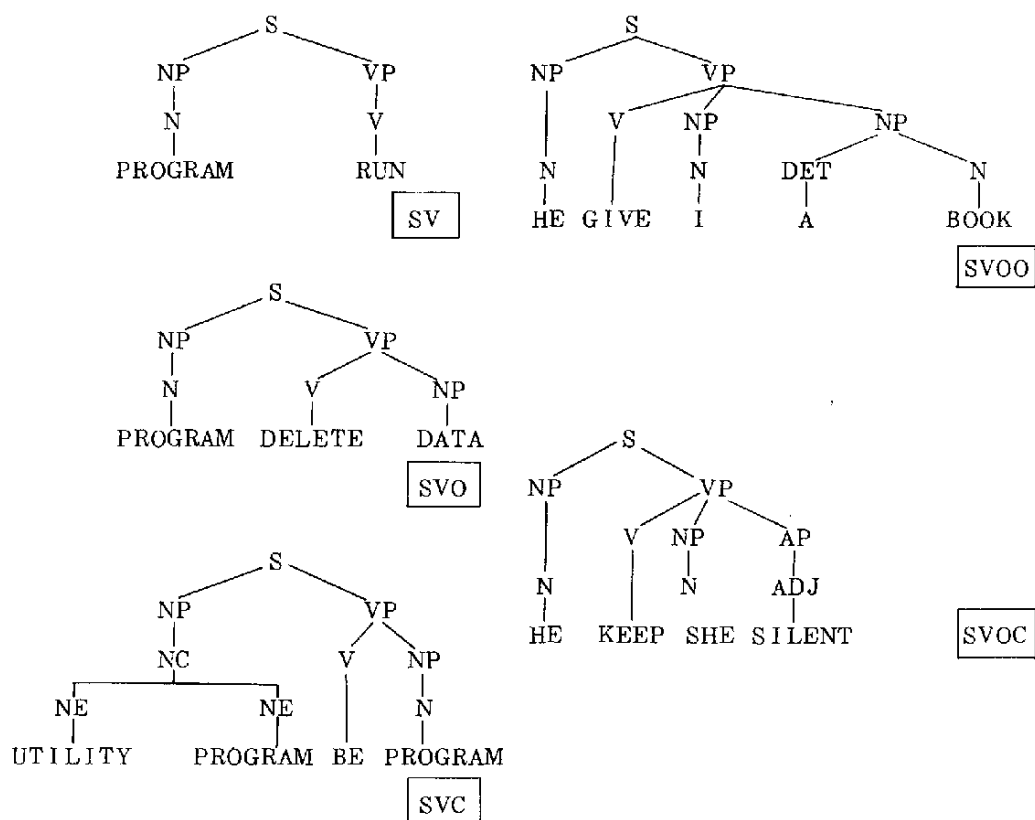


図 4-12 英語基本 5 文型の句構造表現

(2) カテゴリー記号NP

NPは大きく分けて、名詞(NあるいはNC)を含むNP、補文標識(CMP)を含むNP、の3つがある。これらについて例を挙げる。

(例文)

a. the two deleted member of the partitioned data set
aの句構造を図4-13-aに示す。DETは'the','a'などの冠詞、
'his','its','whose'などの所有代名詞あるいは'Jhon's'等の
名詞句のいずれかを示すカテゴリー記号である。QPは、数量限定句であり、
名詞の数量に関する修飾を示す句である。APは、形容詞句を示すカテゴリー
記号であり、一般の形容詞、動詞の過去分詞形、現在分詞形および'first'
などの序数詞等がAPによって表わされる。PPは前置詞句を示すカ
テゴリー記号であり、'前置詞+名詞句'の構造をしている。次に表層上の
単語のカテゴリーの並びは全く同じであるが、構造は全く異なる例を挙げる。

(例文)

b. the name of the member of the dataset (図4-13-b)

c. the name of the member in DS parameter (図4-13-c)

次に補文標識を含むNPの例を挙げる。

(例文)

d. to delete data set (図4-13-d)

e. that user delete data set (図4-13-e)

f. deleting data set (図4-13-f)

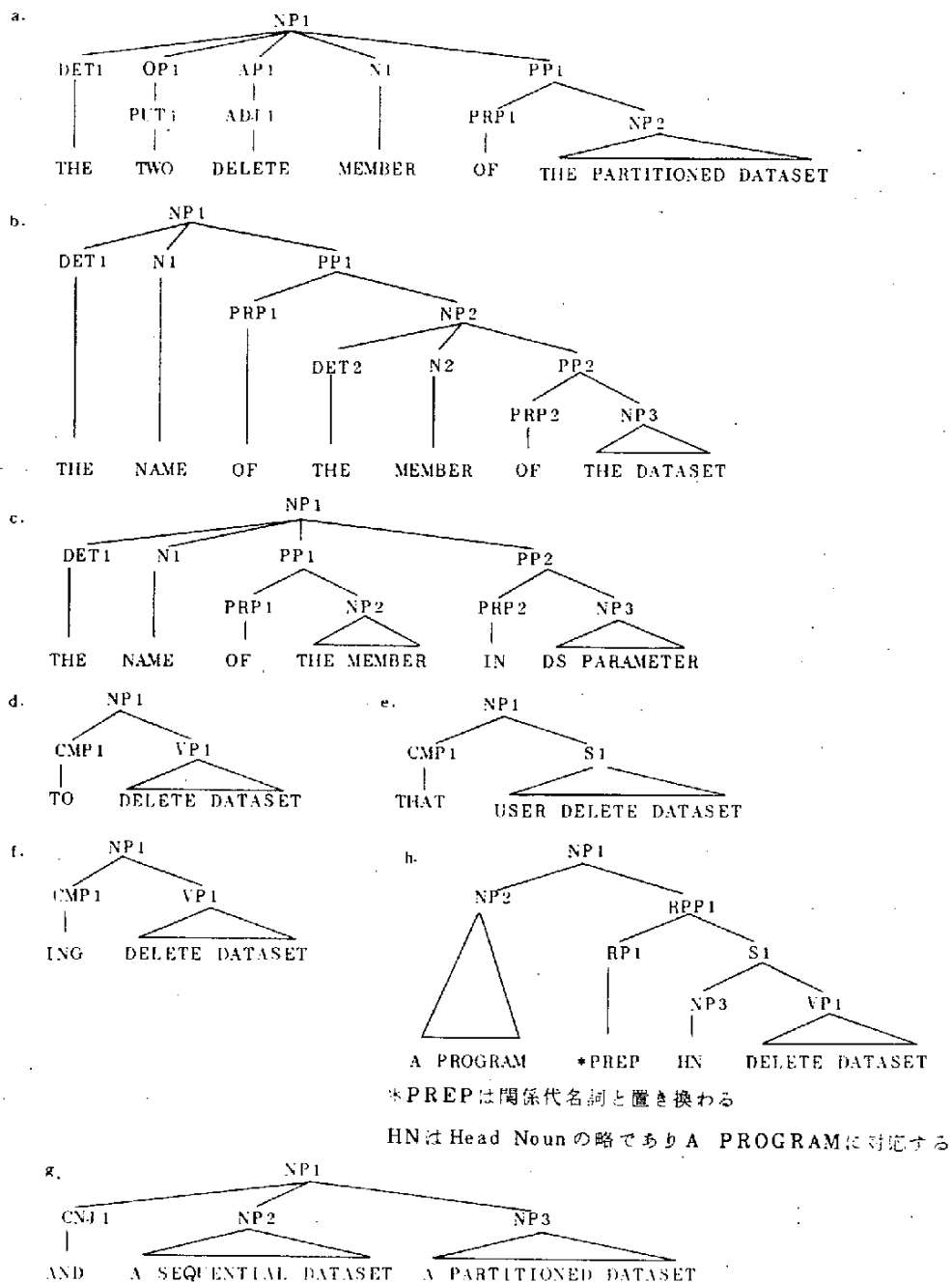
例文のd~fはそれぞれ、to-不定詞、that節、動名詞に対応する。これ
らは主語の位置にくることができ、名詞を含むNPと同じ様な働きをする。

次に、接続詞で名詞句が結合されている例を挙げる。

(例文)

g. a sequential data set and a partitioned data set

(図4-13-g)



英語の中間構造では、接続詞は先頭の位置にあるが、変形過程で正しい位置に移動する。

最後に、下位に関係節を持つNPの例を挙げる。

(例文)

h. a program which deletes data set (図4-13-h)

英語中間構造では、関係代名詞は‘* prep’というシンボルで表わされ、変形の過程で実際の単語に置き換わる。

(3) ノード情報

英語中間構造では、カテゴリー記号Sに文の態、時制、様態、アスペクト、文の種類の情報が付加されている。

4.3 日本語解析過程

自然言語の解析の場合、1個の入力に対し1個の正しい解釈結果が得られるのが理想的である。しかし、このためには単に文中の単語の意味、各単語間の係り受けが解析できるだけでなく、文の前後関係、人間が持つ実世界に対する知識、人間が持つ推論の機能が必要となる。実際、前後の状況がなく、一文だけが与えられると複数個の解釈を生ずることがしばしばある。逆に言えば、解析において十分な情報が与えられないにもかかわらず、可能な全ての解釈が得られなければ、その解析プログラムは正しい解析を行っていないことになる。従って、本解析プログラムでは最初に構文的に可能な解析を全て求め、その後文法記述者が記述する単語の意味、各種制限事項をデータとして解析プログラムが参照して不適当な解釈を除くという方式を取った。こうして、係り受け構造を作る部分とその可否を判断する部分を明確に分離し、解析プログラム自身が解析に不当な制限を暗黙のうちに持ち込むことを防いだ。

また、解析において必要となる各種の制約条件は、意味的な制約条件は当然のことながら、表層表現上の格助詞の選択およびその変化等の統語的な制約も、

各単語に固有なものが多い。したがって、我々は日本語解析プログラムの作成において、各単語固有の辞書記述を重視する立場をとることとした。

4.3.1 解析プログラムの全体構成

統語解析は、句構造文法、依存関係文法（これを日本語向きに改めたものとして、「係り受け関係」による文法がある）によるものが一般的であり、またこの種の文法に変形操作を加えた変形文法の枠組もよく知られている。

我々の日本語解析プログラムは、後に述べるように統語解析には、日本語に適していると考えられる依存関係（係り受け関係）に注目した手法を用い、意味処理に関しては現在の段階では比較的単純な意味ネットワークと意味マーカの手法を用いている。また、依存関係によって解析された結果は、4.2で述べた格関係に基づく日本語中間構造へと写像される。日本語解析の全体的な流れは、次のようになる。

- Step 1. 日本語漢字かな混り文をもとに、文節区切り、文節内の付属語列処理、活用語の終止形への変換（音便処理を含む）、自立語の辞書引き、等いわゆる形態素処理を行う。
- Step 2. 接頭語、接尾語の処理、日本語の文体構造を規定する表層表現（～ば、～する場合、連用中止法、連体形）の処理等、単語の前後数語を参照するだけで、安定に処理できる言語現象の処理を行なう。
- Step 3. 文体構造を認識して、入力文に対応する文体木を作る。
- Step 4. Step 3. で作られた文体木を参照し、名詞句処理・単文処理を行なう。
- Step 5. 連体形埋め込み文に伴う言語現象の処理・連用中止法に伴う言語現象の処理等、名詞句処理や単文処理の結果をまとめあげる、より高次の処理を行なう。

Step 4 と Step 5 の処理は、必ずしもこの2段階に分離して行なうのではなく、Step 3 の段階で作られた文体木をもとに、相互に呼び合う形で実行され

てゆく。また、Step 2においては、日本語述部に後続する‘～てある’，
 ‘～られる’等の、述部の表層格パターンを変化させる表現を処理し、述部辞書
 記述の中から、Step 4 で使われる適切な表層格パターンを選択する処理も同
 時に行なわれる。なお、Step 1は、すでに我々が以前に開発した日本語形態
 素処理プログラムを用いており、数値表現・長い漢字連続を含むいくつかの処
 理にまだ問題を残しているが、一応完成されたプログラムになっている。

Step 1は、PL/1、Step 2以後はLISPをプログラムとして用いており、こ
 の間のデータの受渡しは、現段階ではファイルで行なっている。

本報告では、Step 2 以降について詳しく述べることにする。

4.3.2 文体木の役割とそれを用いた処理

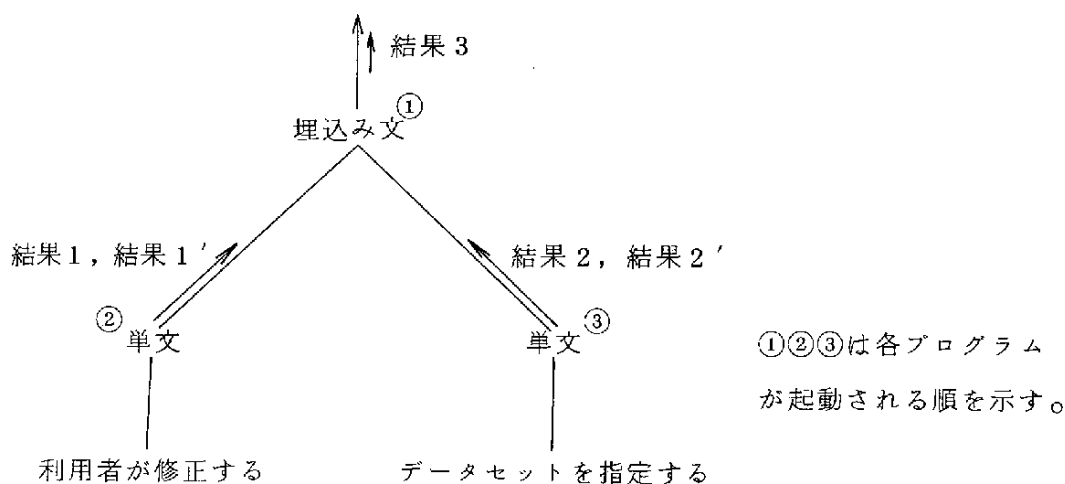


図 4-14 文体木に基づく処理の例

Step 3で作られる文体木の例を、

‘利用者が修正するデータセットを指定する。’

について示したのが、図 4-14 である。この文体木は、文末の‘指定する’の格要素を修飾するものとして、‘修正する’によってまとめられる埋込み文がきていることを示す。この文体木は、入力文の大局的な構造を表示していると同時に、Step 4 以後の処理順序を規定するプログラム木としての役割を果たしており、文体木中の各節点は、それぞれ起動されるべきプログラム名を示している。図 4-14 の場合は、Step 4 以降の処理は、この文体木に従って次のように進められる。

- (1) プログラム埋込み文が起動され、この埋込み文はさらに自分の下位に接続する左側のプログラム単文を起動する。
- (2) (左側の)プログラム単文が値を返す。結果は 2 通りになり、結果 1 は、‘利用者が’が‘修正する’に支配されたもの、結果 1’ は支配されなかったものである。
- (3) プログラム埋込み文はこの結果をうけとって、結果 1 に対しては、単に自分の下位に接続する右側のプログラム単文を起動し、結果 2 を返す。
- (4) 結果 1’ に対しては、‘利用者が’データとして右側のプログラム単文に渡す。(左側の)プログラム単文は、‘利用者がデータセットを指定する’を解析し、結果 2’ を返す。
- (5) プログラム埋込み文は結果 1 と結果 2 と、結果 1’ と結果 2’ とを使って、埋込み構造をつくり、両者を合わせて結果 3 として値を返す。この結果 3 が全体の解析結果になる。

このプログラム埋込み文のように、文体木中におのおのの日本語表現に対応した特別のプログラムを組み込んでいくことによって、個別的な言語現象を処理してゆくことになる。この専用プログラムを設けるということは、現在の文法規則とそれを適用するプログラムの分離という方向に一見反するように見えるが、各専用プログラムごとに固有の文法記述の体系を設けており、全ての

文法規則を統一した体系で記述しないというだけで、文法規則とプログラムの分離は考慮されている。また、この手法を用いることにより、将来新しく発見した言語現象や種々の手続き的な規則も比較的容易に組み込むことが可能である。また、自然言語解析のプログラムにおいては、局所的にみると解析結果が複数個存在するという状況が常に生じ、これをどのように処理するかが問題となる。とくに、プログラムによって処理する場合には、個々の部分の解析において生じるこの複数個の解析結果をどのように保存し、解釈を進めてゆくか、また、このような book keeping 的な処理と、文法的な処理をうまく分離して、わかり易いプログラムとするかが問題になる。我々のシステムでは、こういった並列性を文法記述者から隠し、文法記述を行なう場合には、常に一個の解釈結果だけがあると考えて、文法記述することが可能となる機能を付加している。

文体木を設けたのは各専用プログラムの独立性、拡張性の維持と共に、文解析の際の処理、データの流れを明確にし、並列の問題を解決するためである。例えば、例文 (1.1)

例文 (1.1) 利用者が修正するデータセットを指定する。
の場合、‘利用者’は動詞‘修正する’、‘指定する’の両方に係り得る。このとき、連体修飾の埋込みの処理、各単文の処理、‘利用者’の係りによる並列の処理を同時に1つの解析プログラムで行なうことはプログラムの動作を非常に複雑にし、処理過程の見通しを悪くする。図4-14のように文体木に基づけば処理過程は明確になる。

4.3.3 単文解析における問題点

日本語解析プログラムの Step 4 において、中心的役割を果たすのが単文の解析である。連体形埋込み文・連用中止法の処理等の、より高次のまとまりを処理する場合も、すべてこの単文解析プログラムの結果を使って行なうことになる。ここでは、単文を処理する場合に考慮する必要のある言語現象について

考察する。

日本語単文の解析は一般的に格助詞を手がかりとして行なうことができるが、実際に現れる現象は複雑で表層格パターンの詳しい記述、とくに各動詞ごとの記述が必要になる。また、次項(3)で述べるような個々の動詞の表層格パターンでは記述しきれない現象の問題もある。ここでは主に表層格パターンに関連する現象について述べる。

(1) 表層格と述語

ここでいう表層格とは入力文に現れる名詞句が伴っている格助詞である。一般に述部が定まればその述部と依存関係を持つ要素が伴う表層格の組は一意に定まり、異なる依存関係が同一の表層格を取ることはない。

日本語についての変形文法理論においては、格助詞は一般的な規則によって名詞句に付加されると説明されるが、実際の機械処理の立場から、例えば‘道を歩く’、‘フランス語が好き’のように‘ヲ’格が必ずしも目的格を示すとは限らず、場所格を示す場合があること、逆に目的格が‘ガ’格で表記されることもあることなど、述語(動詞)が何であるかによって、かなり固有の部分が多いことを考えると、動詞ごとに固有な表層格のとり方があると考えた方が都合が良い。そこで、我々の日本語解析システムにおいては、各動詞を辞書記述に、その動詞の表層格パターンと、その表層格パターンに対応して、4.2で述べた格表現にもとづく日本語中間表現を対にして与える方式をとることにした。

(2) 複数個の表層格パターン

(例)

- a. ファイルパラメータをNEW { ^に_{とを} } 指定する。
- b. ファイルパラメータにNEW { ^に_{とを} } 指定する。
- c. ファイルパラメータをNEWであると指定する。
- d. ファイルパラメータがNEWであると指定する。

この例のように、一つの動詞は表層上いくつかの可能な格パターンを持ってい

る。文を生成の立場から捉えるのではなく、文構造を認識する立場から初めに与えられるのは、文の表層上の表現のみである。各動詞のとり得る「深層格」を設定し、その深層格が表層にあらわれるときにどのような規則によって具体的な格助詞に写像されるかを記述しておくという方式は、認識の立場からはあまり良い方法ではない。したがって、我々は各動詞辞書にその動詞の取り得る表層上の格パターンをすべて列挙する方式をとっている。各表層格パターンは、それぞれ日本語中間表現に対応している。したがって、処理は入力文（前処理によって単文に分割されている）とその述語の表層格パターン（複数のパターン）のマッチング、マッチングの結果の対応する日本語中間表現へマッピングという2段階の操作で、直接的に行なわれることになる。

(3) 表層格の変化

各動詞の辞書記述に、その表層格パターンと日本語中間表現を対応させることを基本にする我々の方式は、表層格パターンが何らかの変形を受ける場合に、適切に対応する必要がある。表層格パターンを変化させる要因は、生成文法において補文構造をとると判断されている助動詞等の助述表現がくる場合である。表層格を変化させる原因としては、すでによく知られている直接受動文以外に次のようなものがある。このような格パターンの変化についても、我々のシステムではすべて辞書に列挙する方式をとっている。

1. 間接（利害の）受動文

(例) a. 利用者Aがデータセットを消去された。

注) 直接受動文においては、

‘データセットを消去した。’ → ‘データセットが消去された。’

例aでは、利害の受身であるので、‘ヲ’格はそのまま変化せずに残っている。そのかわり、利害関係に立つ‘利用者A’が直接表層上に‘ガ’

格として実現される。したがって、次のような埋込み文を処理するためには、直接受動文だけでなく、間接受動文の可能性も考えておく必要がある。

- (例) b. データセットを消去された利用者
c. データセットが消去された利用者

2. 可 能 文

- (例) d. データセットにパスワードをつける。
→ データセットにパスワード{^を_が}つけられる。
→ パスワードがつけられるデータセット(埋込み化)
→ データセット{^は_も}パスワードがつけられる(副助詞化)

注) 行為者を示す‘ガ’格が表面にでる場合には、二重のガ格になるが、これには下のような制約がある。

- (例) e. 利用者がデータセットにパスワード{^を_が}つけられる。
利用者がパスワード{*^を_が}データセットにつけられる。

3. 難 易 文

- (例) f. この制御文が使いやすい。

注) 可能文においては、‘ヲ’格→‘ガ’格の変化は任意的でどちらも使われるが、難易文の場合は、強制的に‘ガ’格になる。しかも、‘ヲ’格以外の格もかなり自由に‘ガ’格になるようである。

(例) g. このペンが書きやすい。

(* このペンが書く。 * このペンを書く。 * このペンで書く。)

4. 使役文

(例) h. UPDATE プログラムにデータを修正させる。

5. ー である

(例) b. 修正してあるプログラム

c. プログラムが修正してあるユーティリティ・プログラム

‘ーである’は、動詞についてもとの動詞を状態動詞に変えるが、これにともなって、動詞の表層格パターンも変化し、‘修正してある’の場合には、‘が’格となる。本来の‘修正する’が持っていた深層格での行為者格が表層上にはあらわれなくなる。

(4) 表層格の変化 — 副助詞化

表層格パターンによる方式のもう1つの問題点は、副助詞の処理である。副助詞が名詞句につく場合、表層格を表示する格助詞が脱落するため、単純な入力文と表層格パターンとのマッチングが不可能となる。しかしながら、副助詞化による格助詞の脱落には、ある種の規則性があり、特に各動詞固有の格の序列と副助詞化に次のような関係があることが認められる。

(例) a. 利用者がユーティリティプログラムでデータを修正できる。

b. 利用者もユーティリティプログラムでデータを修正できる。

c. 利用者がユーティリティプログラムでデータも修正できる。

d. 利用者がユーティリティプログラムでもデータを修正できる。

上記の例からわかるように、格助詞を省略した形での副助詞化が可能な格は、
‘修正する’の場合、‘ヲ’格と‘ガ’格のみであることがわかる。したが
って、副助詞だけのついた名詞句があれば、‘ヲ’格と‘ガ’格を埋める可
能性のみを調べれば良いことがわかる。この現象はほとんどすべての動詞に
ついては、‘ニ’格の副助詞化になると、各動詞に依存する。

(例) a. INFILE パラメータに NEW を指定する。

→ INFILE パラメータ { ^も_は } NEW を指定する。

b. OUTMEMBER にデータを書き出す。

→ *OUTMEMBER { ^も_は } データを書き出す。

また、この副助詞化の規則は、対応する名詞句が埋める深層格によらず、表
層上の格助詞だけに依存するように思える。

(5) 単文中の格助詞 ‘の’

我々の日本語解析システムにおいては、Step2・Step3 において、文体
構造を認識し、入力文を単文単位に区切っている。したがって、単文処理に
おいては、一般に

(名詞句) + (格助詞 AND/OR 副助詞)

の形式の格要素の並びを処理することになる(この他、副詞等の連用修飾句
が含まれることがある)。名詞句は、格助詞 ‘の’ によって結びつけられ、
名詞から構成されるのが普通であるが、この格助詞 ‘の’ に関しては次のよ
うな現象に注意する必要がある。

(例) a. 名前が JSEUPDTE のプログラムを指定する。

(名前が JSEUPDTE であるプログラムを指定する。)

b. 利用者のパラメータで指定したデータセット

c. このデータセットのセンタにより消去されたメンバ

例 a は、格助詞「の」が述語「である」の代用として使われる場合があること、例 b は、「の」が埋込み文において格助詞「が」の代用となり得ること、例 c は「の」に先行する名詞句が必ずしも直後の名詞に係るのではなく、連体形の述語を飛び越して、埋込み文によって形容されている名詞に係ることがあり得ることを示している。このような言語現象が「の」に関してあるために、

(「(名詞) + の」の繰返し) + (格助詞 AND/OR 副助詞)

の形を見出しても、必ずしも「「(名詞) + の」の繰返し」が 1 つの名詞句としてまとまるとは限らないことになる。我々の単文解析プログラムにおいては、上記例 a, b, c で示された「の」についての表現形をすべて処理するようになっている。

(6) 意味的制約

以上(4)までは、統語上の言語現象であり、日本語の文一般について成立するが、意味的制約は対象をある分野について限定した場合にだけ制約条件として用いることができる。我々のシステムは、現在計算機マニュアルを対象としているため、この分野についての「知識」を援用することによって、統語条件だけでは曖昧さの生じる文をうまく処理することができる。

しかしながら、意味的制約条件は現在のシステムではあくまで便宜的に使用しており、もし意味的制約条件がえられておれば、曖昧さが減じるが、なくてもシステムとしては動作できるという方針でシステムを作成した。現在のシステムでは、意味的制約条件は、簡単なセマンティック・ネットワークによって行なっている。

以上のことから、我々の辞書に記述されている表層格パターンは、その述部が持ち得る依存関係の組と、述部と文中の要素がその依存関係を持つときに

その要素が満たすべき制約条件を記述したものになっている。この表層格パターンに記述できるのは各関係が独立に持ち得る制約だけであり、2つ以上の依存関係にまたがる制約はここに記述することはできない。記述できる条件には例えば次のようなものがある。

- (1) 文中の要素の型
- (2) 文中の要素が取り得る表層格や副助詞化する副助詞
- (3) 要素が満たすべき意味的制約

述部の態の変化等により取り得る表層格のパターンが変化する現象に対処するためにはこの表層格パターンを複数個あらかじめ用意しておき、局所的な関係解析の結果得られた情報をもとに、単文解析に使用する表層格パターンを選択することになる。

4.3.4 単文解析プログラム

処理の手順の概略を図4-15に示す。

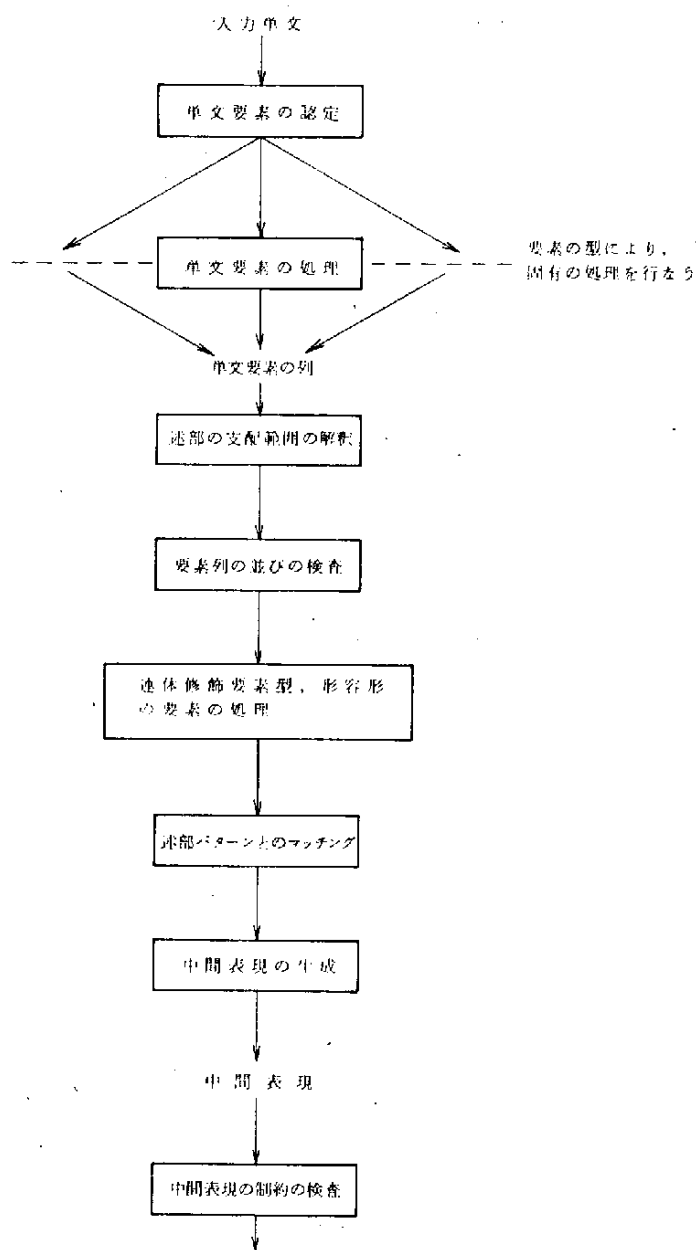


図4-15 単文解析

〔1〕 単文要素の認定

単文解析プログラムが対象とする単文は、述部につづく助述的表現（ことができる、である、等）のまとめあげ等の、ある程度の局所的関係がすでに解析されている以外は、ほぼ形態素処理によって認定された単語単位のままである。しかし、単文解析が主として処理する係り受け関係は、これらの単語と述部との関係ではなく、例えば名詞句等のまとめあげが終わった格要素や副詞といった、連用修飾句と述部との関係である。ここでは、こういった連用修飾句のことを単文要素と呼ぶことにする。従って、単文解析においては、最初に単文から単文要素を取り出し、それらがどのような要素であるか（要素の型）を知る必要がある。なお、この要素の型も要素の持つ属性の1つで、後の処理でもこの要素の型が参照される。

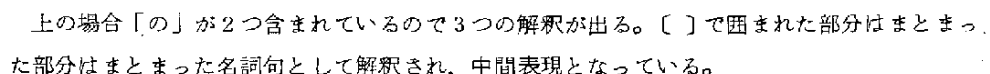
単文要素の取り出しは表4-5に示すような要素取り出しパターンを用い、このパターンを入力単語列の左側から表に示した順序で順番にマッチングさせてゆく。適合したパターンがあればその入力単語列部分を一要素と認定し、要素の型に対応した固有の処理（例えば名詞句処理）を行なう。このようにして入力文のすべての要素が取り出されたなら、得られた単文要素の列を出力として次の処理に進む。

表 4 - 5 要素の型と要素取り出しボタン

要素の型	要素取り出しボタン	ボタンの機能
形 容	形 容 詞 又は形容動詞	形容詞，形容動詞を取り出す。
副 詞	副 詞	副詞を取り出す。
個 数	数 値 表 現	数値表現を取り出す。
独立名詞	名 詞 + >	名詞 1 個よりなる，名詞句を取り出す。
の格要素	* + 「の」 + > 又は * + 「の」 + 名詞以外	格助詞「の」に導びかれる名詞句を取り出す。
格 要 素	* + 格助詞，副助詞 + > 又は * + 格助詞 + 「の」以外	「の」以外の格助詞，副助詞に導びかれる名詞句を取り出す。
独立名詞	* + 名詞 + >	名詞句だけで構成される要素を取り出す。
連体修飾要素	—	パターンによって抜き取られるのではなく，名詞句の分割により生成される。

記号「>」は単文がそこで終わることを，「*」は，0 個以上の任意の単語がくくることを示す。

[1]で述べたように取り出された単文要素は要素の型に固有の処理が行なわれる。この処理のうち特に重要なものが、独立名詞句型、格要素型の要素に含まれる名詞句を処理する名詞句処理である。名詞句は通常格助詞“の”と名詞が連続した単語の並びであるが、4.3.3の[5]で述べたように格助詞“の”が名詞間の係り受けを示す以外の現象を常に引き起こす可能性がある。しかも、この現象は“の”以外の格助詞で区切られた名詞句の範囲を越えて、他の単文要素や述部と直接関係し合うため、どの部分の“の”がこのような現象を引き起こすかを、名詞句の処理時に前もって知ることができない。従って、名詞句処理では図4-16のようにその名詞句全体が1つにまとまった場合の名詞句の中間表現に加えて、“の”の前後でその名詞句を分割した2個の名詞句の中間表現をも同時出力するようになっている。これら複数個の解釈が全て成立するとは限らないが(例えば、現在処理中の単文が埋込み文でなければ、格助詞“が”が“の”に変化するようなことはないetc)、名詞句処理の段階では全てあり得ると仮定して処理する。



- 112 -

[3] 述部の支配範囲の解釈および形容、連体修飾要素の処理

以上の処理の結果、入力単語列は単文要素の列になっているので、これをただちに述部の辞書記述にある表層格パターンとマッチングすることもできる。しかし、前にも述べたように、埋込み文の場合には、どこまでの単文要素が埋込み文中の述語によって支配されるかが明らかでないため、この時点で得られた単文要素列にそのまま表層格パターンを適用することができない。すなわち、単文要素列中の各要素について、その述部の支配範囲に含まれる解釈、含まれない解釈を検討しなければならない。この段階では図4-17のようにあらゆる支配範囲の解釈が行なわれる。

次に各々の支配範囲の要素列を検索し、形容要素、連体修飾要素が存在するか否かを調べる。形容要素が存在する場合は、それは形容詞、形容動詞または“である”の代用の“の”であるので、動詞等の場合と同じく依存関係表現を生成し、埋込み文の構造を作る。

以上の処理が終了すれば要素列中の要素の種類は副詞、個数、独立名詞、の格要素、格要素の5種類となる。

入力：要素1 要素2 ……要素n

支配範囲の解釈（n+1通り）

要素1 要素2 ……要素n

← 支配範囲

要素1 要素2 ……要素n

← 支配範囲

要素1 要素2 ……要素n

← 支配範囲

図4-17 支配範囲の解釈

〔4〕 表層格パターンとのマッチング

〔3〕で得られた述部の支配範囲に含まれる要素列に対し、表層格パターンをマッチングさせ、各要素が表層格パターンに記述されている依存関係の条件を満たしているかどうかを調べる。日本語においては表層格つまり格要素が順序任意で、しかも省略され得ることがあるが、格要素だけでなく一般に単文要素も順序任意で省略可能である。これを考慮し、要素列と表層パターンとのマッチングは「全ての要素が残らず、ある依存関係の制約条件を重複せずに満たす」ときに適合したと判断する。このとき依存関係と要素との対応は必ずしも一意に定まらず、もし複数個存在すれば対応関係をすべて出すことにした。従って、これをもとにして生成される中間表現も1つとは限らず、複数個となり得る。

中間表現は図4-18のようにして生成される。図中、中間表現の基本構造は、「単文への分解」における文体木の基本構造と同じく、中間表現作成のための原型となるものである。この中間表現の基本構造は、表層パターンと対となって述部の単語記述中に記述されている。

〔5〕 依存関係の整合性の検定

〔4〕のようにして生成された中間表現に対し、意味的制約条件を満足しているかどうかを調べ、正しい中間表現であるかどうかを検定する。

〔6〕 単文支配範囲に含まれなかった要素列

〔5〕までの処理で、与えられた単文の中間表現が得られる。この中間表現は、述部の支配範囲での解釈を与えるものであり、その範囲内の単文要素の中間表現は、この単文中間表現中の構成要素として埋め込まれている。一方述部の支配範囲に含まれなかった単文要素、すなわち、現在処理された述部を飛び越して、より後の述部へと係ってゆくと判断された単文要素に関しては、その要素と、それに対応する中間表現がそのまま対として返されることになる。結局、単文解析の結果は、

1. 〔5〕までの処理で得られた単文の中間表現

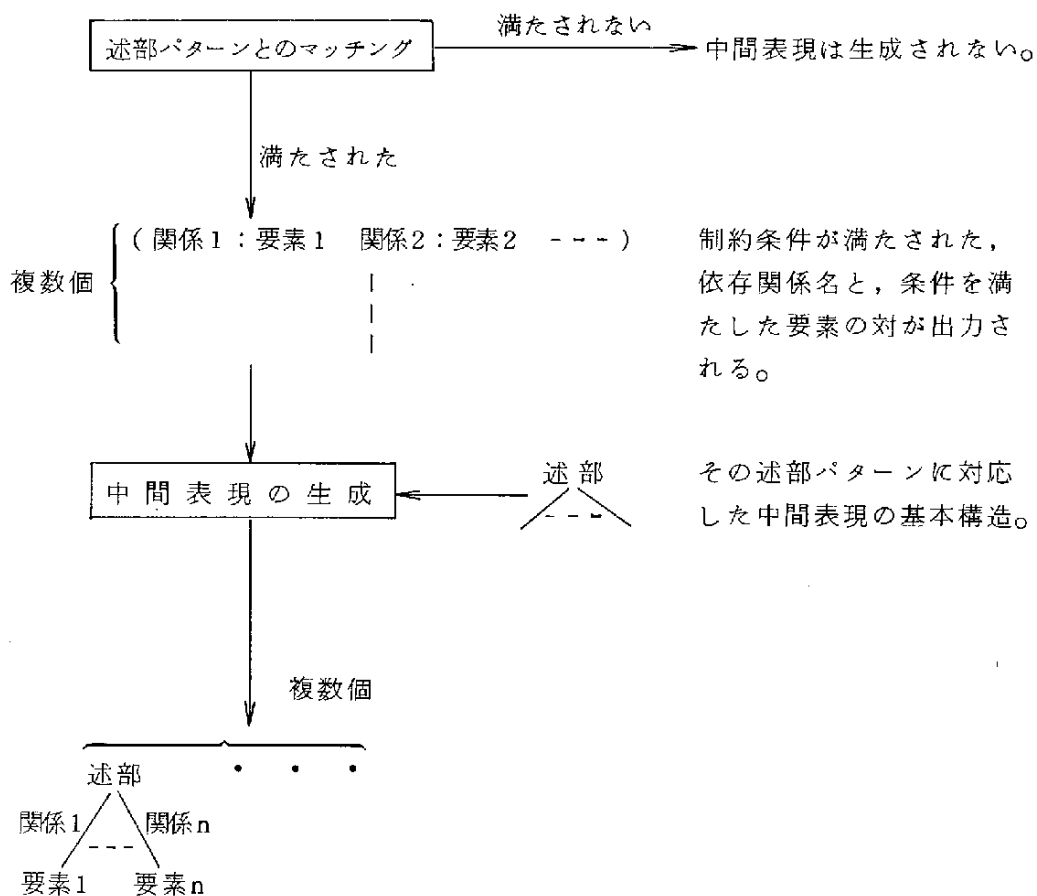


図 4 - 1 8 中間表現の生成

2. 与えられた単文中の要素とはならなかった単文要素とその中間表現の 対のリスト

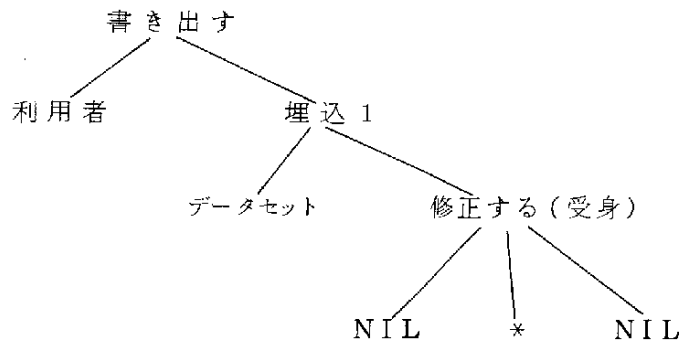
となる。このような単文解析の結果が複数個文体木中の上位のプログラムへ返されることになり、この上位のプログラムがどの解析結果が最も妥当であるかを判断することになる。また、上記2の、残された単文要素の係り先も、この上位のプログラムが決定することになる。

4.3.5 埋込み文処理の問題点

文の中に、その文中の要素を形容する形でもう1つの文が埋込まれたような構造を埋込み構造といい、このうち埋込まれた文を埋込み文、上位の文のことを主文という。例えば、

“ 利用者が修正されたデータセットをデータセットへ書き出す ”

といった文は、埋込み構造をしており、日本語文解析の Step 3 で作られる文体系木は、この文に対しては図 4-19 のようになる。この文体系木中、UMECOMI としたのは日本語の埋込み構造に関する種々の言語現象を処理するためのプログラムであり、4.3.4 で述べた単文解析プログラム TANBUN によって処理された埋込み文と主文の2つの解析結果を参照し、より高次のまとまりへと統合する役割を果たすものである。ここでは、このプログラム UMECOMI の構成に入るまえに、まず日本語の埋込み構造がどのような言語現象をひき起すか、埋込み構造を処理する場合に、どのような点に注意しなければならないかについて考察する。なお、埋込み構造に関連する言語現象としては、「単文要素の埋込み文の飛び越し」・「表層格“が”の“の”への変化」・「埋込み文における“である”の省略」・「埋込み文を飛び越す“の”」があるが、これについてはすでに 4.3.3, 4.3.4 の単文解析の項で述べたので、ここでは触れないことにする。



(SHUSHI NIL NIL (UMEKOMI NIL NIL
 (TANBUN NIL NIL (WD0 WD1) WD2)
 (TANBUN NIL NIL (WD3 WD4 WD5 WD6) WD7)))

WD0 : 利用者
 WD1 : が
 WD2 : 修正する(受身)
 WD3 : データセット
 WD4 : を
 WD5 : データセット
 WD6 : へ
 WD7 : 書き出す

図 4 - 1 9 文体木の例

[1] 埋込み文の型

日本語における埋込み文は、英語の関係節表現と直接対応するわけではなく、埋込み文とそれによって形容される名詞との間の関係は次のような4つ

に類型化できる。解析プログラムは、この4つの構造を処理できなければならない。

(形式1) 形容される名詞句が述語の格スロットを埋める

- (例) a. プログラムを修正する利用者
 b. データセットが存在する領域
 c. 彼が到着した日

これらは、通常もっとも頻度の多い構造である。日本語中間表現では、例 a. b. c. とともに同じ形式で表現されるが、形容される名詞句が埋める格スロットに応じて、英語の世界では関係代名詞・関係副詞の使いわけを行なう必要がある。

(形式2) 形容される名詞句が動詞の格スロットを埋めている要素と何らかの関係でつながる

- (例) a. メンバが消去された区分データセット
 b. 論理レコード長が変更されたデータセット
 c. 中心部を戦争により破壊された東京

例 a. では、“区分データセット”と“メンバ”が構成要素の関係で、例 b. では“データセット”と“論理レコード長”が属性の関係で、例 c. では“東京”と“中心部”が全体部分の関係でそれぞれ意味的關係がつく。どのような意味的關係をもつ場合に、この種の構造をとることができるのかを明らかにする必要がある。

(形式3) 同格表現

- (例) a. 既に存在していたデータセットが消去されてしまう(という)
 事実
 b. データを修正する目的
 c. データを追加する操作

この形式をとるのは、形容される名詞にごく限られた名詞がきた場合である。とくに、同格表現であることを明示するために、“という”が挿入され

ることもある。上記例のうち、例 b. c は同格でなく（形式 1）に類型化できそうであるが、その場合には“データをその操作で追加する操作”の意味になって、例 c が本来意味していた“データを追加するという操作”とは違った解釈になってしまう。また、一般に（形式 1）をとるのは、形容される名詞句が“ガ”格，“ヲ”格，“ニ”格の場合にかぎられており（実際には、この制約は「述語」に依存することになる）。例 b. c は（形式 1）では説明できない。

（形式 4）形容されている名詞句が埋込み文で表現されている事象と何らかの意味で関係する

- （例） a. プログラムを実行した結果
b. プログラムが処理するスピード
c. 機械翻訳システムが正しく翻訳する割合

これらは、同格表現のうちでも奥津が「部分的同格連体名詞」と類型化しているもので、形容される名詞と、埋込み文で表現される事象との間に、「言語外知識」を媒介した関係がある場合に生じる構造である。しかも、これらに対応する英語の関係節構造が直接にはないために、かなりの paraphrase をしなければならない。言語構造と言語外知識の関連をとらえる上で、今後とくに研究すべき問題であろう。

以上 4 つの埋込み構造は、英語においてはそれぞれ異なった表現形態（関係代名詞、関係副詞、所有格の関係代名詞、that 節 etc）で表わされる。とくに、（形式 4）については直接的に対応する構造がない。そこで、我々のシステムの日本語中間表現においては、それぞれ異なった中間語（UMECOMI 1～4）に対応づけられることになる。なお、現在作成した日本語解析システムにおいては、（形式 1）・（形式 2）のみを処理対象としており、（形式 3）・（形式 4）は今後の課題として残されている。

〔2〕 表層格の序列 — 形式1の埋込み文

(例) a. 修正するプログラム

b 1. データを修正するプログラム

b 2. プログラムが修正するデータ

c. 利用者がデータを修正するプログラム

d. 利用者がそれでデータを修正するプログラム

仮りに、“修正する”の表層上の格パターンが“ーガーヲーデ”とする。ここで、形式1の埋込み文によって、形容される名詞が埋込み文中で占め得る格は、“修正する”の場合、“ガ”格か“ヲ”格であり、“デ”格はこの場合対象とはならない。例cにおいては、プログラムは“デ”格を埋めるという形式1の解釈よりも、むしろ“(プログラムの)データを修正するそのプログラム”という形式2の解釈が優勢となる。形式1の解釈をもたせようとすると、代名詞を使った例d.の形式となる。

この条件は、しかしながらあくまで「不確実な」制約条件であり、意味的に強い要請がある場合、例えば、“利用者がデータを修正するユーティリティ・プログラム”といった場合には、“デ”格の解釈を意味的に強く要求する“ユーティリティ・プログラム”が形容される名詞句の位置にきた場合でも、“利用者が修正するユーティリティ・プログラム”といったように、“デ”格よりも上位の格(この場合は“ヲ”格)が埋込み文中にないときには“ヲ”格を埋める。

〔3〕 表層格の序列 — 形式2の埋込み文

(例) a. 論理レコード長が変更されたブロック化レコード

b. 論理レコード長を変更したブロック化レコード

c. 論理レコード長を利用者が変更したブロック化レコード

d. 利用者が論理レコード長を変更したブロック化レコード

形式2の埋込み文の場合も、形式1の場合と同様に、埋込み文中のどの格と形容される名詞が意味的に関係するかに制約がある。“変更する”の場合、

その対象となる格は受け身化されたパターンでの“ガ”格、能動文での“ヲ”格である。したがって、計算機でこの形式の埋込み文の解析を行なう場合、“変更する”については、この2つの場合だけを調べれば良い。また、埋込み文中の先頭の名詞句と関係をもつことが多い(ただし、これは例d.が全く素直な文と認められるので、制約というよりも、解析結果の順序づけに役に立つ程度の heuristic なものである)。

[4] 多重の埋込み文

(例) a. 食べた料理がわからない人

b. 正直であると私が信じている彼

c. 作成したデータセットが消えた利用者

d. 作成したデータセットを登録しているディレクトリがシステムダウンにより破壊された利用者

例a.の解釈は、“食べた料理が(その人が)わからない人”の解釈のほかに、“(その人が)食べた料理が(パーティの主催者等の第3者が)わからない人”の解釈がある。例b. c. d.はいずれもこの構造をとっているもので、埋込み文によって形容されている名詞が、一段目の埋込み文よりもさらに深い埋込み文の格要素になる構造である。例d.からわかるように、この関係は何段から埋込み文を飛び越しておこなう場合がある。例d.では、形容される“利用者”は、最も深いレベルの埋込み文“作成する”の格要素になっている。この構造を認めることは、解析の段階での並列性を増やすだけでなく、この構造に直接対応する英語表現がないために、発生の段階でparaphraseしなければならない。

[5] 並列の埋込み文

(例) a. 利用者がパラメータで指定するデータを書き出すデータセット

b. 実験の結果得られた水素が燃焼するときが発生する熱量

例a.の“・・・指定する”，例b.の“・・・得られた”はいずれも、直後の名詞句を形容する埋込み文ではなく、次の埋込み文と並列になって、

それぞれ“データセット”，“熱量”を形容する（cf. 埋込み文の並列には，“利用者が指定し，システムが解釈するパラメータ”のように連用中止法によるものもある）。

4.3.6 埋込み構造の解析アルゴリズム

埋込み文はまず単文として解析されるが，単文解析の手順はすでに4.3.4で述べたので，ここでは与えられた埋込み構造が形式1，形式2かの判定や各埋込み構造に対応した中間表現をどのようにして生成するかについて述べる。また，形式2の埋込み構造は，形式1ではないと判定されたときのみ作られる。

〔1〕 埋込み構造の中間表現

図4-20に中間表現を示す。

〔2〕 形式1の埋込み構造の解析手順

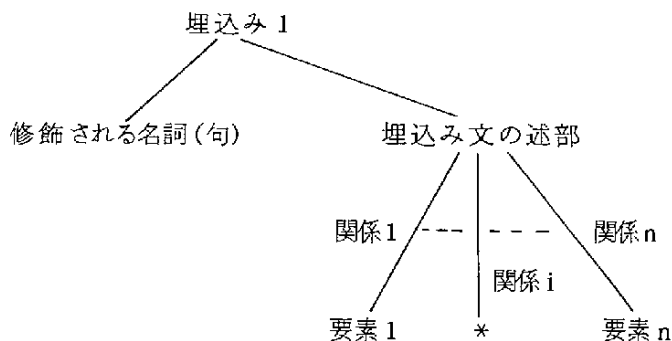
すでに述べたような文体木の構成から，埋込み構造を解析するためのプログラムUMECOMIに制御が戻ってきた時点では，埋込み文もそれによって形容される名詞句もすでに中間表現に整理されている。したがって，プログラムUMECOMIの役割は，埋込み構造の形式の決定と，対応する中間構造の生成，埋込み文を飛び越す文要素の係り先の決定等を行なうことになる。解析は次のように行なわれる。

(1) 修飾される名詞句の中間表現を用いて，格要素型の要素を生成する。

ここで生成する格要素型の要素はダミーの格助詞として“EMBED”を持つ。この“EMBED”は形式1の埋込み構造解析のために本プログラムが特別に用意する格助詞である。

(2) この格要素型の要素を埋込み文の述部の支配範囲に含まれると考えて，埋込み文の解析時に用いた表層格パターンとマッチングさせる。

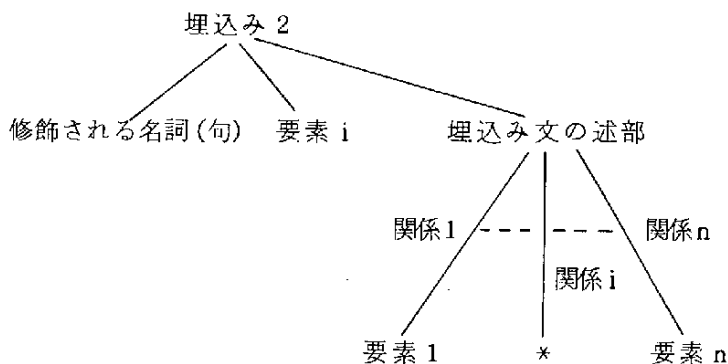
埋込み文で用いられる表層格パターンには，形式1の埋込み構造を取り得る依存関係の制約条件の部分に，(1)で作った特殊な格要素型“EMBED”の要素を取ることを記述する（前述のように，形式1の名詞句を受け入れ



ただし、埋込み 1 は形式 1 の埋込み文であることを示し、* は修飾される名詞(句) が埋込み文の述部と関係 i で結ばれることを示す。

埋込み文が「修飾される名詞(句)」を修飾していることがわかると、上記の中間表現が生成され、もとの「修飾される名詞(句)」の中間表現と置き換えられる。

(a) 形式 1 の中間表現



ただし、埋込み 2 は形式 2 の埋込み文であることを示し、* は修飾される名詞(句) と意味的関係を持つ埋込み文の要素が、関係 i で埋込み文の述部と結びついていたことを示す。

この中間表現も形式 1 のときと同じく、もとの「修飾される名詞(句)」の中間表現と置き換えられる。

(b) 形式 2 の中間表現

図 4 - 2 0 埋込み構造の中間表現

る格は動詞によって限定されているので、動詞の格パターンごとに指定する)。もし条件を満たす依存関係が存在したなら、

- (3) この関係が既に埋込み文中の要素と結び付いていないかどうかを調べる。

埋込み文中に既に結び付いた要素があれば、その関係に対する埋込み構造の解析は失敗となる。

- (4) 修飾される名詞句の中間表現を埋込み文の述部の持つ中間表現の所定の場所に入れ、中間表現を生成する。そして、この中間表現に対して、単文解析の場合と同じく全体的な意味的整合性を調べる。

- (5) 整合性が得られればこの中間表現中の修飾される名詞句を“*”で置き換えた中間表現を生成する。

- (6) 最終的な中間表現として、図4-20(a)の構造を生成する。

もし(3)の段階で複数個の結果が得られると(6)の結果は複数個の可能性があり得る。

[3] 形式2の埋込み構造の解析手順

形式2の解析には、修飾される名詞と意味的關係を持ち得るのはどの格かを辞書中に記述しておく必要がある。どの關係が形式2の埋込み構造をとるかは述部によるから、この記述は表層格パターンと同様に各述部の辞書記述に含まれていなければならない。ここではこれを形式2の關係リストとよぶ。また、埋込み文によって修飾される名詞句と、埋込み文中の格要素間に形式2で許される意味的關係が成立するか否かは、この両者を名詞句内の2つとみなした時に、係り受けの關係が成立するかどうかによって判断する。すなわち、2つの名詞句を“の”でつないだとき、全体として意味をなすかどうかを調べる。手順は次のようになっている。

- (1) 形式2の關係リストに含まれる全ての依存關係についてそれぞれ(2)以下のことを行なう。したがって複数個の解釈が得られることがある。
- (2) 埋込み文の中間表現から、記述されている依存關係に対応した要素を取り出し、修飾される名詞句との間で係り受けが成り立つかどうかを調べる。

(3) 係り受けが成り立てば図4-20(b)の埋込み構造の中間表現を生成する。

[4] 埋込み構造の全体的な流れ

埋込み構造を処理するためのプログラム UMECOMI への入力には次のものがある。

- (1) 出口関数
- (2) チェック関数
- (3) 左下に位置する文体木の部分木
- (4) 右下に位置する文体木の部分木

図4-21にプログラム全体の解析手順を示す。この図に示されているように、プログラム UMECOMI は、まず文体木中の自分の左下についた部分木（埋込み文）と右下についた部分木（主文）の評価を行ない、各々の中間表現形を得る。次に、左下の部分木の評価結果である埋込み文の中間表現と、埋込み文によって修飾されている名詞句の中間表現をもとに、埋込み構造の形式決定を行なう。以下は、単文解析の場合と同様に、支配範囲の解釈等を行ない、全体の構造をまとめることになる。

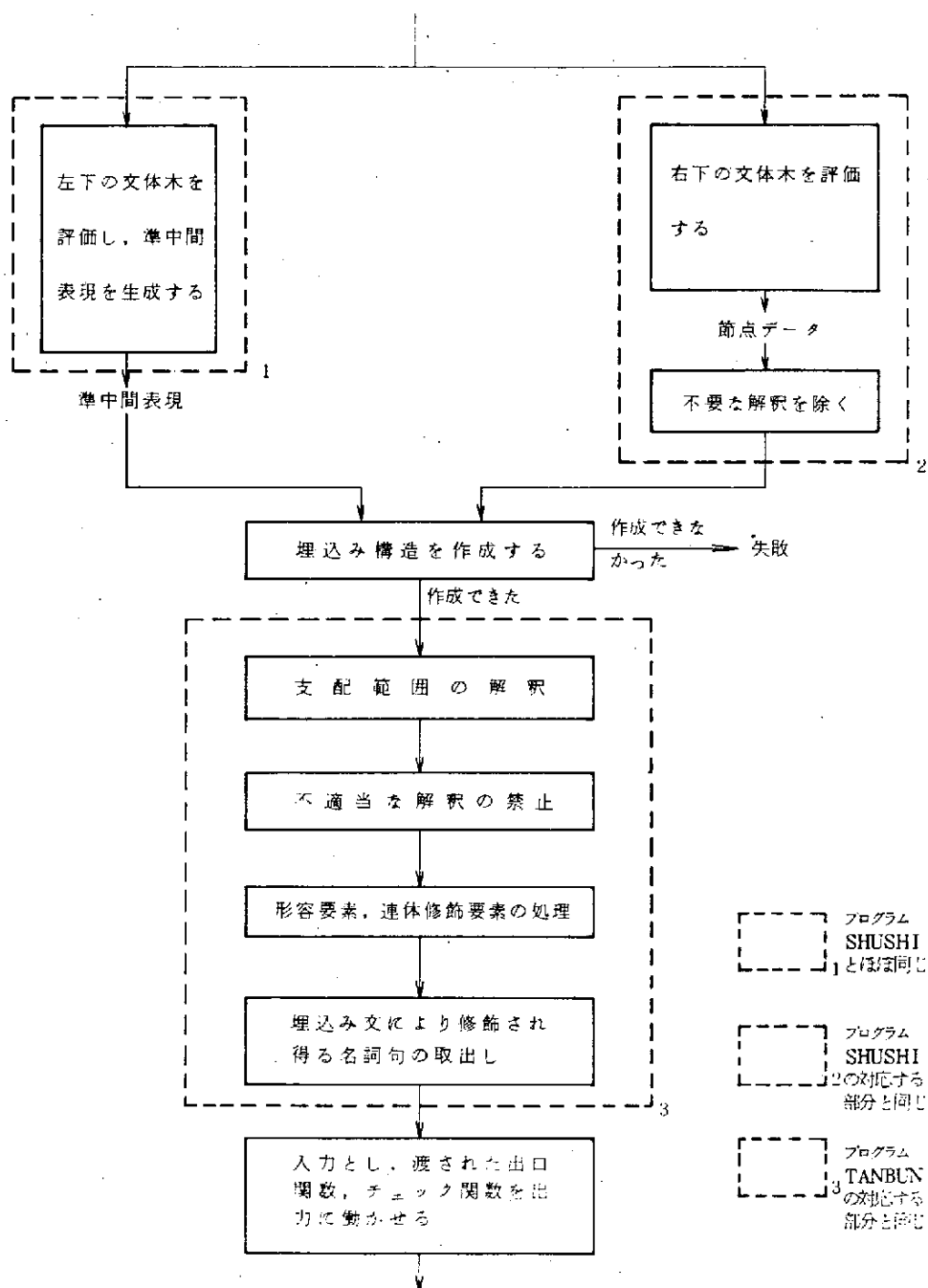


図 4 - 21 埋込み文解析の処理の流れ

4.3.7 日本語解析システムの結果とその評価

本解析プログラムによって処理された典型的な例題を付録に示す。本解析プログラムは、翻訳システムの一部になることを考慮して意味的な処理は簡単なセマンティック・ネットワークと意味マーカを利用する程度にとどめられている。そのかわりに、日本語に現われる統語的な言語現象については、できるかぎり忠実にとらえ、解析に利用するようにした。現在、本解析プログラムで解析できる文の種類は、連体形埋込み文に限られており、連用中止法・条件節等を含む構造は処理できない。しかしながら、このような限定された言語現象を対象とした場合であっても、これまでの日本語解析システムが注目していなかった様々な現象があり、本解析プログラムはこれらの現象についてキメの細かい処理を行なっている。機械翻訳を考える場合、たとえある程度文体を規制するとしても、自然言語によるデータベース・アクセスシステム等と比べると、はるかに多様性のある言語現象を処理対象としなければならない。理想的な日本語表現だけでなく、実際に使われている日本語表現を検討し、それを処理するための機能を付加してゆくことが、今後の日本語解析システムでは不可欠となる。その意味で、今回作成した解析システムは、対象は埋込み構造に限られているが、意味のある試みであると考えている。しかも、文体木構造を全体の処理過程を制御する枠組として使っているので、連用中止法・条件節等の言語現象を処理するためのプログラムを組み込むのは比較的容易である。以下、本解析プログラムの特徴と残された問題点を示す。

[1] 特 徴 点

- (1) 現在のところ扱える文が単文の場合と埋込み文の一部（形式1，形式2）に限られるが、この範囲内においては考え得る解釈は全て出るようになっており、解析プログラムが制限を加えることがない。
- (2) 解析において使用するデータは個々の単語記述や外部から与えるデータとして用意するようになっており、データの変更はプログラムとは独立して行なえるようになっている。このためデータの変更がプログラムの動作

に悪影響を与えることが少ない。また、局所的な関係等は各単語に記述するので、1つの規則の変化が解析過程全体に悪影響を与えることもない。

- (3) 局所的な関係を単文解析等の前の初期の段階に行なっており、単文解析時等の後段の処理が簡単化される。
- (4) 文体木を作成することにより、解析過程が容易に把握でき、新しい専用プログラムの追加も簡単に行なえる。また、各専用プログラムが完全に独立しているため、各プログラムの変更が他のプログラムに不当な干渉を引き起こすことがない。
- (5) 各専用プログラムや表層格パターン等に出口関数、チェック関数が指定でき、これによって中間表現の変更や中間表現の妥当性が検定できる。

[2] 問 題 点

- (1) 単語ごとに解析に使用する情報を辞書記述として与えるため、また、辞書の記述形式が必ずしもコンパクトな形式になっていないために辞書の大きさが大きくなりすぎる。もちろん、この欠点については辞書記述用のマクロ定義等の機能を充実させることによってかなりの程度解消ができると考えられる。
- (2) 解析過程のトレース情報が分かりにくく、データ、プログラムのディバックが不便である。このためのソフトウェア的なサポート機能を今後充実させてゆく必要がある。
- (3) 現在、解析中に複数個の解釈が出てくる場合には、序列が付けられず、全て同一の序列で扱われている。しかし係り受けに強弱があるときや、係り受けの距離に差があるときには、もっともらしい解析結果を優先して出力する等の工夫がいることになる。
- (4) 単文の解析時に単文要素は単に順序任意で省略可として扱っているが、実際には副詞や係り助詞と述部の呼応の関係や標準的な語順など、現在のシステムでは有効に扱えない情報が残されている。
- (5) 省略の問題を扱うためや、文間にまたがって中間表現を変えるような場

合には単なる依存関係構造ではなく、格構造のような一般的意味関係をシステムとして表現しておく必要がある。

(1)、(2)および(4)の各問題は、それぞれ、適当な記述形式を定め、それを実際に解析過程で使う形にコンパイルするようなプログラムを用意すること、ルール記述に使用した汎用パターンマッチングプログラムの機能を強化すること等により解決できよう。残りの問題については、本プログラムの出口関数、チェック関数の指定、表層格パターンへの記述の追加によっても解決可能であり、今後種々の例題を実際に処理してゆくことによって、システムの不備・ルールの追加等を行なってゆく予定である。

4.4 構造変換過程（トランスファー過程）

本節では、日本語中間構造から英語中間構造への構造変換に関して、プログラム構成、構造変換辞書、訳語および構造選択の処理過程について述べる。

4.4.1 構造変換部の構成および手法

構造変換プログラムは、構造変換辞書を参照しながら訳語選択および英語中間構造の生成を行なう。この過程では、訳語選択条件、日本語中間構造の探索、生成する英語中間構造の記述および日本語中間語の意味マークが処理の対象となる。この構造変換過程は、次の点を満たさなければならない。

- (1) 日本語中間表現に関係記述語を含む新しい中間語が追加されても、構造変換辞書の追加を行なうだけで良い。
- (2) 生成する英語中間構造を構造変換辞書の記述を変えるだけで自由に変更できる。
- (3) 訳語選択の条件記述が容易に変更、追加でき、また、任意の条件記述ができる。

これらの条件を満足するよう、本システムでは、次に示す特徴を持った構造

変換プログラムおよび構造変換辞書を作成した。

- (1) 訳語選択条件および英語中間構造に関する記述は、構造変換辞書内に、それぞれ独立に記述している。
- (2) 各中間語に関する構造変換辞書の記述形式を統一し、名詞、動詞といった分類に依存しない辞書構成にしている。
- (3) 各中間語は英単語に対応するのではなく、英語中間構造の部分構造に対応する。また、この部分構造を生成する部分はプログラムとし、任意の英語中間構造に変換できるようにしている。
- (4) 訳語選択条件等を簡潔にするために、日本語中間構造を探索するいくつかの基本的なプログラムを用意している。

図4-22に構造変換処理部分の構成を示す。

日本語中間表現から英語中間表現への変換は、構造変換辞書(4.4.2参照)中に記述された各中間語に対応する部分的な英語中間構造を参照し、日本語中間構造を下位から順に英語中間構造に置き換えてゆくことにより実行される。例えば、“利用者がファイルにデータセットを割り当てる”という文は、図4-23に示す様に構造変換される。図4-24に、“ファイル”、“割り当てる”、“注意する”の3つの中間語に対応づけられている部分的な英語中間構造を示す。図中のslot-nは、その日本語中間語が支配しているn番目の日本語中間構造を英語中間構造に変換した構造が、その位置に埋め込まれるということを示している。日本語中間語の全てが、以上の様な英語中間構造の部分構造に対応づけられているのではなく、時制、様態、アスペクトを示す日本語中間語のように、構造変換後の変形操作によって英語表層上に実現されるものは、この構造変換の段階ではカテゴリー記号Sに対応するノード情報を付加するだけで、英語中間構造自体を変えることはない。これらの情報は、次の英文生成における構造変形の過程で使われることになる。

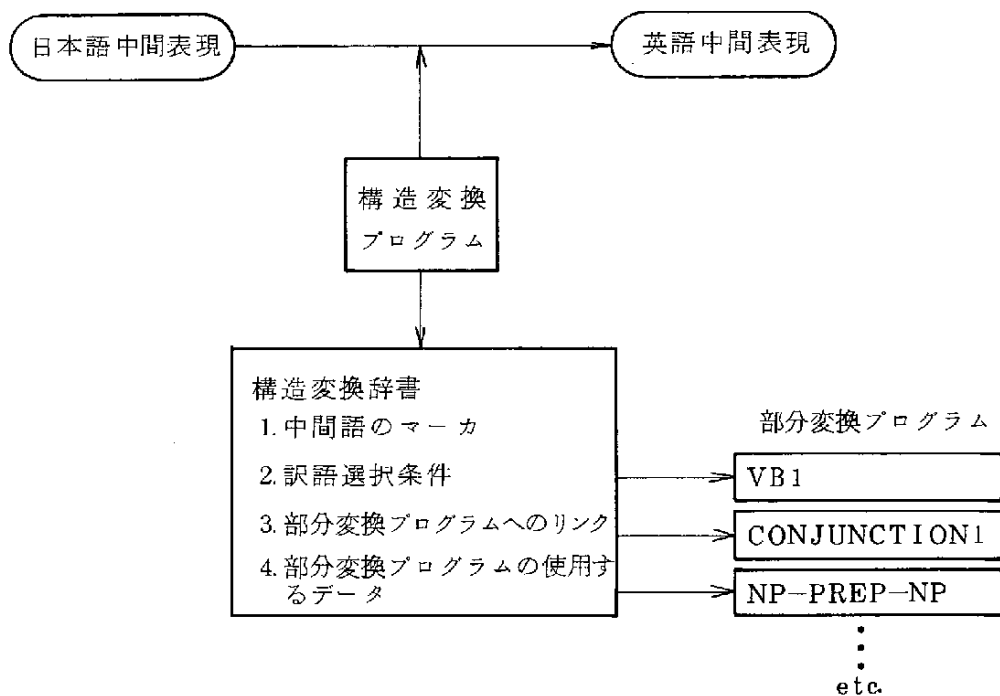


図 4 - 2 2 構造変換部のブロック図

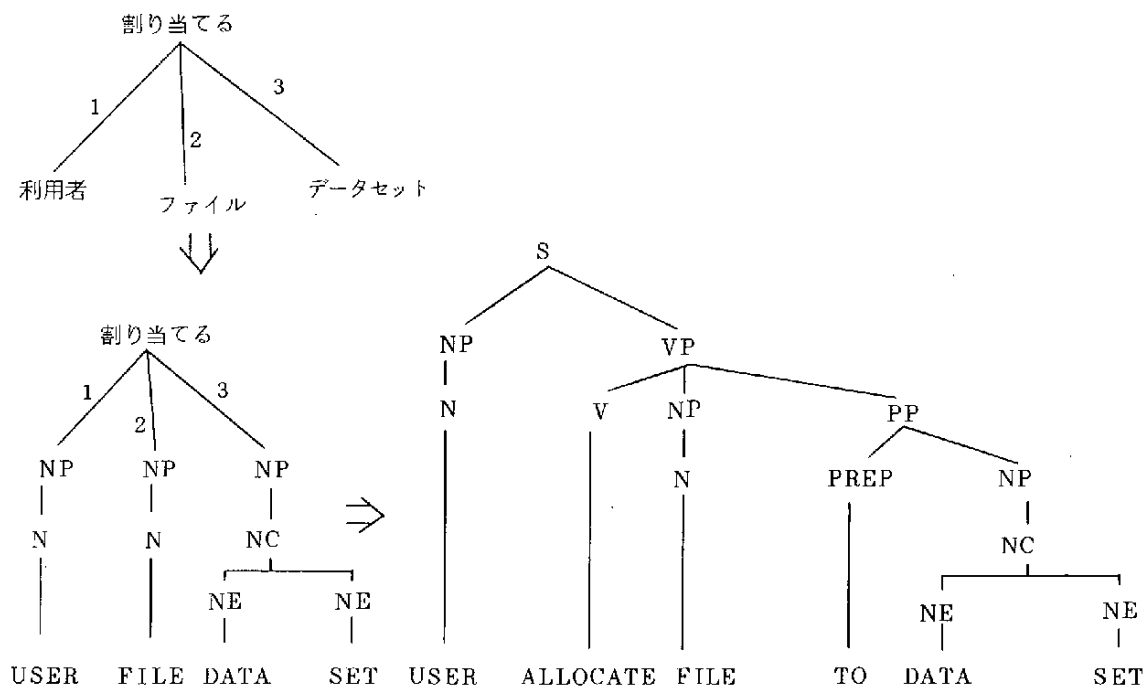


図 4 - 2 3 構造変換の過程

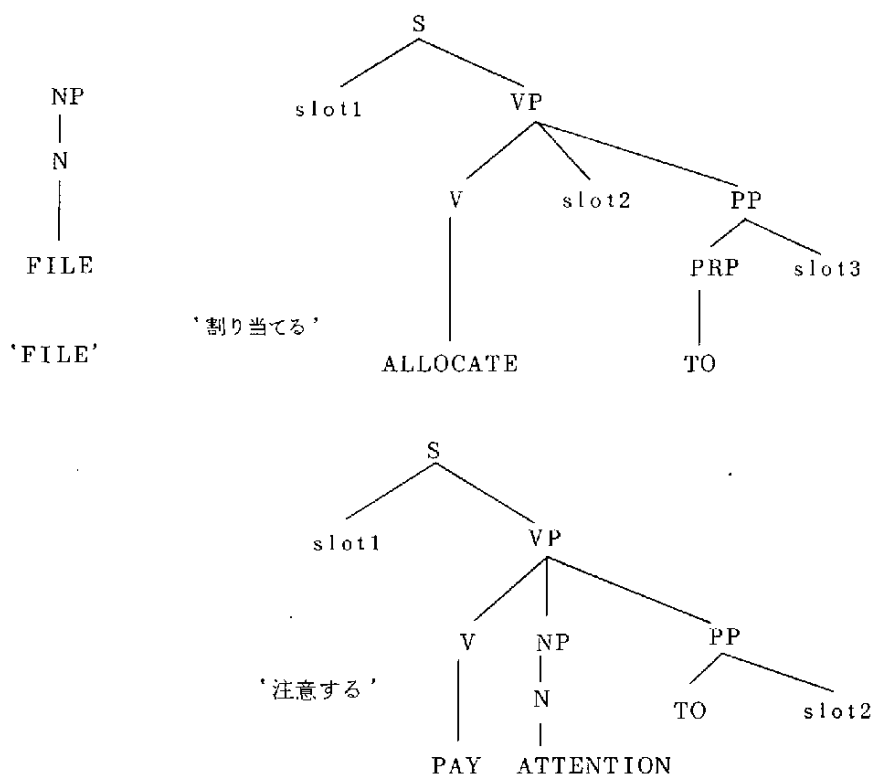


図 4 - 2 4 中間語に対する句構造表現

4.4.2 構造変換辞書および部分変換プログラム

構造変換辞書は、日本語中間語を見出しとした任意長の辞書記述を持つ機械辞書である。図4-25に構造変換辞書の形式をバックス表記で示す。

```

<辞書記述> ::= <項目番号> ( <見出し> <項> . . . <変換記述> )
<項目番号> ::= <正の整数>
<見出し> ::= <任意の LISP アトム>
<項> ::= <任意の LISP S-式>
<変換記述> ::= ( <変換記述要素> . . . )
<変換記述要素> ::= ( <条件式> <変換単位> . . . )
<条件式> ::= <評価可能な LISP S-式>
<変換単位> ::= ( <生成カテゴリー記述> <プログラム名記述> { <パラメータ記述> . . . } )
<生成カテゴリー記述> ::= ( TYPE . <ノードカテゴリー> )
<プログラム名記述> ::= ( PNAME . <プログラム名> )
<パラメータ記述> ::= <任意の LISP S-式>
<ノードカテゴリー> ::= S | NP | VP | AP | PP | QP | AVP | NC | RPP | RP | GPRP | N | NE |
    QUT | QM | ADJ | ADV | CNJ | PRP | PRPE | RPR | RAV | VSA |
    VSP | PUNC | V | AUX | DET | CMP

```

(注) この構文は、BNF記法で記述しているが、次の記法を追加した。

. . . : その左側の超言語変数が1個以上並ぶことを示す。

{ } : 有無を問わないことを示す。

図4-25 構造変換辞書の形式

〈見出し〉は日本語中間語のつづりで、〈項〉は、意味マーカや中間語のカテゴリーなど、その日本語中間語自身に関する各種の情報を記述する部分である。〈項〉は、任意個数並べることができ、日本語中間語に関する情報を必要に応じて追加できる。〈条件式〉は、訳語および英語中間構造を選択する条件を記述する部分である。〈変換単位〉は、呼び出すべき部分変換プログラムとそれへのパラメータを指定する部分である。パラメータは任意であるが、英単語のスペル等が書かれている。現在の所、構造変換辞書中には、477項目が登録してある。中間語のカテゴリー別項目数を表4-6に、構造変換辞書の具体例を図4-26に示す。

表4-6 日本語中間語のカテゴリー別項目表

カテゴリー	項目数
ME	243
JU	87
JO	37
SZ	52
SH	31
KJ	27

```

22870      IVARIATERU
22880      (CAT JU)
22890      ((INCHK 2 'DEVICE)
22900          ((PNAME . VB1) (TYPE . S) (SLOTNUM . 3)
22910          (SUBJ . 1) (OBJ . 2) (PP (3 . TO))
22920          (SPELL . ASSIGN))
22930          ((PNAME . NOM1) (TYPE . NP) (DET . 1)
22940          (PP (2 . OF) (3 . TO) (SPELL . ASSIGNMENT)))
22950      ((INCHK 2 'REGION)
22960          ((PNAME . VB1) (TYPE . S) (SLOTNUM . 3)
22970          (SUBJ . 1) (OBJ . 2) (PP (3 . TO))
22980          (SPELL . ALLOCATE))
22990          ((PNAME . NOM1) (TYPE . NP) (DET . 1)
23000          (PP (2 . OF) (3 . TO) (SPELL . ALLOCATION)
23010          (SLOTNUM . 3))))

```

図4-26 構造変換辞書の例

部分変換プログラムは、日本語中間語に対応する英語中間構造を生成するプログラムである。日本語中間語と部分変換プログラムは、構造変換辞書で結びつけられている。部分変換プログラムは、構造変換辞書の〈変換単位〉と1対1対応し、〈パラメータ記述〉がこのプログラムへのデータとなる。生成する英語部分中間構造は、プログラム内に埋め込まれていても良いし、データとして〈パラメータ記述〉に書いておいても良い。

部分変換プログラムは次の処理を行なう。

- (1) 支配語である日本語中間語に対応する部分変換プログラムは、まず日本語中間表現中で、自分より下位の日本語中間表現を、英語中間表現に変換する。この操作は、部分変換プログラムの中で、再度構造変換プログラムを呼び出すことにより再帰的に実行される。
- (2) 自分より下位の日本語中間表現に対応する英語中間表現が決定されると、部分変換プログラムは、その結果を適切な形にまとめあげて、自分のレベルの英語中間表現を作り出す。
- (3) カテゴリーME（名詞）のような被支配語に対応する中間語については、自分より下位に日本語中間構造を持たないために、再帰的に構造変換プログラムを呼び出す必要はない。この場合には、訳語選択だけを行なって、その結果を自分よりも上位の部分変換プログラムへ戻ることになる。
- (4) 一般に、部分変換プログラムは複数個の英語中間構造を返すので、上記の全過程は、並列的に行なわれることになる。

以上のように、構造変換の中心である部分変換プログラムは、基本的には各中間語ごとに個別的に定義可能であり、単語ごとの例外処理を記述できるようになっている（PNAME部に、そのプログラム名を指定すれば良い）。現在は、表4-7に示す部分変換プログラムが標準的なものとして用意されており、これを使って構造変換辞書は記述されている。

構造変換プログラムは、与えられた日本語中間構造が構造変換辞書中の〈条件式〉を満足しているかどうかを調べ、満足している〈変換単位〉を取り出し、

PNAME において指定された部分変換プログラムを実行し、その結果を返す。
日本語中間構造を一種のプログラム木、構造変換辞書をプログラムの定義体と考えると、構造変換プログラムは、与えられたプログラム木を解釈し、順々に指定された部分変換プログラムを実行してゆく一種のインタプリタと考えることができる。

表 4-7 部分変換プログラム

部分変換プログラム名	生成する部分中間構造	部分交換プログラム名	生成する部分中間構造
VB1	SVOの構文	QUANT-NOM	数量修飾された名詞句
VB2	SVの構文	MODIFIR-QUANT	副詞修飾された数量詞句
VB3	SVCの構文	LOC1	場所を表わす前置詞句
EMBED1	関係節を持つ名詞句	LOC2	起点、場所を表わす前置詞句
EMBED2	関係代名詞 WHOSE を含む名詞句	LOC3	終点、場所を表わす前置詞句
EMBED3	同格 THAT を含む名詞句	TIME1	時を表わす前置詞句
CONJUNCTION1	文、名詞その他の接続	TIME2	起点時を表わす前置詞句
CONJUNCTION2	文間の接続 1	TIME3	終点時を表わす前置詞句
CONJUNCTION3	文間の接続 2	INFORM	ノード情報を付加する
COM-N	名詞句	COMPEN	補文標識
TT	名詞句（但し専門用語）	NO-SN	前置詞句を持つ名詞句
NOM1	動詞の名詞形を用いた名詞句	IN-ORDER	群前置詞 INORDERを持つ前置詞句
ADJ-NOM	形容詞により修飾された名詞句	NOUN-DET	限定詞を持つ名詞句
DETNUM	限定詞句	ADJ2-SLOT	necessary for, enough to の構文
NP-PREP-NP	前置詞を持つ名詞句	*HEADNOUN	同定を示す特殊名詞句

4.4.3 訳語選択条件の記述

機械翻訳では、適切な訳語の選択が常に問題になるが、訳語の決定に必要な情報が日本語文の解析過程でどの程度得られているか、あるいはまた、日本語の多義語が解析の過程でどの程度解消されているか、によって、その困難さの度合いは著しく異なる。訳語の決定は、文の統語構造を参照することにより可能な場合、単語間の意味的な関係を調べることにより可能な場合、知的な推論や文脈情報を活用することにより初めて可能になる場合等があるが、我々のシステムでは、1文単位の翻訳を対象としていること、また分野についての詳細な知識をシステムに組み込んでいないことなどから、統語構造あるいは簡単な意味関係から行なえる訳語の選択のみを現在行なっている。

(例文)

a. どのファイルを消去するか。(図4-27-a)

b. どのファイルを消去しても良い。(図4-27-b)

“どの”という単語は、例文a, bの両方に現われるが、英文に翻訳する場合、例文aでは“which”, 例文bでは“any”と訳し分ける必要がある。この訳語選択は、“どの”の英語中間構造を生成する際に、“どの”の支配語に、“許可”あるいは“可能”等があれば“any”, “疑問”だけがあれば“which”を選択する。

(例文)

c. 出力をラインプリンターに割り当てる。(図4-27-c)

d. ファイルをデータセットに割り当てる。(図4-27-d)

“割り当てる”は、cの場合“assign”, dの場合“allocate”が正しい。“assign”は、装置等の割り当てに用い、“allocate”は、データセット等の概念的なものの割り当てに用いる。この場合、“ラインプリンター”等の装置類の構造変換辞書の<項>に“装置”という意味マーカを、“データセット”等の辞書項目には、“領域”の意味マーカを記述しておき、中間語“割り当てる”の構造変換辞書の<条件記述>の部分に、これらの意味マーカに関する条

件を記述しておき、訳語の選択を行なう。

以上の2つの例からもわかる様に、訳語選択条件は、日本語中間構造を参照することによって行なうことができる。本システムでは、〈条件記述〉の部分はLISPのプログラムになっているため、任意の条件が記述できる反面、そのプログラムが実際にどのような動きをするか知らなければ、訳語の選択条件を理解できないという欠点を持っている。

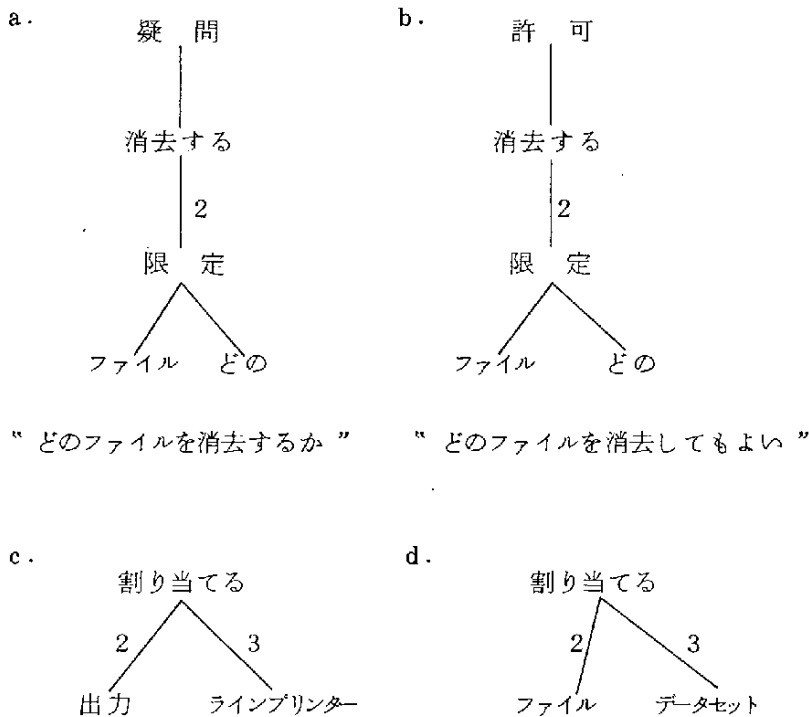


図 4 - 2 7 訳語の選択

次に訳語だけでなく、英語中間表現の構造自体を大きく変更する様な構造変換の例を挙げる。

(例文)

e. 利用者がデータセットの修正を実行する。(図4-28-e)

f. 利用者がプログラムを実行する。(図4-28-f)

g. 利用者がデータセットを修正する。(図4-28-g)

日本語では、“行なり”“実行する”などの補助用言がある。例文eと例文gは、意味的には同じ事を述べている。日本語中間構造は図に示す様に全く異なっている。構造変換過程では、例文eと例文gを、図4-28-e'f'の様に変換し、例文fはf'に変換することが望ましい。本システムでは補助用言に関しては、下位の日本語中間構造が、文であるならば補助用言は英単語として出さない。この他にも日本語固有の表現、例えば“～のは～だ”のような分裂構文があり、統語情報を参照することでどの程度の処理が可能であるか詳細に調べてゆく必要がある。

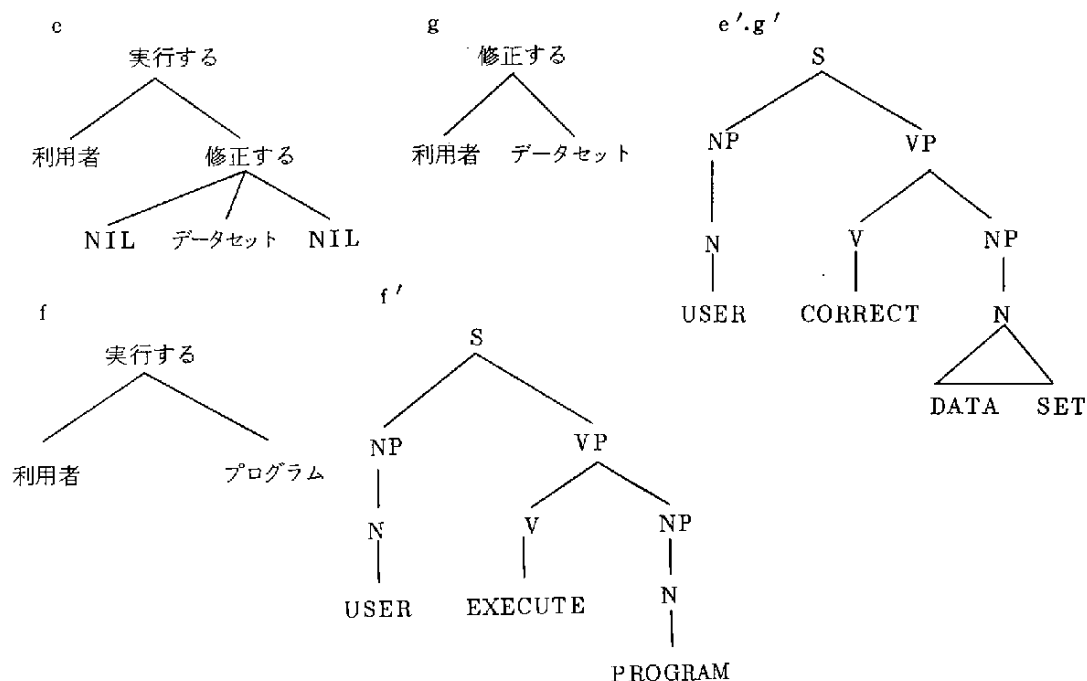


図4-28 補助用言と構造変換

4.4.4 構造変換過程における問題点

日本語と英語の言語体系の違いにより構造変換が非常に困難な場合が少なくない。それらのうち、特に解決しなければならない問題を以下に示す。

(1) 冠詞の付加

英語では、定冠詞“the”，不定冠詞“a, an”が名詞に付いて、主として限定、指定等の情報を表現している。これに対し、日本語では，“この，その”等の連体詞を用いた表現があるが英語の冠詞と異なり，義務的ではない。実際，対象とした計算機マニュアルでは，“この”が35，“その”が18，現われているのに対し，英文マニュアルでは“the”が881，“a, an”が165，現われている。翻訳を行なう時，定冠詞を使うか，不定冠詞を使うか，あるいは冠詞を付けないかという選択は構造変換過程で扱うべき問題であるが，日本語の解析過程において，どこまで情報が抽出できるか等とも密接に関連することになり，現在のシステムでは一応対象外としている。冠詞の問題を解決するためには，最終的に文脈処理や知的な推論が必要であり，今後解決されなければならない課題の1つである。

(2) 時制およびアスペクト

日本語と英語の言語体系としての大きな違いの1つとして時制およびアスペクトがあげられる。4.2.2で述べた様に，日本語中間構造でかなり意味的なレベルになっているが，この時制アスペクトを適切な英語表現に生成するためには，うまく副詞を補ったりする必要がある。マニュアルの分野では，複雑な時制，アスペクト表現は見られなかったが，今後，より広範囲な分野を対象とする場合には，大きな問題点となると言える。

(3) 対応する英語表現が存在しない場合

以下に，日本語表現に直接対応する英語表現が存在しない例を挙げる。

(例文)

- a. プログラムを実行した結果
- b. オペランドの前後

c. 日本一の高速計算機

例文 a は、日本語中間表現で、“埋込み 4”となる例である。例文 b では、“前後”に対応する英単語が無いため、何らかの言い替えが必要である。例文 c も“日本で一番速い計算機”と言い替えなければ翻訳が困難である。これらの例は、単に構造変換過程だけでなく、日本文解析過程も含めて検討してゆく必要がある。

4.5 英文生成過程

本節では、英文生成過程における変形操作、変形プログラム、変形規則の構造および英語辞書について述べる。

4.5.1 英語機械辞書

構造変換過程が終了すると、日本語中間構造に対応する英語中間構造が得られる。英語中間構造は、語順が違っていたり、主語が欠けていたりするため、正しい英文として出力するまでにはかなりの処理を必要とする。

英語中間構造に対して、構造変形過程の最初で各英単語の辞書引きを行なう。英語機械辞書には、名詞や動詞、形容詞等の語形変化の情報、受動態を取り得るか等の文法的な情報が記述してある(図 4-29, 表 4-8)。これらの情報は、後に変形規則が適用される時点で参照される。

動 詞	(entry	PS3	<三单元に対する形>	
		PAST	<過 去 形>	
		PAP	<過去分詞形>	
		ING	<現在分詞形>	
		[AN]		
		PASSIVE-PREP	<前 置 詞>	
		[UNPASSIVE]		
		[STATE]		)
形容詞				
{ 副 詞	(entry	COMP	<比較級形> *MORE	
		SUPER	<最上級形> *MOST	
		[AN]		)
副 詞	(entry	[V-ADV]		
		[VP-TOP]		
		[VP-AFT]		
		[S-ADV]		
		[S-TOP]		
		[S-AFT]		)
名 詞	(entry	PLUR	<複 数 形>	
		[AN]		
		[UNCOUNT]		
		[MAN]		
		[S*S]		)
その他	(entry	[AN]		)

図 4 - 2 9 英文生成のための機械辞書

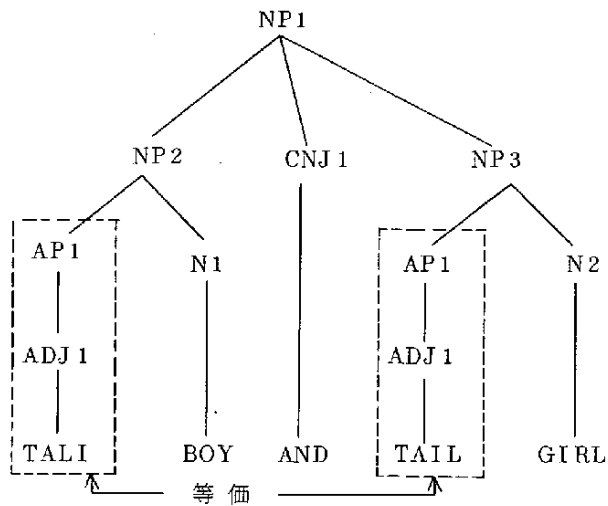
表 4—8 英語辞書に記入される記号表

フ ラ ッ グ	意 味	単 語 例
AN	母音で始まる単語である。 (動詞の形容詞化, 複合語に対処するため名詞・形容詞に必らずすべての品詞に付ける。)	allocation old insert
UNPASSIVE	受身化されない動詞である。	exist
STATE	状態を表わす動詞である。	fix
V-ADV	動詞を修飾する副詞である。	fast
VP-TOP	V-ADVであるが, 動詞の主の前に置かれる副詞である。	already
VP-AFT	V-ADVであるが, 動詞の主の後に置かれる副詞である。	fast long
S-ADV	文を修飾する副詞である。	before
S-TOP	S-ADVであるが, 文の主の先頭に置かれる副詞である。	however
S-AFT	S-ADVであるが, 文の主の後尾に置かれる副詞である。	after before
UNCOUNT	不可算名詞である。	attention
S*S	語尾がsas,ses,sis,sos,susなどの固有名詞である。	Jesus Moses

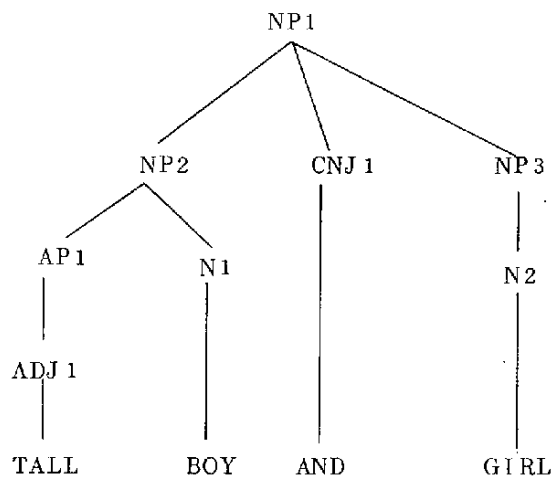
4.5.2 変形操作

[1] 変形ユニットとその処理

英単語の辞書引きが終了した後、英語中間構造から英語表層構造への変換は全て変形規則を用いて行なう。変形規則は、基本的には、木構造がある条件を満たしていた場合、その木構造の部分木の移動や消去を行なう規則である。例えば、同一形容詞句削除規則(EQ1-AP-DELETION)は、図4-30に示す様に、等価な形容詞句を削除する規則である。我々は、計算機上で変形規則を実行するために、変形条件のチェックにパターンマッチングを用い、変形結果を生成するのに木構造を変更するのではなく新たに作り直すという手法(以後、パターン生成と呼ぶ)を用いて、変形規則を手続き的にではなく、記述的に書ける様にした。本システムでは、変形規則は、図4-31に示す形式で記述されており、ファイル内に保存されている。この変形規則を1つの単位として変形ユニットと呼ぶ。変形ユニットには、変形規則の記述と変形規則の適用順序に関する情報が記述されている。



規則適用前の句構造



規則適用後の句構造

図 4-30 EQI-AP-DELETION の例

NUMBER (NAME COM TYPE MP BPL CP PL IAL INAL)

NUMBER : 変形規則の通し番号

COM : 変形規則に関するコメントを記述する部分

NAME : 変形規則名, 最大 80 文字の文字列

TYPE : 義務的なルール (OPTIONAL RULE), 任意的なルール (OBLIGATORY RULE) かの区別 'OB' あるいは 'OP' が入る。

MP : 変形条件となるマッチングパターン (MATCHING PATTERN)

BPL : 変形条件となるプログラムのリスト (BOOLEAN PROGRAM LIST)

CP : 変形の結果として生成するパターン (CREATING PATTERN)

PL : 変形の結果に適用するプログラムのリスト (PROGRAM LIST)

IAL : 変形規則が適用された場合に次に実行すべき変形規則名のリスト (IF APPLIED LIST)

INAL : 変形規則が適用されなかった場合, 次に実行すべき変形規則名のリスト (IF NOT APPLIED LIST)

これらのうち, MP, CP, PL が変形規則自体の記述であり, TYPE, IAL, INAL が, 規則の適用順序の制御を記述する部分である。

図 4-3-1 変形規則の記述形式

〔Ⅱ〕 パターンマッチングとパターン生成

変形規則を記述的に書くという目的のために我々の開発したパターンマッチングには, 次の特徴がある。

- (1) マッチングパターンの AND, OR, NOT の条件記述ができる。
- (2) プログラムによる条件チェックをマッチングパターン中に記述できる。
- (3) マッチングの対象となる木構造の任意の部分木を, 指定した変数にセッ

トすることができ、マッチング終了後も、その変数を参照することでセットされた部分木を処理することが可能である。

(4) 木構造中の任意の部分が同一の構造かどうかをチェックすることができる。

(5) 記述したマッチングパターンの順番に依存しないマッチングができる。また、パターン生成については、次のことが特徴といえる。

(1) 生成する木構造の任意の部分木を、指定した変数にセットすることができ、生成終了後もその変数を参照することでセットされた部分木を処理できる。

(2) プログラムによる部分木の生成を生成パターン中に記述することができる。

パターンマッチングおよびパターン生成の仕様、アルゴリズムの詳細については、省略する。

本システムでの変形規則の適用アルゴリズムを図4-32に示す。プログラムによる条件チェックおよび木構造の変形は、マッチングパターンおよび生成パターンの中にそれぞれ埋め込むことができるが、その場合、パターンの記述が複雑になり、誤りの原因となることが多い。このため、プログラムによるチェックと変形は、マッチングパターンおよび生成パターンとは独立に記述できる構成をとっている。通常、これらのプログラムは、1～数行程度の短いものである。

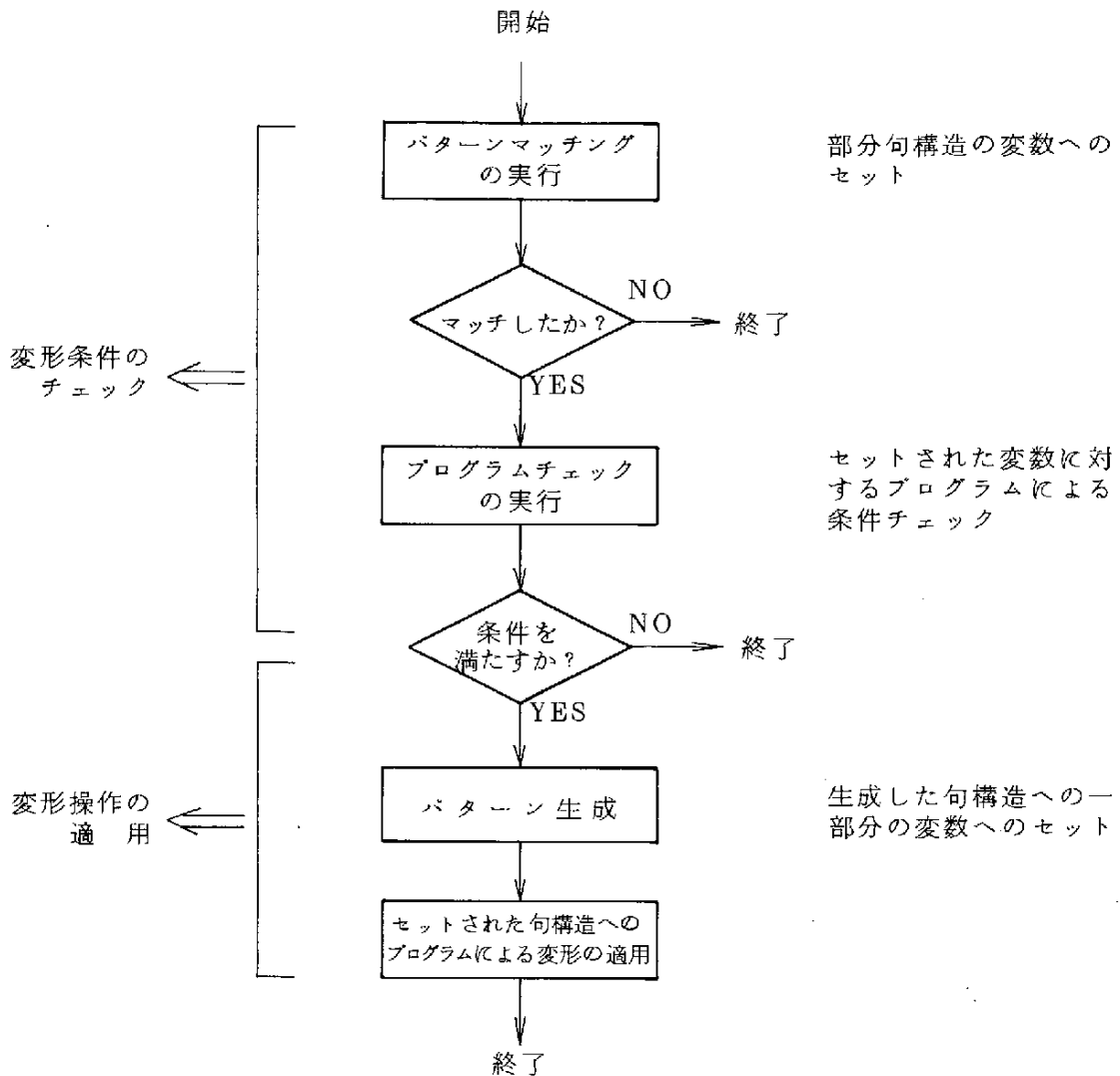


図 4-32 変形規則の適用アルゴリズム

〔Ⅱ〕 変形規則の適用順序の制御

英語中間表現を文法的に正しい英語文として出力するまでには、多数の変形規則が適用されるが、この際の規則の適用順序は、変形ユニット中の IAL および INAL に記述する変形規則名によって制御する。以下では、変形規則の適用順序の説明のため、変形規則を R_n 、変形規則の種類を R の右上に、

I A Lの変形規則との結合を実線の矢印、I N A Lの変形規則との結合を破線の矢印で示した変形規則の流れ図を使用する(図4-33)。流れ図を参照する際、任意的な変形規則がかかった場合には、複数の木構造が生じ、それぞれ続けて変形されてゆくので注意が必要である。

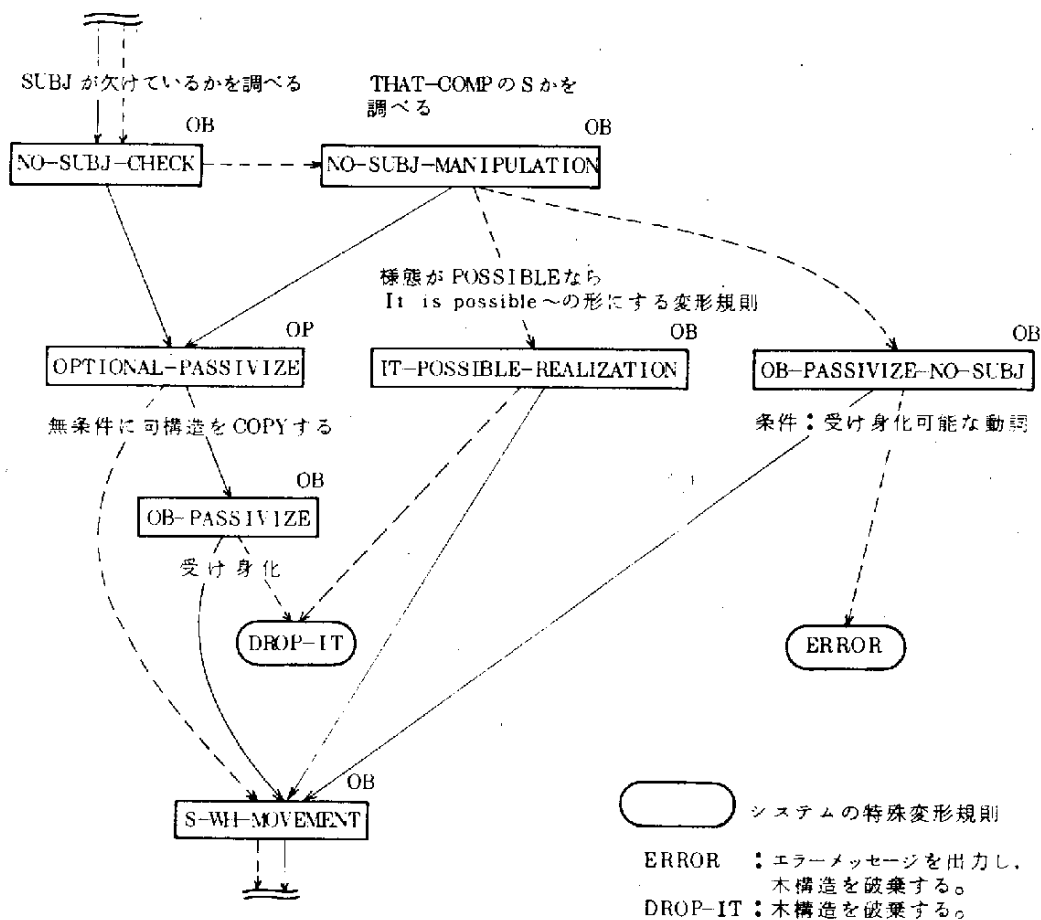


図4-33 変形規則の流れ図(部分)

ある条件 I を満たす木構造があり、この木構造に対して、 $R_1 \sim R_n$ の n 個の変形規則が試みられるという場合を考える。今、条件 I が解決必須の場合とそうでない場合の 2 通りに対して、 $R_1 \sim R_n$ の適用結果を 1 つだけ出す場合と全て出す場合の組み合わせを考える（表 4-9）。この 4 通りを実現する変形規則の流れ図を図 4-34 a ~ d に示す。変形規則 R_0 は、条件 I をチェックするだけの規則、 R_{n+1} は条件 I に無関係な次の変形規則、ERROR はエラーメッセージを出力し木構造を破棄する特殊な規則、DROP は、木構造を破棄する特殊な規則である。図 4-34 a と b は、 $R_1 \sim R_n$ の順序によって R_{n+1} に渡される木構造が異なる。この様に、変形規則の種類と変形規則間の接続を適切に選択することで、種々の木構造を作り出す規則体系を記述することができる。なお、変形規則の流れがループを成しても良いが、その場合は、有限個の規則の適用で、そのループから必ず抜け出す様に注意しなければならない。

表 4-9 規則の適用と並列性

$R_1 \sim R_n$ のパラレル / 条件 I	必 須	任 意
無	case-a	case-b
有	case-c	case-d

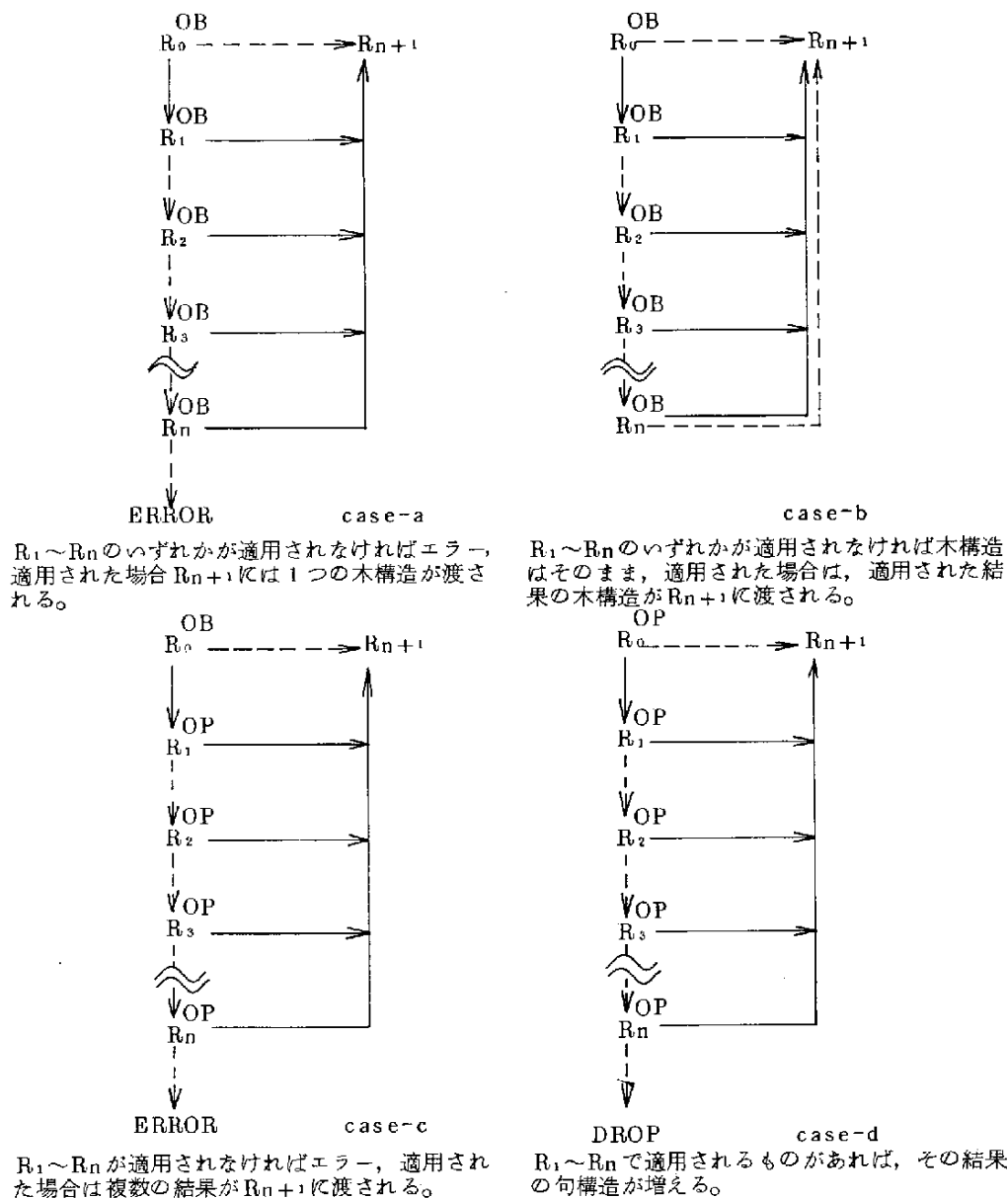


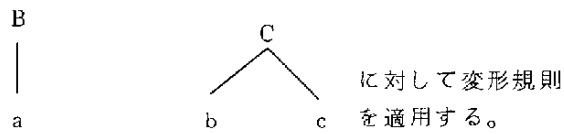
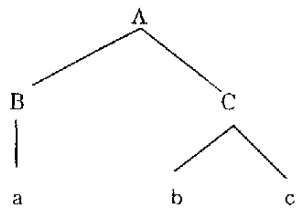
図 4—3 4 変形規則の流れ図

〔Ⅳ〕 構造変形プログラム

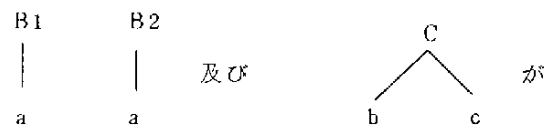
構造変形プログラムは、句構造を入力とし、その全ての部分構造に変形規則を適用するプログラムである。全ての部分構造は、出発点となる変形規則から変形が試みられ、次に適用すべき変形規則が無くなるまで、変形規則の流れに従って次々と変形規則が適用される。この1つの部分構造に変形規則の集合が適用される過程を規則集合の適用と呼ぶ。構造変形プログラムは、図4-35に示す様に下位の部分構造から規則集合の適用を行なってゆく。この過程で1つの部分構造に対して複数個の部分構造が結果として得られるが、この並列の処理は構造変形プログラムが行なう。構造変形プログラムは、1つの部分構造に対して規則集合の適用を行なうプログラム (APPLYTRANSFORM) と、APPLYTRANSFORM の全ての部分構造に適用し、並列の処理を行なうプログラム (TRANSFORM) から成っている。以下では、APPLYTRANSFORMのアルゴリズム等について述べる。

APPLYTRANSFORMは、句構造に変形規則テーブルを参照しながら変形規則を順次適用してゆく一種のインタープリターであり、データ (マッチングパターン、生成パターン) および制御の流れ (変形規則の適用順序) の両者が完全に外部定義になっている。任意的な変形を行なった場合は、句構造中の各カテゴリー記号の一意性を保つために、その生成パターンをコピーするという処理を行なう。この処理により、任意的な変形により生じた句構造と元の句構造が、計算機上で全く別のものとなり、サイドイフェクトが生じないことを保障している。

入力構造



その結果



得られれば、次に

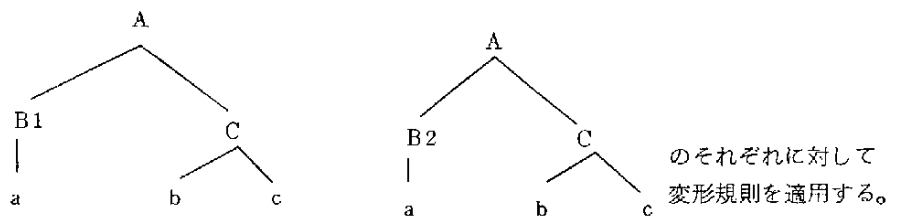


図 4—3 5 構造変形の進行過程

4.5.3 実際の変形規則体系

〔I〕 変形過程の3つのステージ

本システムでは、英語中間構造から英語表層構造までの間に変形規則の3つの集合が適用される。これらは、互いに素な変形規則の集合であり、それぞれの規則適用過程をステージと呼び、ステージ1～3と書く。この様に変形過程を3つのステージに分けた理由として次のことがあげられる。

- (1) 変形規則が下位の部分構造から適用されるため、上位の構造から見た大局的な情報をあらかじめ下位の部分構造に伝達しておく必要がある(Ⅱ参照)。
- (2) 全ての処理を一時に行なうと、規則が複雑になり、文法体系としての見通しが悪くなる。

以下では、ステージ1～3の変形処理について述べてゆく。なお、ステージ1～3のために用意した変形ユニットは、各々、7, 103, 10個である。

〔Ⅱ〕 ステージ1の変形規則

変形規則集合は、下位の部分句構造から順に適用されてゆく。このため、図4-36の2つの句構造に変形を適用する場合、 S_1 以下の句構造と S_3 以下の句構造には全く同じ変形規則が適用され、同じ結果が得られる。しかしながら実際には、図中のa, bはそれぞれ次の様に出力されなければならない。

a' Does program delete file ?

b' User specifies whether program delete file or not.

この様な例を処理するためには、 S_3 以下の部分句構造に変形がかかる時点で、 S_3 が補文標識 'that' に導かれた節であるという情報があれば良い。ステージ1では、 S_3 に 'THAT-CMP' というノード情報を付加する変形操作を行なう。このように、ステージ1では、上位から見た大局的な構造情報を下位のカテゴリー記号にノード情報として付加する操作を行なう。また、関係節の先行詞を関係節の文中の対応する位置に埋め込む変形、文副詞等を

主文の中に下げる変形等が行なわれる。

ステージ 1 は、ステージ 2 の準備段階であり、結果が複数個に増えることは無い。図 4-37 に変形例を示す。

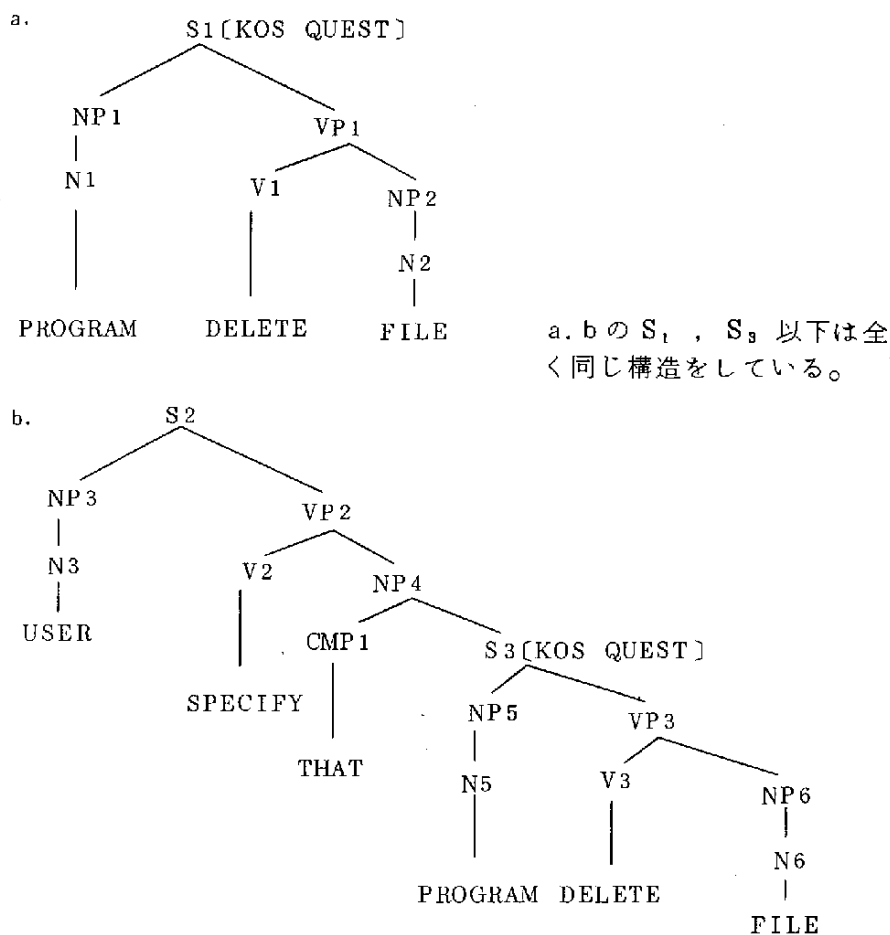


図 4-36 ステージ 1 の変形規則

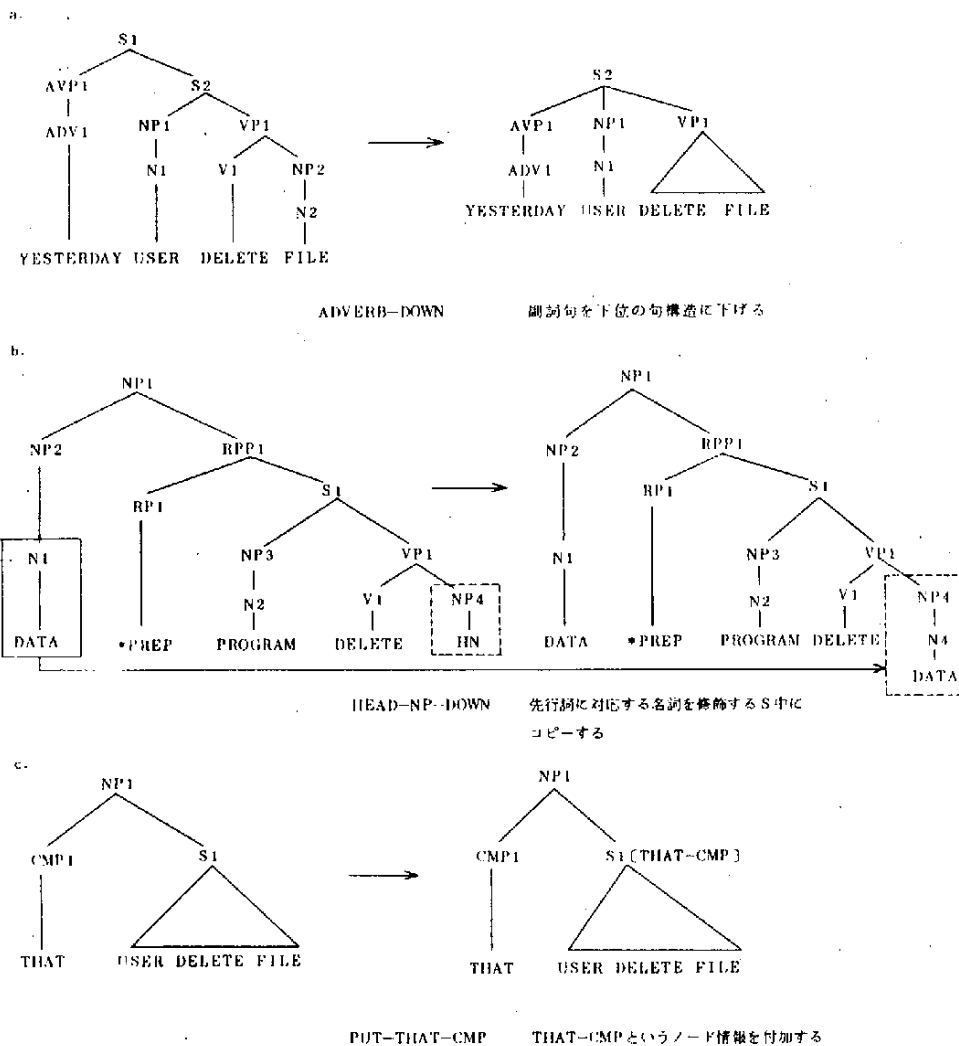


図 4-37 ステージ 1 の変形規則例

〔Ⅲ〕 ステージ 2 の変形規則

ステージ 3 では、英語辞書を参照して、英単語の語尾変化等を行なう。句構造に対する構造変化等を行なわないため、パラレルは生じない。

4.5.4 サポートプログラム

本システムには、モニタープログラムおよび辞書管理プログラムが用意されており、部分変換プログラムや変形規則のデバッグが容易に行なえる。以下では、これらのサポート機能を列挙し、簡単に説明する。

(1) 変形ユニットの構文チェックプログラム

データセット中の変形ユニットに対して簡単な構文チェックを行ない、構文に誤りがあればメッセージを出力する。

(2) トレース機能

変形規則の実行時に、変形規則名、変形適用結果等のトレースメッセージを出力する。

(3) ブレーク機能

指定した変形規則の所で、一旦処理を中断し、ブレイクグループに入る。ブレイクグループでは、端末からコマンドを入力することによって句構造、マッチングパターン、生成パターン、ノード情報、変形規則のステータス等を出力できる。

(4) ルールマップ機能

変形規則のリファレンスマップを出力する。

(5) 辞書管理プログラム

構造変換辞書、英語辞書、変形規則テーブルの追加、変更等が容易にできる。

4.6 英文生成の実験結果と検討

現在英文生成のためには、210個の変形ユニットが用意されており、また日本語中間表現から英語中間表現への構造変換を指示する構造変換辞書には、約500語用意されている。いくつかの実行例を付録に示す。

例題(1)は、形式1の埋め込み文の例で、英文では関係代名詞を用いた表現に

なっている。出力結果が複数個あるのは、関係節内に任意的な受け身化変形が適用されたためである。主文の動詞 BE には、これが受け身化できないことを示すマーカ `unpassive` が生成用辞書に指定されているためである。

例題(2)は、並列接続詞を含み、かつ可能の様態 ' ことができる ' の生成例である。

例題(3)は、目的を表わす中間語 ' ため ' によって2つの文が接続され、さらに主文の中に埋込み文が入っている例である。

以上のように、現在の英文生成システムにおいては一応満足のゆく出力結果を得た。以下では、システムの側面と文法的側面から本システムの整理をする。

(1) システムの側面

- (a) 構造変換では、構造変換プログラム・構造変換辞書及び部分変換プログラムをうまく分離することにより、変換構造の変更・追加が容易になった。
- (b) 構造変形は、パターンマッチング・パターン生成を基本とすることにより、記述的な規則の集合で表現されている。したがって、文法規則の追加・変換がきわめて容易である。
- (c) 変形過程は複数のステージに分割され、生成過程の見通しが良い。
- (d) 辞書や規則をメンテナンスするサポートプログラムが用意されており、規則やプログラムのデバッグが容易である。

(2) 言語・文法的側面

- (a) システム的な整備は一応完了したが、今後様々な例題を考えることにより、適切な日本語中間表現・英語中間表現を設定してゆく必要がある。
- (b) 変換過程では、日本語中間構造の格情報が、英語中間構造での位置に反映される。しかしながら、変形規則の種類によっては、格関係を参照する必要がある場合があり、格マークの体系を整理し、英語中間構造に残しておく必要がある。
- (c) 変形規則を追加・変更し、英文の出力を改良してゆくとともに、最良の文を選択する規則を追加する必要がある。

4.7 今後の方向

今回作成した日英機械翻訳システムは、日本語形態素処理・統語処理・日英中間構造間の変換・英語変形生成処理・英語形態素発生からなる統合的なシステムである。現時点では、この各構成要素について、解決しなければならない問題がある。とくに、日本語の統語処理については、連体形埋込み文に関連した構造の処理だけを取りあげてシステムを作成した。連用中止法・条件節等の複文構造、日本語でよく使われる副詞的表現の処理等、残された課題は多い。今後、これらの処理機構を付加し、より多くの例文を処理することにより完全な機械翻訳システムを作成してゆく予定である。

付録「日本語文の解析結果」

```

TOPO=      8;
メンバーが修正された区分データセットを指定する。
TOPO=      9;
メンバーの修正された区分データセットを指定する。
TOPO=     10;
データをJSEUPDTEプログラムで修正するプログラムを指定する。
TOPO=     11;
利用者が修正するデータを指定する。
TOPO=     12;
利用者が修正されるデータを指定する。
TOPO=     13;
データを修正するプログラムを指定する。
TOPO=     14;
利用者の修正するデータを指定する。
TOPO=     15;
編成が順編成のデータセットを修正する。
TOPO=     16;
編成が順編成であるデータセットを修正する。
TOPO=     17;
区分データセットの指定されたメンバーを修正する。
TOPO=     18;
利用者がデータをプログラムで修正する。

```

```

[ ANATOP (T)
    *** SUN NO=5
    *** HEIRETSU NO=1
    *** BUILD OUTPUT NO1
    *** NAMEWD RESULT NO1

```

*****PART 1*****

```

SHITEISURU
|
+-----+-----+-----+-----+
|           UKEKOMI 2          |         |         | |
|                               |         |         |
|               |             |         |         |
|               +----- KAKO -----+         |         |
|               |                 |         |         |
|               |       UKEMI      |         |         |
|               |                 |         |         |
|               |     SHUSEISURU   |         |         |
|               |                 |         |         |
+-----+-----+-----+-----+
NIL KUBUNDATASEY MEMBER    NIL - "NIL NIL NIL

```

T =
TIME 1228MS.
I

```

I ANATOP (T:
*** SUN NO=6
*** HEIRETSU NO=1
*** BUILD OUTPUT NO1
*** NAMEVD RESULT NO1

```

*****PART 1*****

SHITEISURU

```

      I
      |-----UNEKOMI2-----|
      |                         |
      |                         | KAKO
      |                         |
      |                         | UKEMI
      |                         |
      |                         | SHUSEISURU
      |                         |
      |                         |
NIL KUBUNDASET MEMBER NIL-NIL NIL NIL

```

T =

TIME 1410MS.

I

```

I ANATOP (T:
*** SUN NO=7
*** HEIRETSU NO=1
*** BUILD OUTPUT NO1
*** HEIRETSU NO=2
*** BUILD OUTPUT NO1
*** NAMEVD RESULT NO1
*** NAMEVD RESULT NO2
*** MATCHSTR RULE WASN'T FOUND

```

*****PART 1*****

SHITEISURU

```

      I
      |-----UNEKOMI1-----|
      |                         |
      |                         |
      |                         | SHUSEISURU
      |                         |
      |                         |
      |                         |
NIL PROGRAM *DATA*JSEUPDTEPROGRAM NIL NIL

```

T =

TIME 1783MS.

I

ANATOP (T)

*** SUN NO=9
*** HEIRETSU NO=1
*** BUILD OUTPUT NO1
*** NAMEWD RESULT NO1

*****PART 1*****

SHITEISURU

1
|-----UMEKONI 1-----|
		-----SHUSEISURU		
NIL DATA RIYOSHA " " NIL NIL NIL

*****PART 1*****

SHITEISURU

1
|-----UMEKONI 1-----|
		-----SHUSEISURU		
RIYOSHA DATA NIL " " NIL NIL NIL

T =

TIME 1156MS.

1

ANATOP (T)

*** SUN NO=10
*** HEIRETSU NO=1
*** BUILD OUTPUT NO1
*** NAMEWD RESULT NO1

*****PART 1*****

SHITEISURU

1
|-----UMEKONI 1-----|
		-----UKEMI		
		-----SHUSEISURU		
RIYOSHA DATA NIL " " NIL NIL NIL

T =

TIME 949MS.

1

ANATOP IT:

*** BUN NO=11
*** HEIRETSU NO=1
*** BUILD OUTPUT NO1
*** NAMEVD RESULT NO1

*****PART 1*****

SHITEISURU

|

```
|-----UHEKOH1-----|
|           |           |
|           |           |
|           |-----SHUSEISURU-----|
|           |           |
|           |           |
```

NIL PROGRAM *-DATA-NIL NIL NIL

T =

TIME 966MS.

I

ANATOP IT:

*** BUN NO=12
*** HEIRETSU NO=1
*** BUILD OUTPUT NO1
*** NAMEVD RESULT NO1

*****PART 1*****

SHITEISURU

|

```
|-----UHEKOH1-----|
|           |           |
|           |           |
|           |-----SHUSEISURU-----|
|           |           |
|           |           |
```

NIL DATA RIYOSHA*-NIL NIL NIL

T =

TIME 1100MS.

I

```

***  RUN NO=13
***  RETRETSU NO=1
***  BUILD OUTPUT NO1
***  NAMEWD RESULT NO1

```

SHUSEI SU RU

[illegible]

ANATOP 111

```

*** SUN NO=14
*** HEIREYSU NO=1
*** BUILD OUTPUT NO1
*** NAMEVD RESULT NO1

```

SHUSEI SURU

[illegible]

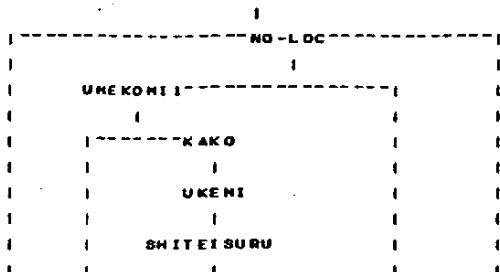
T =
TIME 915MS.
I

ANATOP (T:

*** SUN NO=16
*** HEIRETSU NO=1
*** BUILD OUTPUT NO1
*** NAMEWD RESULT NO1

*****PART 1*****

SHUSE ISURU



NIL MEMBER NIL*-NIL-NIL KUBUNDASET NIL

T =

TIME 1499MS.

I

ANATOP (T:

*** SUN NO=16
*** HEIRETSU NO=1
*** BUILD OUTPUT NO1
*** HEIRETSU NO=2
*** BUILD OUTPUT NO1
*** NAMEWD RESULT NO1
*** NAMEWD RESULT NO2
*** MATCHSTR RULE WASN'T FOUND

*****PART 1*****

SHUSE ISURU

RIYOSHA-DAI-PROGRAM

T =

TIME 936MS.

I ANATOP (T:

*** SUN NO=17
*** HEIRETSU NO=1
*** BUILD OUTPUT NO1
*** NAMEWD RESULT NO1

*****PART 1*****

SHUSE ISURU

NIL-DAI-NIL

T =

TIME 423MS.

I

「英語文の生成結果」

1 EG (1)

ENTER START AND END NUMBERS

R 1 1

SENTENCE-NUMBER NO. 1

-- JAPANESE SENTENCE --

JSEUPDTEプログラムは、ソースプログラムライブラリ
を作成するデータセットユーティリティである。

--- INTERNAL REPRESENTATION ---

*****PART 1*****

DEARU

-----UMEKONI1

-----SAKUSE ISURU

JSEUPDTEPROGRAM DATASETUTILITY = "SOURCEPROGRAMLIBRARY"NIL-NIL-NIL

*** TRANSFER ENDED. PARALLEL= 1

*** TRANSFORM STAGE 1 ENDED. INITRULE= TD-INITIAL, PARALLEL= 1

*** TRANSFORM STAGE 2 ENDED. INITRULE= INITIAL, PARALLEL= 2

*** TRANSFORM STAGE 3 ENDED. INITRULE= NOUN-PHASE, PARALLEL= 2

*** #1 ***

JSEUPDTE program is data set utility which creates source program library

*** #2 ***

JSEUPDTE program is data set utility by which source program library is
created.

*** That's all ***

= NIL

TIME 5010MS.

1

SENTENCE NUMBER NO. 10

-- JAPANESE SENTENCE --

順データセットに対しては、ユーザ見出しラベル及びユーザ終
わりラベルを書き出すことができる。

--- INTERNAL REPRESENTATION ---

*****PART 1*****

KOTOGADEKIRU

|

KAKIDASU

|

NEISHIRETSUI

|

NIL USERMIDASHILABEL USEROWARILABEL JUNDATASET NIL

*** TRANSFER ENDED. PARALLEL= 1

*** TRANSFORM STAGE 1 ENDED. INITRULE= TO-INITIAL, PARALLEL= 1

*** TRANSFORM STAGE 2 ENDED. INITRULE= INITIAL, PARALLEL= 1

*** TRANSFORM STAGE 3 ENDED. INITRULE= NOUN-PHRASE, PARALLEL= 1

*** #1 ***

It is possible to write out user header label and user trailer label to
sequential data set.

*** That's all ***

= NIL

TIME 2710MS.

I

SENTENCE NUMBER NO. 39

JAPANESE SENTENCE

制御データセットのユーザ終わりラベルを処理するため、利用
者が用意するルーチン名を指定する。

INTERNAL REPRESENTATION

*****PART 1*****

TAME1

SHORISURU

SHITE ISURU

NO-HAS

UNEKONI

YOISURU

NIL USEROWARILABEL SEIGYODATASET NIL NIL ROUTINEMEI RIYOSHA* NIL NIL

*** TRANSFER ENDED. PARALLEL= 1

*** TRANSFORM STAGE 1 ENDED. INITRULE= TD-INITIAL, PARALLEL= 1

*** TRANSFORM STAGE 2 ENDED. INITRULE= INITIAL, PARALLEL= 3

*** TRANSFORM STAGE 3 ENDED. INITRULE= NOUN-PHASE, PARALLEL= 3

*** #1 ***

Routine name provided by user is specified in order to process user
trailer label of control data set.

*** #2 ***

Routine name which user provides is specified in order to process user
trailer label of control data set.

*** #3 ***

Routine name which is provided by user is specified in order to process
user trailer label of control data set.

*** That's all ***

= NIL

TIME 7797MS.

I

R 28 28

SENTENCE NUMBER NO. 28

-- JAPANESE SENTENCE --

JSEUPDTE プログラム実行のために必要なジョブ制御文
は、EXEC 文である。

--- INTERNAL REPRESENTATION ---

*****PART 1*****

DEARU

UMEKON11

HEISHIRETSU1

HITSUYODA

JIKKOSURU

JOBSEIGYOBUN

* NIL JSEUPDTEPROGRAM EXEC2UN SYSUT1DDBUN SYSUT2DDBUN

*** TRANSFER ENDED. PARALLEL= 1

*** TRANSFORM STAGE 1 ENDED. INITRULE= TD-INITIAL, PARALLEL= 1

*** TRANSFORM STAGE 2 ENDED. INITRULE= INITIAL, PARALLEL= 2

*** TRANSFORM STAGE 3 ENDED. INITRULE= NOUN-PHRASE, PARALLEL= 2

*** #1 ***

Job control statements necessary for the execution of JSEUPDTE program
are EXEC statement, SYSUT1 DD statement and SYSUT2 DD statement.

*** #2 ***

Job control statement which is necessary for the execution of JSEUPDTE
program are EXEC statement, SYSUT1 DD statement and SYSUT2 DD statement.

*** That's all ***

= NIL

TIME 6675MS.

1

SENTENCE NUMBER NO. 02

-- JAPANESE SENTENCE --

ラベルを記述した場合、このラベルの後に、1つ以上のブランクを置かなければならない。

--- INTERNAL REPRESENTATION ---

*****PART 1*****

```

      BAAI
      |
  KANRYO-----NAKEREBANARANAI
      |
KIJUTSUSURU      OKU
      |
|-----|    |---QUANTIFY-----NO-ATO
|   |   |   |   |   |
|   |   |   |   |---QUANT-MOD    DET
|   |   |   |   |   | |
|   |   |   |   |   |
|   |   |   |   |---SU---|-----|
|   |   |   |   |   |   |
NIL LABEL NIL NIL BLANK      IJO LABEL KONO

```

*** TRANSFER ENDED. PARALLEL= 2

*** TRANSFORM STAGE 1 ENDED. INITRULE= TD-INITIAL, PARALLEL= 2

*** TRANSFORM STAGE 2 ENDED. INITRULE= INITIAL, PARALLEL= 2

*** TRANSFORM STAGE 3 ENDED. INITRULE= NOUN-PHASE, PARALLEL= 2

*** #1 ***

When label was described, more than one blank must be placed behind this label.

*** #2 ***

When label was described, more than one blank must be placed after this label.

*** That's all ***

= NIL

TIME 7786MS.

I

-- JAPANESE SENTENCE --

--- INTERNAL REPRESENTATION ---

UKEN I
I
KIJUTBUSURU
I

```

      UNK ON I1 ----- |-----|
            |             |       |
      -----UNKI        |       |
            |             |       |
            KUGIRU         |       |
            |             |       |
KEYWORDPARAMETER NIL *-CONNA OPERAND NIL

```

*** TRANSFORM STAGE 1 ENDED. INITRULE= TD-INITIAL. PARALLEL= 1

*** TRANSFORM STAGE 2 ENDED. INITRULE= INITIAL, PARALLEL= 4

*** TRANSFORM STAGE 3 ENDED. INITRULE= NOUN-PHASE, PARALLEL= 4

*** 21 ***

Keyword parameter delimited by comma describes operand.

*** 2 ***

Keyword parameter which is delimited by comma describes operand.

*** 23 ***

Operand is described by keyword parameter delimited by comma.

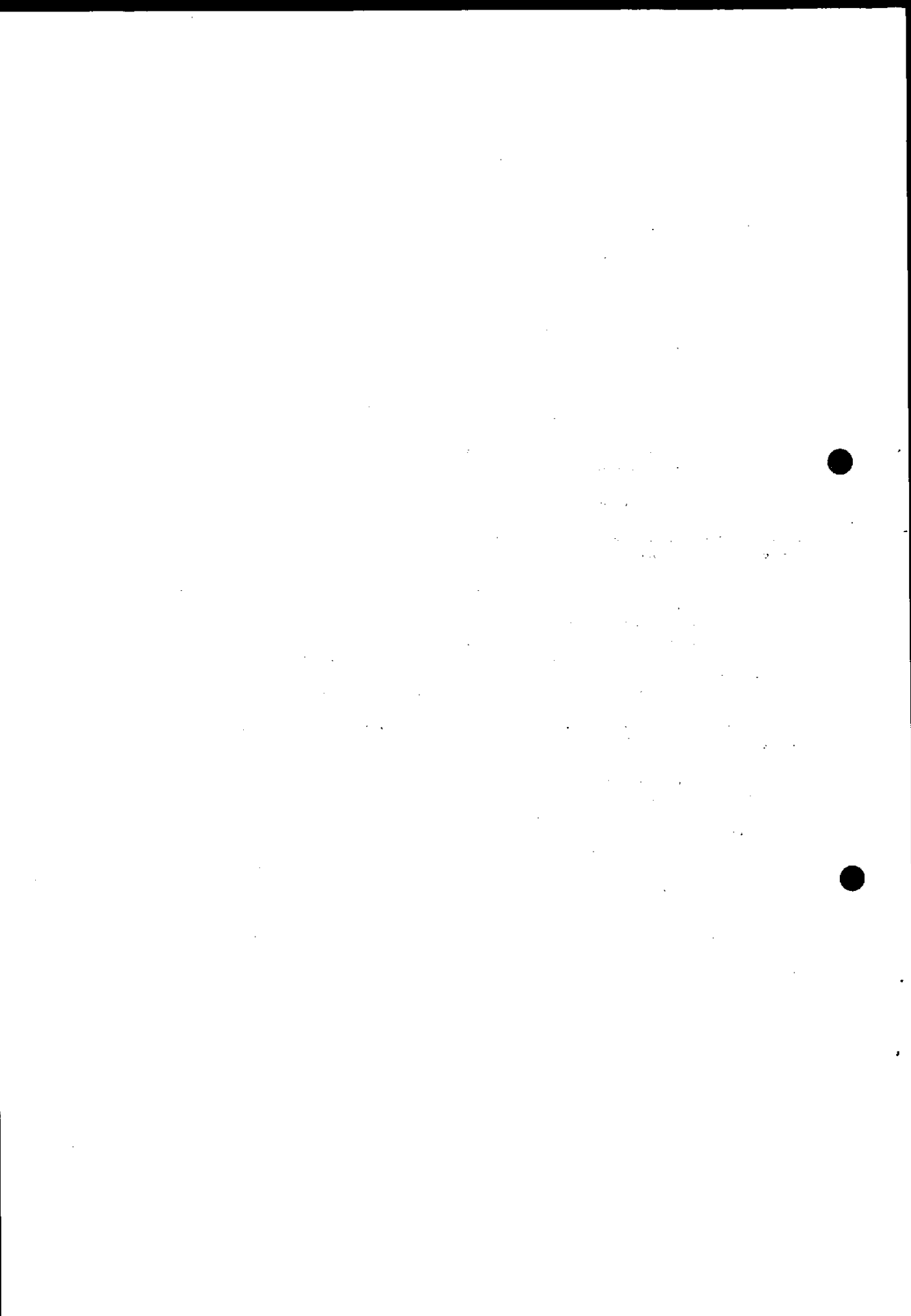
*** 附 ***

Operand is described by keyword parameter which is delimited by comma.

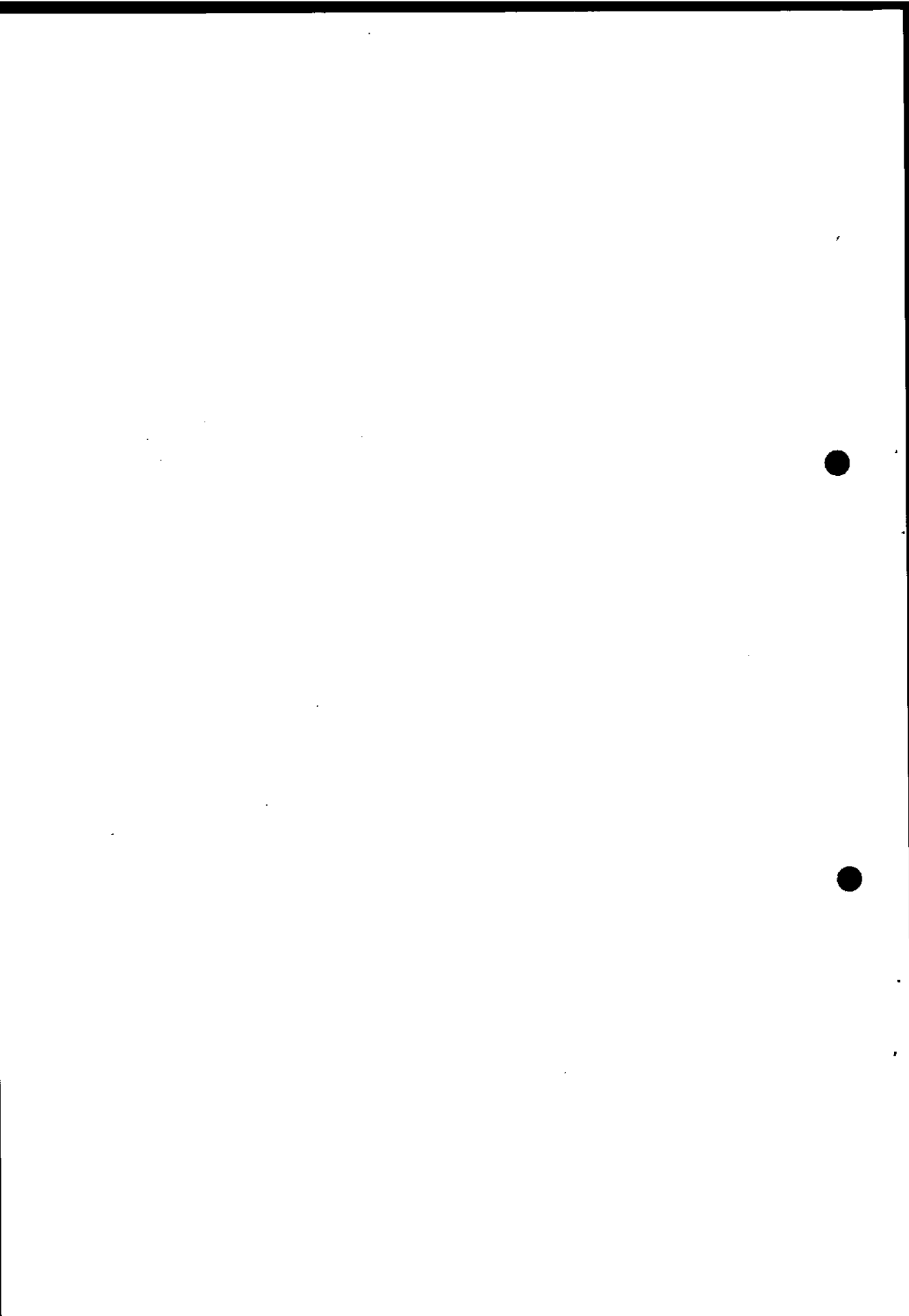
*** That's all! ***

= NIL

TIME 6347 HRS.



5. 言語処理のためのソフトウェア



5. 言語処理のためのソフトウェア

機械翻訳システムを系統的に開発し、新しい言語現象が観察されたときに適切にそれをシステムに反映してゆくためには、文法規則・辞書記述といった言語データと、それを活用して実際に文解析・文生成を実行するシステムとが明確に分離されていることが望ましい。すなわち、文法規則・辞書記述はあらかじめ完全なものを用意しておくことが不可能であるため、翻訳実験をくり返してゆく過程で、記述の不備を修正し、evolutional にシステムを作成してゆくことになる。

文法規則・辞書記述等は、入力文を解析し、その解析結果をもとに出力文を生成するというメカニズムを定義しているという意味で、プログラムの役割を果たすと同時に、部分的な変更・追加が随時行われるデータとしての性質も備えていることになる。第1世代の代表的な機械翻訳システム SYSTRAN においては、この言語データとシステムとの分離が不十分で、SYSTRAN Macro と呼ばれる一種のプログラミング言語を使用して、直接文法規則に相当するものまでプログラムとして表現している。SYSTRAN は全体としてはサブルーティンの集まりで、文法規則を記述していると考えてもよい。このために、新たな文法規則を追加したり、既存の文法体系を修正しようとするとき、ある場合にはシステム全体のプログラムの再検討からはじめなければならない、ある一定以上の規模の文法をインプリメントする際の大きな障害になっているといわれている。この欠点は、SYSTRAN では、文法規則や辞書記述といった言語データが、直接システムプログラムの中に埋め込まれているためである。

このような欠点を補うために、第1世代以降の機械翻訳システムの研究においては（一般に自然言語処理研究全体においても）、文法規則とそれを適用するメカニズムとを分離するためのソフトウェア・ツールの開発に大きな力が注がれている。このようなソフトウェア・ツールは次の2つのカテゴリーに分け

て考えることができる。

1. 文法記述用システム（言語）

2. サポート用システム

ここで、2のサポート用システムという形でまとめたものは、pre-editing・post-editing用のテキストエディタ、辞書メンテナンス用の各種ユーティリティ・プログラム等を指し、一般にWord Processing 技術と呼ばれているものである。しかしながら、この種のソフトウェアは対象とする機械翻訳システムのハードウェアを含むシステム構成に依存したものになり、また SYSTRAN, GETA, TAUM 等の現行の機械翻訳システムにおいては、まだほとんど実現されておらず、今後の研究に待つところが大きい。従って、本章では主として1のソフトウェアについて考察する。

1の文法記述用言語は、自然言語文の解析や生成のための文法を記述するためのシステムであり、これはさらに次の2つに下位区分される。

- 1 a. 解析用文法を記述するためのシステム（言語）

- 1 b. 汎用の木構造変換規則を記述するためのシステム（言語）

1 aの枠組に属するものとしては、LINGOL・拡張LINGOL・ATN (Augmented Transition Network)・APS (Augmented Phrase Structure) 等があり、1 bに属するものとしては、PLATON・ROBRA・SYSTEM-Q・REZO 等が考えられる。1 aは一般に parser と呼ばれるシステムであるが、機械翻訳においては、単に入力文を parsing (構文解析) するだけでは不十分で、その結果を相手言語の構造へと変換したり、parsing の場合とは逆に木構造から単語の並びへと変換する必要もある。こういった一連の処理は木構造の変換プロセスと捉えることができるため、機械翻訳のための文法記述用システムとしては、1 bの枠組をとるものが多い。従って、ここでは1 aに属するシステムについて簡単に触れ、その後1 bのシステムについて少し詳しく考えることにする。

5.1 構文解析用のシステム

自然言語文、特に日本語文の解析アルゴリズムについては、佐藤等が優れた解説を行っている〔佐藤 1979〕のでここではその詳細は述べないが、自然言語の文法を形式言語の文法と同様に考え、オートマトン理論の分野で良く研究されている形式言語の構文解析アルゴリズムをほぼそのまま自然言語の構文解析に適用することが多い。特に、形式言語の中でも構文解析アルゴリズムが最も良く研究され、しかも具体的システムとして実現されているのが文脈自由型文法 C F G である。従って、自然言語の文法規則もこの C F G を用いて記述するのが一般的である。しかしながら、自然言語文の構文解析においては、

1. 自然言語の曖昧さの取扱い
2. 統語情報だけでなく、意味情報の取扱い等、文法規則では表現できない情報の取扱い
3. 変形規則の取扱い

が問題となり、単なる形式言語の構文解析アルゴリズムをそのまま適用するだけでは不十分である。多くの自然言語処理用の構文解析システムにおいては、このため、C F G の文法記述をもとに、形式言語の構文解析システムには見られないいくつかの付加的な機能を設けている。

形式言語、特に C F G の文法を記述する枠組としては、大きく分類すると、書換え規則による方法と、その言語に属する文を受理するオートマトンで表現する方法の 2 通りの方法がある。新しい規則の追加や既存の規則の変更を、他の規則との関連を一応無視して、独立に行えるという意味からは、書換規則による方法が優れており、文法規則のデータとしての性質がより強く反映されることになる。これに対して、オートマトンを定義する方式においては、文法規則を一連の状態遷移というコントロールの流れの中で捕えることになり、文法規則の修正・変更を行う場合も、このコントロールの流れを意識しなければならない。逆に、文法記述者が文法を定義する時点で、いつ、どのような状況で

その規則が適用されるかが意識できるため、きめ細かい処理が可能になっている。

5.1.1 書換え規則によるシステム —

LINGOL, 拡張LINGOL, APS

書換え規則による文法記述を、自然言語文の構文解析に適用する代表的なシステムとしては、MITのPrattによって開発されたLINGOL[Pratt 1973, 1975], これを改良し、日本語文の分ち書きを行いながら構文解析を行ってゆく等の機能を付け加えた拡張LINGOL[田中 1977], この拡張LINGOLに意味的妥当性の評価を行いつつ、複数個の解析結果を並列に保持しながら構文解析を行ってゆくMELING[白井 1980], LINGOLの2分木による解析を多分木に拡張したシステム, 句構造文法の枠組に意味的なチェック機能を追加して, IBMで開発されたAPS (Augmented Phrase Structure) [Heidorn 1972]等がある。

CFGにもとづく構文解析アルゴリズムは、解析の結果つくられる構文木の作り方により、top-down / bottom-up, serial / parallelの区別がある。top-down法は木を上から作り、bottom-upは木を下から作る。一度の構文解析により、serial法は一つの木を作ろうとし、parallel法はすべての可能な木を作ろうとする。一般的によく使用されるのは、top-down serial または bottom-up & parallelの組み合わせであり、京大で開発された機械翻訳システムは、基本的にはbottom-up & parallelのアルゴリズムを使用していた。

基本的にはbottom-up法で、木を下から生成してゆくが、同時にtop-down的な手法を並用して、下から延びて来た部分木が最終的な構文木にまで到達するかどうかをチェックしながら構文解析を進めてゆく手法に、Earleyのアルゴリズムがある。前述のLINGOL系のシステムは、すべてのこのEarleyのアルゴリズムの改良形であるPrattのアルゴリズムを用いて

いる。

前述したように、純粋なC F規則の集合だけでは、自然言語の文法を記述することはできない。以下では、拡張LINGOLを例にとって、書換え規則にもとづくシステムがどのような形式で、書換え規則の表現力を増強しているかを考えてみることにする。

拡張LINGOLの文法記述形式は次のようになっている。

($\langle \text{left} \rangle \langle \text{right} \rangle (\langle \text{advice} \rangle \langle \text{cog} \rangle) \langle \text{sem} \rangle$)

ここで、 $\langle \text{left} \rangle$ と $\langle \text{right} \rangle$ の対が、C F規則

$\langle \text{left} \rangle \rightarrow \langle \text{right} \rangle$

を表現している。ただし、拡張LINGOLにおいては、C F規則の形式に制限があり、 $\langle \text{right} \rangle$ には高々2つの終端もしくは非終端記号が許されるだけである（いわゆるC F規則をChomsky Normal Formに変換しておくことを意味する。すべてのC F文法はChomsky Normal Formに従った規則で表現できることが証明されているので、これ自体で文法表現の記述力が制限されることはない。ただしChomsky Normal Formで文法を記述することを強制することは、文法記述者に対して、ある場合には、ダミーの、無意味な非終端記号の使用を強制することになり、直観的な文法の理解を困難にする可能性がある。このC F規則を表現する $\langle \text{left} \rangle$ 、 $\langle \text{right} \rangle$ 以外の要素は拡張LINGOLにおいて、自然言語処理用に導入された部分である。

〔I〕 曖昧さの取扱い

自然言語における曖昧さは構文解析過程において、表層上の同一の単語並びに対して、複数個の異なった構文木が生じることに対応する。拡張LINGOLにおいては、同一の単語並びに対して、複数個の構文木が生成されると、その各々の構文木を構成するとC F文法規則の $\langle \text{log} \rangle$ 部分のS式（LISPの評価可能な関数）が評価され、各部分構文木の妥当性を示す数値が返されることになる。図5-1に、曖昧さのある場合の状況を示す。

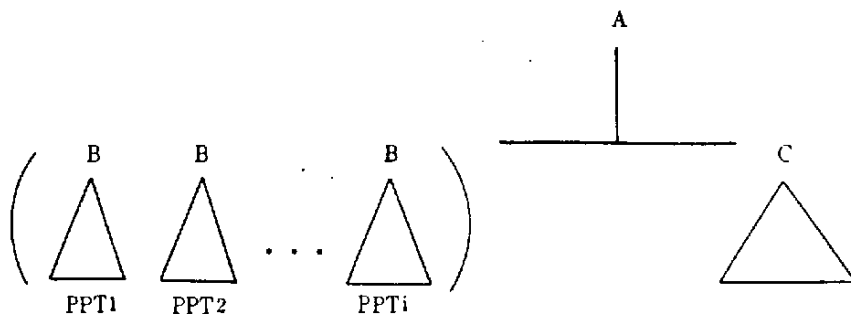


図 5-1 曖昧な木構造

ここで、PPT1, PPT2, ……PPTi と示された各部分構造木についての妥当性が、各CF規則中に記述された<cog>部を評価することによって決定され、このうち最も妥当性が高いと判断された構造木が採用されることになる。

この<cog>の機能は解釈に曖昧さが生じた場合に、CF規則で表現される統語規則以外の、意味的な評価基準等を用いてそれを解消するための枠組である。しかしながら、後述するように、拡張LINGOLにおいては、本格的な意味処理（統語構造から意味構造への写像etc）は、規則記述中の<sem>部で指定され、この<sem>部の評価は、CF規則による構文解析が完全に終了した後に行われる。このため、構文解析の途中での曖昧さを解消するために使われるこの<cog>部の記述では、本格的な意味処理を実行することができず、ある意味で不完全である。事実、「複雑な意味処理を<cog>部で行わないことが、拡張LINGOLの上手な使用法である」というコメントがシステム製作者によってつけられている。

この拡張LINGOLの欠点をMELING〔白井 1980〕では<cog>部において本格的な「意味解釈」まで行うことによって解決している。

〔II〕統語規則以外の情報の活用

自然言語表現を規制する制約条件としては、CF規則で表現される統語

条件の他に、意味的・文脈的な条件がある。このような統語条件以外の様々な制約条件もすべて、統語条件と同じに取扱い、C F 規則として表現してゆこうとする意味文法の試み〔Hendrix 1978〕もある。しかしながら、この方式では統語的には全く同じ振舞いをする語や句であっても、意味的に異った性質をもつ場合には、すべて別の非終端記号として用意しなければならない。非終端記号の増大と、それに伴うC F 規則の増大を招くことになる。従って、この意味文法の考え方は統語的な複雑さの少ない自然言語によるデータベースアクセスシステムのような場合には、適用可能であろうが、機械翻訳システムのように、あらゆる統語的な現象を対象としなければならないシステムには不向きであろう。

以上の事情は統語的条件と意味的条件を同一のC F 規則の中に埋め込んだ場合についていえるだけでなく、統語的条件と形態素的条件とを同一の枠組で取扱おうとする場合にも生じる。

拡張LINGOLにおいては、このような無意味なC F 規則の増大を防ぐために、パーザの動作をC F 規則によって表現される情報（統語情報）以外の情報（意味情報、形態素情報）によって、動的に制御するための枠組が用意されている。これが〈advice〉部の役割である。この〈advice〉部は、C F 規則を実行する時点でいつも参照・実行されることになる。

〔Ⅲ〕変換規則、意味解釈の実行

C F 規則によって作られる木構造は、表層上の語順を保って、それをより高次のまとまりへと整理したものにはすぎない。従って、この木構造はあくまで単語間の係受け構造を表示しているだけで、入力文の意味構造ではない。統語的な構造から意味構造への写像を記述するのが、〈sem〉部の役割である。〈sem〉部の記述は必ずしも統語構造から意味構造への写像を指定するのに使われるだけでなく、一般に統語構造をもとにして、別の構造を作る作業一般に使用できる。Pratt がLINGOL の使用例として示した簡単な英仏翻訳システムでは、その〈sem〉部は英語を仏語に変換

する目的に使用されている。

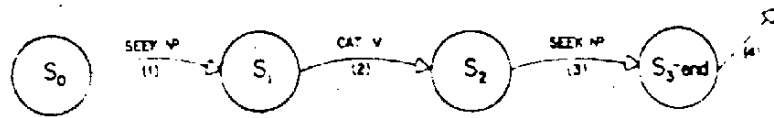
最近の変形生成文法においては、受身文等を含む多くの構造が「意味解釈」規則によって説明されているので、この<sem>部の記述をうまく使うことによって変形された構造をもとの形に戻す作業も実行できよう。しかしながら、純粹に変形規則としてしか定式化の方法がない「wh - 移動の規則」等を拡張LINGOLの枠組でどうわかり易く表現するかは、今後の問題となろう。

5.1.2 オートマトンによる文法の表現 — ATN

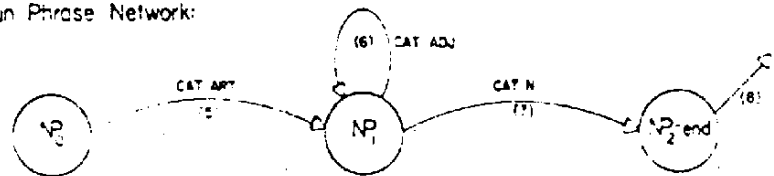
オートマトンによって文法記述を行う代表的なシステムとしては、Woods の ATN (Augmented Transition Network) がある。CF による書換え規則が与えられると、その書換え規則の集合によって定義される文の集合を受理する push - down automaton を構成することが可能である。従って、CF 規則による文法の定義と push - down automaton による定義とは、その表現力の点からは同等の能力を持っている。しかしながら、前述のように automaton を定義する方式では、文法記述者が規則適用の順序までも指定するために、より procedural な記述が可能になっている。

図 5 - 2 に Woods の ATN モデルに従って、簡単な文法を記述した例を示す。図中、円で囲まれた S_0 , S_1 , ..., S_n が状態、状態間をつなぐ枝が文法規則に対応している。枝につけられた SEEK, CAT, WORD はその規則の適用条件を指定するものである。また、図の下に記述されているのが、各規則に付加された ACTION である。オートマトンの遷移図で示される部分が CF 規則で表現される統語的な情報の処理、ACTION 部分が拡張 LINGOL の < advice >, < cog >, < sem > といった統語的情報以外の情報の活用を指定する部分になっている。

Sentence Network:



Noun Phrase Network:



ARC	ACTION
1	ASSIGN SUBJECT to current phrase
2	ASSIGN ACTION to current word
3	ASSIGN OBJECT to current phrase
4	ASSEMBLE CLAUSE SEND current clause
5	ASSIGN DET to current word
6	ASSIGN MOD to current word
7	ASSIGN HEAD to current word
8	ASSEMBLE NOUN PHRASE SEND current phrase

図 5-2 A T Nによる文法の記述

[1] 曖昧さの取扱い

前節の拡張 LINGOL では、文法規則の適用は並列的に進行してゆく。すなわち、入力単語列を左から右へ読んでゆく過程で適用可能な規則はすべて適用されてゆく。その結果生じる複数の解析結果が、< cog > 部分を評価することにより選択される。これに対して、A T N にもとづくシステムにおいては、拡張 LINGOL と同様に単語を左から右へと走査してゆくことによって解析が進められてゆくが、ある単語を走査している時点での文法規則の適用は、必ず 1 つに限られている。ネットワークで示される状態遷移図において、必ず 1 つの状態遷移が選択され、2 つ以上の文法規則が適用可能な場合でも、そのうちのいずれか 1 つが選ばれて、以後の処

理が進行してゆくことになる。もし以後の処理で、それ以前の文法規則適用が誤りであることがわかると、以前の状態へ *backtracking* を起こすことによって、別の規則適用に切替えられる。したがって拡張 *LINGOL* が、曖昧さを *parallel* に保持して進行してゆくのにに対して、*ATN* の方式は解析過程のどの時点をとっても、それまでの解析結果は一意に決まっており、その解析結果が誤りである場合には、*backtracking* によってそれを修正する。この *backtracking* による曖昧さの取扱いは、*ATN* に基づく解析システムに共通の思想で、我々が以前開発した *PLATON* [長尾 1974]、モントリオール大学の *REZO*、グルノーブル大学の *ROBRA* [BOITET 1979] (*ROBRA* は、後述するように完全な意味で *ATN* に基づいているわけではないが、*sab-grammar* 間の順序づけの機構に *ATN* と同様の状態遷移の考え方を入れている) にもみられる。

〔II〕 統語規則以外の情報の活用

ATN の状態遷移を規定する条件は、基本的には *SEEK* (*sub-net* work へ制御を移す)、*CAT* (走査中の単語の品詞をチェックする)、*WORD* (走査中の単語との一致をとる) の3つの操作を使って記述されるが、実際にはこの3つの操作以外に任意の評価可能な *LISP* 表現を使用することができる。したがって、ここで文法的な情報以外に単語の意味的な性質等を参照することによって、意味的な処理も指定することができる。また *ACTION* 部分の活用も可能である。

〔III〕 変換規則・意味解析の実行

ATN では、ルール中の *ACTION* の部分において、処理の中間結果や現在走査中の単語をレジスタンスと呼ばれる変数にセットしておき、これを後のルールにおいて参照することができる。また、いくつかのレジスタに保持された部分構造を組み合わせて、より大きなまとまりの構造を作り出すこともできる (*BUILDQ* 操作)。このレジスタを介した部分構造の

移動をうまく使うことによって、変形規則の多くを表現できることが知られている。

5.2 機械翻訳用プログラミング言語

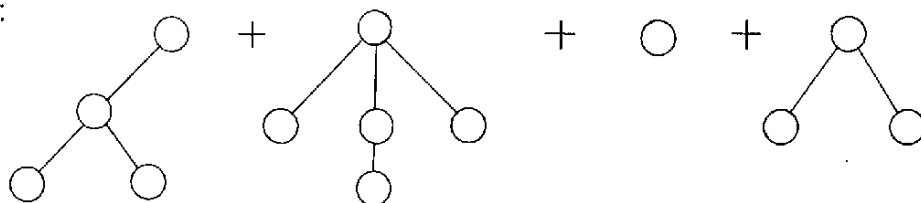
前節で述べた ECFG, ATN がいずれも文の解析に使われる枠組であったのに対して、機械翻訳のプログラミング言語としては、SL 言語から TL 言語への構造変換・TL 中間表現（木構造）から TL 文の生成等を記述する必要があるため、木構造変換の機構を基本とするものが多い。

5.2.1 TAUM: SYSTEM-Q, REZO

モントリオール大学（カナダ）の用いている文法記述言語は、SYSTEM-Q と呼ばれる。Q は Quebec の頭文字である。SYSTEM-Q の特徴はデータ構造の完全性にあり、辞書の記述から文法の記述まで、すべてこの言語だけで表現される。

基本となるデータ構造は、鎖状木構造と呼ばれるもので、順序が意味をもつリストと、構造が意味をもつ木構造の混合形が許される。

例：



すなわち、リスト（鎖）の各要素が木である構造を持ったものである。

SYSTEM-Q で記述された文法は、5.1 の CFG の項で述べたと同様の書換え規則の集合として表現されるが、SYSTEM-Q においては、書換え規則 $\alpha \rightarrow \beta$ の左辺に木構造を許しており、汎用の木構造変換規則になっている。

α として任意の鎖状木構造を許しているために、SYSTEM-Q の文法記述能力は変形文法と同等であり、形式的な能力としてはチューリング機械と同じになっている。

なお、SYSTEM-Q は TAUM の天気予報についての機械翻訳システム TAUM-METEO の開発に全面的に使用されたが、あまりにも汎用的な能力を持ち、効率的な面で問題があるため、TAUM-AVIATION では、統語-意味分析部には、REZO と呼ぶ新しい言語を開発している。REZO は、前述の ATN を基本としており、木-木変換は行わず、String から木構造への変換を基本にしている。

5.2.2 GETA - ROBRA

グルノーブル大学(フランス)の文法記述用プログラム言語も、基本的には木-木変換を基本としている点では、TAUM の SYSTEM-Q と同じである。ただし、グルノーブル大学では、SYSTEM-Q のように辞書記述から文法までをすべて記述するようなプログラム言語は使用していない。すなわち、SYSTEM-Q のように形態素分析から、構文のトランスファ、訳文の生成までを同じプログラミング言語で行なうよりも、それぞれの処理に応じた一連の道具を取りそろえた方が良く、というのがグルノーブル大学の立場である。形態素処理や形態素生成のような簡単な処理をするのに、一般の木-木変換を行うような強力すぎる道具を用いるのは賢明ではない。グルノーブル大学は、この方針に従って次の4つのソフトウェア・ツールを用意している。

- 1) ATEF: String から木への変換システムである。このシステムは、入力文の String (文字列) をうけとり、単位に分割する。ここでの単位とは、単語単位あるいは単語単位よりも細い形態素である。この単位のことを form と呼び、分析の結果は form ごとに木構造に整理され、文全体としては木構造のリストとして、次の段階の処理に渡される。

2) ROBRA: 木構造から木構造への変換システムであり,

a) 入力文の統語的分析, 深層構造の生成

b) SLの中間表現からTLの中間表現へのトランスファ

c) TLの表層構造の生成

の3つの段階で使用され, 機械翻訳システムの中心的役割を果たす。このROBRAについては後に触れる。

3) SYGMOR: 木からStringへの変換システムで, ATEFの逆操作を行う。ROBRAによって生成されたTLの表層構造(木構造)をTLの形態素発生を行いながらStringに変換し, 最終的な出力文を生成する。

4) TRANS: SLの単語をTLの単語に変換する単語変換システムである。

以上のように, 4つのソフトウェアツールの中で, 機械翻訳システムの中心となるのは木構造から木構造への変換を行うROBRAである。以下ではROBRAについて説明する。

[I] 木構造の表現

ROBRAの処理対象は, 木構造である。木構造は, prefix notation によって表現され, 図5-3の木は,

AB(BC(BC(AB(X), AB), BC(AB), CD(B(Y), A(AB),
X, C(X, Y), B(X), A(Y, AB)))

と表現される。

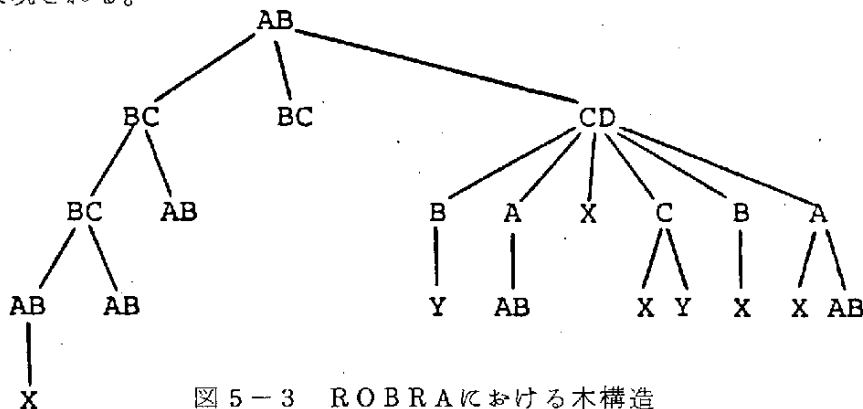


図5-3 ROBRAにおける木構造

[II] Subtree Schema

[1] で与えた prefix notation は、処理対象となる木構造であるが、これに対して適用される規則は、木構造中に変数部分を含んでいたり、節点に条件が付加された一種の木構造パターンを使って記述される。この木構造パターンのことを subtree schema と呼ぶ。subtree schema を用いて木構造間の変換規則を記述した例を次に示す。

(例)

$$\begin{aligned} R1 : (\mathbb{X}3., \text{niv} \leq 3) \quad & 1. AB(?2.AB \& B(\mathbb{X}) \ 3. \&ND \ (A, \mathbb{X}, B(\$L))) / \\ & 2? = BC \ \& \ 3. \in \{CD, XX\} \\ \Rightarrow \ (C, \ \$L, \ & 4.(A), 1.) / \mathbb{X} \leftarrow B., \ 1. \leftarrow A ; C, 4., A \leftarrow \mathbb{X} / \\ & 4. = 3. ; 1. ; = 3. \end{aligned}$$

この例の木構造変換を図 5-4 に示す。

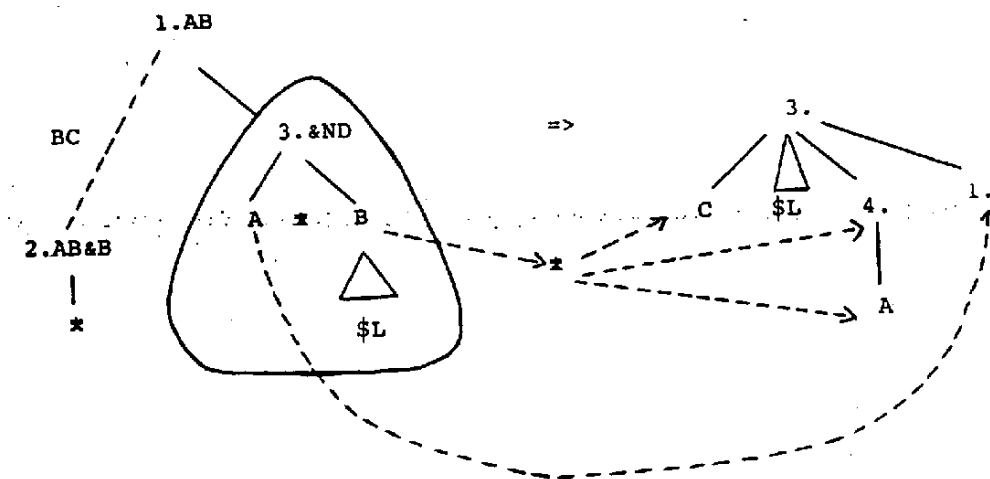


図 5-4 subtree schema 間の交換規則

(III) 部分文法

一般に自然言語の文法規則は、このような subtree schema を用いた木構造間の変換規則の集合で記述されるが、規則の数は 500 を越える膨

大なものとなる。しかも、この規則は、形式言語の場合のように、一度決定すればそれ以後変更されないといったものではなく、翻訳実験を繰返すことによって修正・追加される。このため、文法はできる限り、modularityを保った形式で記述される必要がある。

ROBRAでは、文法のmodularityを保持するために、全体の文法を部分文法(sub-grammar)と呼ぶいくつかの集合に分割し、文法記述者は、この部分文法内でconsistencyを保つようにさえ配慮すればよいようになっている。

[IV] 部分文法間の制御の流れ

部分文法間の関係は、図5-5のようなネットワークによって表現される。図5-5では、部分文法G1を出発点とし、ネットワークに示された枝に沿って次々と処理は移動してゆくことになる。最終的に Λ に到達すれば処理が終了する。部分文法間をつなぐ枝にS1, S2, S3とあるのは、その枝に沿って次の部分文法へと移行するための条件(これも木構造に対するsubtree schemaによる条件チェックにより示される)を示す。 ϵ とあるのは、無条件で移行できることを示す。1つの部分文法から次の部分文法への移行は、あらかじめ定められた優先度に従って行われ、もし以後の処理において不都合が生じた場合には、backtrackingの機構が働くことになる。

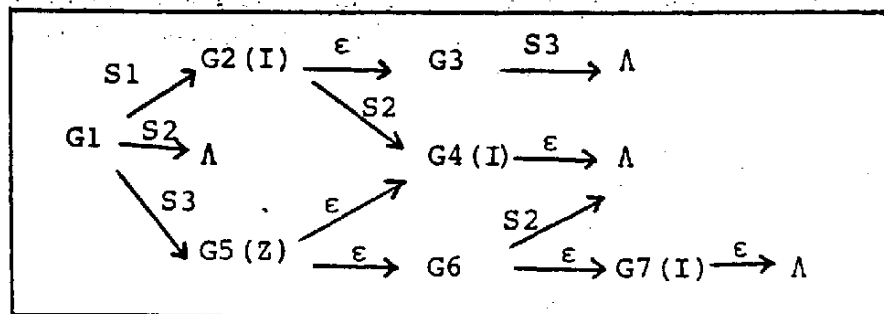


図5-5 部分文法間の制御の流れ

〔V〕 部分文法内での制御

部分文法内での規則の適用は、unitary mode（部分文法中の1つの規則だけが適用される）とexhaustive mode（部分文法内のすべての規則が、適用できる規則がなくなるまで、繰返し適用される）の2つのmodeのいずれかを指定することができる。この他、部分文法内での規則適用を制御するために、7つのパラメータが用意されており、文法記述者が任意に記述できるようになっている。

以上のように、ROBRAはそれまでのグルノーブル大学における機械翻訳研究の経験を総合し、非常に使い易いプログラム言語になっているようである。

5.2.3 ザールブリュッケン大学 — COMSKEE

TAUM, グルノーブル大学がいずれも木構造変換の規則を中心とした、かなり自然言語処理固有の操作をとり込んだプログラム言語を開発しているのに対して、西独のザールブリュッケン大学は、一般のプログラム言語を基本として、これに自然言語処理用のデータ構造をつけ加えたプログラム言語を開発している。

具体的には、PASCALを基礎とし、これに木構造、集合といったデータ・タイプを導入している。ただし、自然言語の持つ曖昧さ、これに伴うbacktracking等の機能は、COMSKEEには何ら用意されておらず、機械翻訳用のプログラム言語というよりも、汎用のプログラム言語の性質が強い。COMSKEEを使った文法記述はまだほとんど手がつけられておらず、今後の課題として残っている。

5.3 おわりに

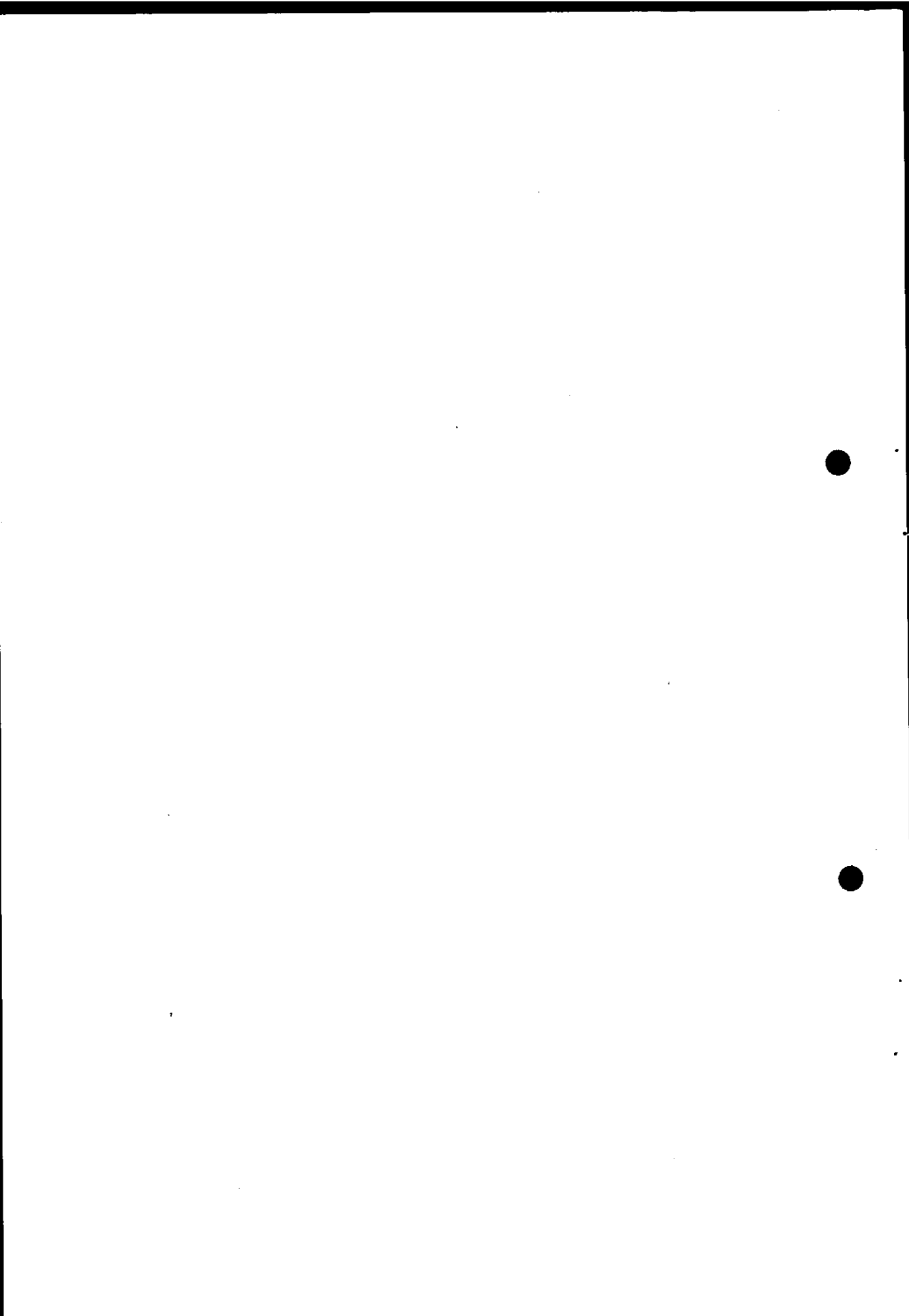
機械翻訳用のプログラミング言語について、いくつか代表的なものを紹介した。しかしながら、いずれの言語もまだ実験レベルの翻訳システムに対し

てしか適用されておらず、本当に詳細な辞書記述を含めて、表現可能なのかどうかは未検討の段階である。今後、実験を繰返すことによって、より操作性の良いものが開発されてゆこう。特に、日本語の解析・発生の文法記述に適したプログラミング言語とはどのようなものかを、我々としても真剣に検討してゆく必要がある。

参考文献

- [BOITET 1979] Boitet, C.: Automatic Production of CF and CS - Analysis using a general Tree Transducer
- [Heidorn 1972] Heidorn, G.E. 'Natural Language Inputs to a Simulation Programming System, 'TR NPS-55 HD72101A, NAVAL Post graduate School
- [Hedrix 1978] Hendrix, G.G. et.al.: Developing a Natural Language Interface to Complex Data, ACM, Trans. on Data Base System, Vol.3, No.2
- [Pratt 1973] Pratt, V.R.: 'A linguistic Oriented Programming Language, 'IJCAI 3, 1973
- [Pratt 1975] Pratt, V.R.: 'LINGOL-A Progress Report, 'IJCAI 4, 1975
- [佐藤 1979] 佐藤, 田中: 日本語の構文解析, 情報処理, vol. 20, No.10
- [白井 1980] 白井, 横尾: 統語処理と意味解釈を同時に行うシステムについて, 情報処理学会, 計算言語研究会
- [田中 1977] 田中, 佐藤, 元吉: 自然言語処理のためのプログラミング・システムー 拡張LINGOLについて, 電子通信学会論文誌 D, J60-D, 12
- [長尾 1974] 長尾, 辻井: 自然言語処理のためのプログラミング言語 PLATON, 情報処理, Vol.15, No.9

6. 辞書の役割とその構造



6. 辞書の役割とその構造

6.1 はじめに

機械翻訳システム (Machine Translation System),あるいは機械援助による翻訳システム (Machine Aided Human Translation System) において, 機械辞書の果たす役割は非常に大きい。3章で述べた「表題文の英和翻訳システム」においては, 専門用語辞書・複数の単語からなるイディオム辞書に, どの程度多くの単語が登録されているかが, システムのパフォーマンスをほとんど決定することになる。この事情は, ECの翻訳局で使われている商用システム SYSTRAN の場合にも全く同じで, 各単語ごとに記述されたイディオム的な表現 (LS—Limited Semantics, CLS—Conditional Limited Semantics) が, 多義語の解消や流暢な翻訳結果を得るために重要な役割を果たしている。また, 4章の「計算機マニュアルの日英翻訳システム」においても, 各動詞ごとの標準的な表層格パターンおよび表層格パターンの変換規則といった単語ごとの辞書記述が単文解析の中心となっている。

以上のような, 機械翻訳システムのための機械辞書を作成するためには, 実際の機械翻訳システムの構成をどうするか, どのような情報が文解析や文生成の時点で参照されるか等を詳細に規定しなければならない。本章では, このような機械翻訳用の辞書とは別に, 我々が日常的に使用している辞書を計算機にデータベースとして蓄積・記憶し, これを活用する試みについて報告する。したがって, 辞書記述として各項目ごとに与えられている内容は, 出版されている辞書の内容そのものである。

このような辞書を計算機で利用できる形式に整備することの意義は, 辞書という貴重な言語資料を様々な視点から検討するという言語研究からの意義のほかに,

(1) 大量の言語情報の管理を行なう際の基本的な手法

(2) 形態素処理を含む柔軟な辞書検索システムの開発（綴りやスペルが完全に一致する項目を検索するだけでなく、語形変化の処理や、日本語の場合には、漢字かな混り表記の処理を同時に行なって、辞書検索を行なう柔軟な手法の開発）

(3) 言語処理アルゴリズムを構成する際の基礎的な言語資料の蓄積

等が考えられる。特に、Machine Aided Human Translation System を構成する場合には、訳語の統一・非専門家のための専門用語辞書の整備（翻訳専門家は、翻訳テキストの主題については非専門家であることが多く、特に専門用語について適切な訳語を与えることが困難であるといわれる）が非常に重要であり、人間による翻訳作業の多くの時間が辞書引きに費されていること考えると、こういった辞書的データを取扱うための Knowhow を蓄積してゆくことは非常に重要である。

この種の目的をもった機械辞書は、すでにヨーロッパ等で積極的につくられており、ドイツでは 1,200,000 語からなる Terminology Data Bank LEXIS, Machine-Aided Translation 用のものとしては、EC の EURODICAUTOM, モントリオール大学の TERMIMUM が有名である。この他、民間会社であるが、Siemens も、かなり膨大な Multi-lingual Dictionary を計算機で管理している。また、テキスト中にあらわれる単語についての、出現順の辞書やアルファベット順辞書を、与えられたテキストごとに作る Text Oriented Dictionary の試みも、この種の Terminology Data Bank の応用として開発されつつある。しかしながら、日本においては、この種の試みはほとんどなく、JICST が規制されたキーワードについてのシソーラスを開発している程度である。

本章では、このような Terminology Data Bank を作成する際の基礎的な問題点を明らかにすることを目的に、我々の研究グループが開発した国語辞典データベースと英和辞典データベースの概要について述べる。我々が計算機

化の対象としたのは、次の2つの辞書である。

- (1) 新明解国語辞典（三省堂出版，電子技術総合研究所によって入力）
- (2) 新コンサイス英和辞典（三省堂出版，文部省特定研究「言語」総括班によって入力）

6.2 国語辞典データベース

6.2.1 日本語辞書の記憶方式

日本語の辞書を引く際には，基本的には漢字で引く場合と読みのひらがなで引く場合の2通りの検索方法がある。すなわち，各辞書項目には検索用キーとして，その語の漢字書き文字列と読みのひらがな列の2つがある。これらのキーは次のような性質を持っている。

- (1) キーによる項目の指定が一意的でない。一般に，漢字で表記されるような語には，同音異語が非常に多くある。また，数は少ないが同字異語（二重ーにじゅう，ふたえ）も存在する。
- (2) 辞書項目を指定するのに，上記の2つのキー以外に，それを変形した複数個の指定方法がある。これは，日本語の漢字かな混り文による正書法が確立していないために，同じ単語の表記が複数個存在するためである。

ここで，(1)の同音異語・同字異音の問題は，情報検索システムで使われる転置ファイルを各キーごとに用意することによって解決できる。しかしながら，(2)の表記のゆらぎについては，すべての変形されたキーに対して転置ファイルを作るとは，記憶量の増大につながり，得策ではない。しかも，ほとんどの単語の検索は漢字列かひらがな列で行なわれ，変形されたキーで検索されることは，（通常の日本語文を対象としているときには）非常に少ないと考えられる。したがって，その場合の検索経路を転置ファイルのような固定的なデータ構造で表現しておくことは得策ではないと考えられる。そこ

で我々は、漢字列・ひらがな列による検索にはできるだけ高速に回答でき、しかも変形されたキーによる検索にも多少時間はかかるが、すばやく回答できるような記憶構造として、図6-1に示す構造をとっている。

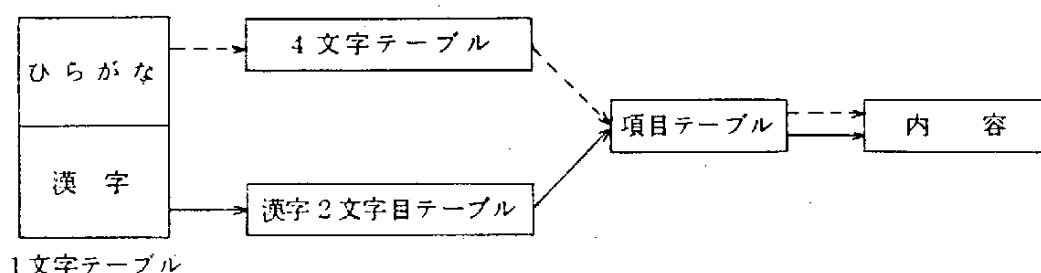


図6-1 国語辞典のためのデータ構造

図6-1の1文字テーブル・5文字テーブル・漢字2文字目テーブルはそれぞれ項目テーブルを検索するためのインデックス用のテーブルである。太線は漢字から、点線は読みから検索経路を示す。ここで、項目テーブルは読みのひらがな列によってソートされており、また漢字の2文字目テーブルは漢字の1文字目をキーにした転置ファイルになっており、2文字目についてソートされている。前者が国語辞典的な配列を、後者が漢和辞典的な配列を表現している。

各テーブルの構成を次に簡単に示す。

- 〔1文字テーブル〕 漢字コードに対応して4096のエントリからできているテーブル。ひらがなのコードに対応するエントリには、そのひらがなから始まる4文字テーブルの最初のエントリへのポインタ、漢字のコードに対応するエントリには、漢字2文字目テーブル中のその漢字に対するエントリへのポインタがそれぞれ入っており、メインコアに常駐している。
- 〔4文字テーブル〕 項目テーブルの45項目ごとに先頭の項目の見出しから第2文字～第5文字の4文字をとってきて、その4文字の項目テーブルへのポインタを設けたもの。ひらがな表記からのアクセスの場合には、そ

の語の表記の先頭の1文字をみて、1文字テーブルからこの4文字テーブルの対応するエントリをひき出し、対応する項目テーブルのブロックを読み出してくることになる。

〔漢字2文字目テーブル〕 漢字表記から辞書項目へのアクセスをするためのテーブル。2文字目にくる漢字コードと、対応する辞書項目へのポイントが対になっている。入力として漢字表記が与えられると、その先頭の漢字に対応する1文字テーブルのエントリからこのテーブルが参照され、2文字目の漢字との対応をとることによって、項目テーブルのエントリがひかれる。

〔項目テーブル〕 見出し語についての品詞、ひらがな表記、漢字表記、活用型、アクセント・パターン、重要語と非重要語の区別、意味記述部分へのポイント等の情報を持つ。ひらがな表記のコードによってソーティングされている。

図6-2に1文字テーブル・4文字テーブル・漢字2文字目テーブル・項目テーブルの関係を示す。表6-1に各テーブルに必要となるサイズを示す。

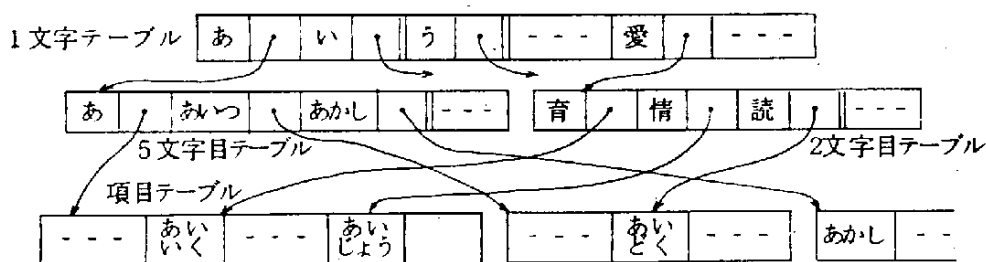


図6-2 国語辞典データベースのインデックス構造

表 6-1 インデックスに必要な記憶容量

テーブル名	サイズ
ひらがな用 1 文字テーブル	340 バイト
4 文字テーブル	12,520 バイト
漢字用 1 文字テーブル	26 K バイト
2 文字目テーブル	1.3 M バイト
漢字音訓用テーブル	130 K バイト

6.2.2 辞書の検索方式

基本的な辞書検索機能として、前節で示したデータ構造に基づいて、次のようなものを設けている。

(1) ひらがな書きされた見出しをキーとする検索

(2) 漢字で書かれた見出しをキーとする検索

(3) (1)で同音異義語を次々に出力する検索

(4) (2)で同音異義語を次々に出力する検索

(3)・(4)はそれぞれ項目テーブルが「ひらがな書き」でソートされ、漢字 2 文字目が先頭の漢字でまとめられていることを使っている。

さらに、漢字かな混り表記の処理を行なうためには、変形されたキーによる検索を行なう必要があり、このために下記に示す検索機能が設けてある。

(5) 省略可能な送りなを省略した見出しをキーとする検索

(6) 漢字表記の単語の一部をひらがな書きした見出しをキーとする検索

(7) 活用語の活用形での検索

(1)・(2)・(3)・(4)の検索機能がいずれも前述した 4 文字テーブルや漢字 2 文字目テーブル等の内部構造を直接参照するだけで実現できたのに対して、この(5)・(6)・(7)の各機能は、4 文字テーブルや漢字 2 文字目テーブルを手掛り

にして、項目テーブル中の特定の個所へ到達した後は、項目テーブル中の記述をプログラムでチェックしながら検索する。

特に、(6)の混合表記による辞書項目のアクセスは、日本語の表記体系が固定しておらず、しばしばひらがなと漢字との混合があらわれる以上、実際に日本語の機械処理を考える場合には、絶対に解決しておかなければならない問題である。我々は当初、1文字目がひらがなであるか漢字表記であるかに応じて、1文字目テーブルのエントリを読み出し、後は4文字テーブル・漢字2文字目テーブルに従って網羅的に項目テーブルを検索する方式をとっていたが、これでは時間がかかりすぎるので、各漢字の音訓表を活用する方式に変更した。この方式では、漢字ひらがな混り表記が与えられた場合に、まず漢字部分に関して可能なひらがな読みをすべて与え、通常のひらがな表記に対するアクセスを行なうことになる。現在、当用漢字に関するすべての音読み・訓読みが用意されており、このためのテーブルサイズは、130 Kbyte になっている。

6.2.3 国語辞典の応用

国語辞典データベースの応用として我々は図6-3に示すような逆引き辞書、および科学技術情報センタから発行されている文献速報磁気テープにあらわれた長い漢字連続の分割プログラムを作成した。このプログラムは表6-2に示す各漢字の統計的振舞いを利用している。分割例を図6-4に示す。また、機械辞書としてこの国語辞典を使用した例を図6-5に示す。

*****HI BASHI*****SEI SHO*****HINSI*****		
いりつける	通(り)付ける	他 下
うりつける	売(り)付ける	他 下
えりつける	彫(り)付ける	他 下
きりつける	切(り)付ける	他 下
すりつける	擦(り)付ける	他 下
こすりつける	擦(り)付ける	他 下
なすりつける	擦り付ける	他 下
てりつける	照(り)付ける	他 下
とりつける	取(り)付ける	自 下
どなりつける		他 下
ぬりつける	塗(り)付ける	他 下
のりつける	乗(り)付ける	他 下
はりつける	張(り)付ける	自 下
もりつける	盛(り)付ける	他 下
かたづける	片付ける	他 下
とりかたづける	取(り)片付ける	他 下
たちつづける	立ち(ち)統ける	他 下
あとづける	跡付ける	自 下
ことづける	言付ける	他 下
なづける	名付ける	他 下
ふんづける	踏ん付ける	他 下
こける	出来る	他 下
とける	溶ける	自 下
とける	解ける	自 下
とける	退ける	自 下
おどける		他 下
ととける	届ける	自 下
ききとどける	聞(き)届ける	他 下
みととける	見届ける	他 下
なける	泣ける	自 下
ぬける	抜ける	自 下
あかぬける	赤抜ける	自 下
つきぬける	突(き)抜ける	自 下
かきぬける	掻(き)抜ける	自 下
すぬける	す抜ける	自 下
ずばぬける	ずば抜ける	自 下
とびぬける	飛(び)抜ける	自 下
すつはぬける	すつは抜ける	自 下
きりぬける	切(り)抜ける	他 下
すりぬける	擦(り)抜ける	自 下
ぬける	退ける	他 下
ぬける	除ける	他 下
おのける	仰(の)ける	他 下
かきのける	二(き)退ける	他 下
つきのける	突(き)除ける	他 下

*****RECHU= 3183 TO 3227*****

図 6 - 3 逆引き辞書の出力例

表 6 - 2 接頭字・接尾字テーブル

(1) 接頭字テーブル

各 = 0 9 5 / 0 1 3 6	他 = 0 9 0 / 0 0 3 1	勝 = 0 8 5 / 0 0 0 7
月 = 0 8 0 / 0 0 0 5	双 = 0 8 0 / 0 0 0 5	石 = 0 8 0 / 0 0 0 5
通 = 0 7 8 / 0 0 3 3	新 = 0 7 6 / 0 0 4 3	旧 = 0 7 5 / 0 0 0 4
弱 = 0 7 2 / 0 0 1 1	商 = 0 7 2 / 0 0 5 1	主 = 0 7 0 / 0 0 5 5
積 = 0 6 9 / 0 0 1 3	恒 = 0 6 6 / 0 0 0 3	親 = 0 6 6 / 0 0 0 3
礙 = 0 6 6 / 0 0 1 2	全 = 0 6 6 / 0 1 0 9	魂 = 0 6 6 / 0 0 0 3
低 = 0 6 5 / 0 0 7 8	無 = 0 6 4 / 0 0 5 0	再 = 0 6 3 / 0 0 4 9
枝 = 0 6 2 / 0 0 0 8	考 = 0 6 2 / 0 0 2 4	愚 = 0 6 0 / 0 0 0 5
熟 = 0 6 0 / 0 1 0 7	本 = 0 6 0 / 0 0 7 8	孩 = 0 5 8 / 0 0 1 7
何 = 0 5 7 / 0 0 0 7	未 = 0 5 7 / 0 0 0 7	四 = 0 5 4 / 0 0 1 1
総 = 0 5 3 / 0 0 1 5	断 = 0 5 3 / 0 0 6 4	宣 = 0 5 3 / 0 0 1 3
萬 = 0 5 1 / 0 0 9 6	引 = 0 5 0 / 0 0 0 8	團 = 0 5 0 / 0 0 0 2
丸 = 0 5 0 / 0 0 0 2	湯 = 0 5 0 / 0 0 0 2	銀 = 0 5 0 / 0 0 1 0
句 = 0 5 0 / 0 0 0 2	隅 = 0 5 0 / 0 0 0 2	小 = 0 5 0 / 0 0 6 1
床 = 0 5 0 / 0 0 0 6	遇 = 0 5 0 / 0 0 0 2	育 = 0 5 0 / 0 0 0 2
特 = 0 5 0 / 0 0 0 6	田 = 0 5 0 / 0 0 0 2	鶴 = 0 5 0 / 0 0 0 4
脇 = 0 5 0 / 0 0 0 6	助 = 0 5 0 / 0 0 0 2	同 = 0 4 5 / 0 0 6 0
未 = 0 4 5 / 0 0 1 1	多 = 0 4 3 / 0 0 4 6	巻 = 0 4 2 / 0 0 0 7
半 = 0 4 2 / 0 0 2 8	的 = 0 4 1 / 0 0 1 7	鋼 = 0 4 1 / 0 0 1 7
二 = 0 4 1 / 0 0 5 8	英 = 0 4 0 / 0 0 0 5	底 = 0 4 0 / 0 0 0 5
今 = 0 4 0 / 0 0 0 5	幽 = 0 4 0 / 0 0 0 5	信 = 0 4 0 / 0 0 1 0
暗 = 0 3 8 / 0 0 1 3	筋 = 0 3 8 / 0 0 1 3	静 = 0 3 8 / 0 0 2 1
備 = 0 3 8 / 0 0 1 8	油 = 0 3 8 / 0 0 1 3	軟 = 0 3 7 / 0 0 0 8
耐 = 0 3 6 / 0 0 1 9	頭 = 0 3 6 / 0 0 1 1	際 = 0 3 5 / 0 0 1 4
復 = 0 3 5 / 0 0 2 0	光 = 0 3 4 / 0 1 8 0	閉 = 0 3 4 / 0 0 2 6

(2) 接尾字テーブル

者 = 0 8 9 / 0 0 9 3	例 = 0 8 9 / 0 0 9 6	梯 = 0 8 8 / 0 0 1 8
承 = 0 8 7 / 0 2 1 0	型 = 0 8 7 / 0 0 8 9	利 = 0 8 5 / 0 0 2 1
割 = 0 8 5 / 0 0 6 1	法 = 0 8 4 / 0 2 7 8	所 = 0 8 2 / 0 0 2 8
器 = 0 8 1 / 0 1 8 1	鏡 = 0 8 1 / 0 0 2 2	等 = 0 8 1 / 0 1 3 3
盤 = 0 8 1 / 0 0 1 6	片 = 0 8 1 / 0 0 1 6	貝 = 0 8 0 / 0 0 1 0
厚 = 0 8 0 / 0 0 0 5	室 = 0 8 0 / 0 0 4 2	台 = 0 8 0 / 0 0 2 0
窓 = 0 7 8 / 0 0 1 4	値 = 0 7 8 / 0 1 8 5	塔 = 0 7 7 / 0 0 0 9
鋼 = 0 7 7 / 0 0 6 2	群 = 0 7 6 / 0 0 4 2	園 = 0 7 5 / 0 0 0 4
接 = 0 7 5 / 0 0 8 3	店 = 0 7 5 / 0 0 0 4	粉 = 0 7 5 / 0 0 0 4
弁 = 0 7 5 / 0 0 2 4	上 = 0 7 4 / 0 2 0 8	閉 = 0 7 4 / 0 0 7 5
部 = 0 7 4 / 0 2 2 4	則 = 0 7 3 / 0 0 3 4	層 = 0 7 3 / 0 0 9 1
率 = 0 7 3 / 0 1 3 3	局 = 0 7 2 / 0 0 5 1	項 = 0 7 2 / 0 0 3 7
性 = 0 7 2 / 0 4 2 1	板 = 0 7 2 / 0 0 6 6	案 = 0 7 1 / 0 0 1 4
所 = 0 7 1 / 0 0 4 2	化 = 0 7 0 / 0 2 2 1	輪 = 0 6 9 / 0 0 4 2
省 = 0 6 9 / 0 0 1 3	親 = 0 6 9 / 0 0 3 9	式 = 0 6 8 / 0 1 6 6
甘 = 0 6 6 / 0 0 0 3	顔 = 0 6 6 / 0 0 0 6	広 = 0 6 6 / 0 0 0 6
個 = 0 6 6 / 0 0 0 9	湖 = 0 6 6 / 0 0 0 3	財 = 0 6 6 / 0 0 0 3
士 = 0 6 6 / 0 0 0 3	版 = 0 6 6 / 0 0 0 3	業 = 0 6 6 / 0 0 0 3
騎 = 0 6 6 / 0 0 0 3	比 = 0 6 5 / 0 0 8 2	城 = 0 6 4 / 0 0 9 8
車 = 0 6 4 / 0 0 2 8	紙 = 0 6 4 / 0 0 1 4	角 = 0 6 3 / 0 0 6 0
内 = 0 6 3 / 0 1 1 9	幣 = 0 6 2 / 0 0 4 5	物 = 0 6 2 / 0 0 9 3
量 = 0 6 2 / 0 2 1 7	源 = 0 6 1 / 0 0 8 1	中 = 0 6 1 / 0 2 4 5
点 = 0 6 1 / 0 1 8 6	費 = 0 6 1 / 0 0 4 9	側 = 0 6 1 / 0 2 7 0
雲 = 0 6 0 / 0 0 0 5	街 = 0 6 0 / 0 0 0 5	孔 = 0 6 0 / 0 0 1 5
川 = 0 6 0 / 0 0 0 5	卓 = 0 6 0 / 0 0 0 5	時 = 0 5 9 / 0 1 9 9
空 = 0 5 9 / 0 0 2 7	家 = 0 5 8 / 0 0 1 7	前 = 0 5 8 / 0 0 3 6

成功例

高 非 擬似 代數 磁氣 巨視 過 信號 數學 非 不 多 誘電 光學	周波 直線 逆 方程 異方 的 負荷 對 的 線系 均一 結晶 的 的	線路 動作 行列 式 性 設計 保護 漏話 統計 時變 媒体 共振 非 均質	法 比 法 系 內 器 對稱 性	性
--	--	---	---------------------------------------	---

失敗例

相 複 電磁 非 複	對論 々線 界方 常用 素伝	的 化 程 通信 搬	効果 工事 式 網 定數
------------------------	----------------------------	------------------------	--------------------------

図 6 - 4 漢字連続の分割例

KANJI= 言語

MOJISU= 2:
14400 14444 6958 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0

KOMOKUSU= 2:
けん ご 言語 名

①「言語」 【「けん」は漢音、「ご」は呉音】人間の表現活動の一つで、音声（や文字）によって、思想・意思や感情を表現したり伝達したりする行為。また、それに用いられる音声（や文字）。言葉。「一に絶する【＝言う言葉が無い】」①・障害部一シャウ①言葉を正しく言ったり、理解したり出来なくなる障害。どもり、ふめいりょうな発音、失語症など。

GET NEXT? YES(1),NO(2)

1
1
ごん ご 言語 名

②「言語」 「言葉に出して表現する」意の仏教語。「一に絶する」①・道徳どうだん①一ダウ① 【なんとも表現しがたい意】語にならないほど、正道からはずれていること。

例1 漢字表記からの辞書引き — 「言語」

KAN+MOJI= あいじよう

MOJISU= 5:
9250 9252 9272 9319 9254 6958 0 0 0 0
0 0 0 0 0 0 0 0 0 0

KOMOKUSU= 2:
あい じよう 愛嬌 名

部一シャウ①愛嬌① かawaii（と思っている）顔。まなむすめ。

GET NEXT? YES(1),NO(2)

1
1
あい じよう 愛情 名

部一シャウ①愛情① こ自分の身近に有る人や、自分より弱い立場に有る人に対して、何とか（尽く）してあげたいと思う心。「皆（へ）の」①異性に対して、特にその人でなければと思ひ、親しむ心。「一を打ち解ける」

例2 ひらがな表記からの辞書引き — 「あいじよう」

KANJI= あい色

MOJISU= 5:
9250 9252 16167 6958 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0

==OKUNTEL USED==

色 しき；じき；しよく；いろ；

あいしき

NOT FOUND IN KOMOKUTBL

あいじき

NOT FOUND IN KOMOKUTBL

あいしよく

NOT FOUND IN KOMOKUTBL

あいいろ

KOMOKUSU= 1:

HATCHING_KOMOKUSU= 1:
あい いろ 藍色 名

部一シャウ①藍色① 濃く深い青色。

例3 混合表記からの辞書引き — 「あい色」

図6-5 辞書引きの例

6.3 英和辞典データベース

6.3.1 英和辞典をデータベース化する場合の問題点

国語辞典では、見出しのひらがな表記・漢字表記以外はあまり項目分けがされず、言葉で説明されているのに対して、英和辞典においては、見出し語のスペル・品詞・発音・語義・用例・イディオム等が、系統だった木構造に整理されている（図6-6）。したがって、データベース化するには、これら項目間の関係が失われないように注意する必要がある。

このような記述に構造のあるデータを入力する場合には、予め人手によって整理した後に計算機入力することも考えられるが、このためには膨大な人々と時間を要することになる。そこで、我々は印刷・出版された辞書のイメージをそのまま（ゴシックやイタリックの字種情報も含めて）入力し、計算機プログラムによってこれを前述の木構造に整理することにした（図6-7に入力された辞書データを示す）。この変換を直接通常のプログラム言語で実行することは、各種例外記述の処理を含む試行錯誤的なプログラムを書くことになり、ほとんど不可能である。そこで、まずこのような印刷イメージのデータの持つ構造を規則の形式で表現すると、その規則に従ってデータの構造化を行なうソフトウェアツールを開発、これを用いることにした。このシステムは、規則に従ったデータの構造化と同時に、入力データに含まれる入力エラーの検出も行なうので、以下このシステムのことをData Verifier / Compilerと呼ぶことにする。

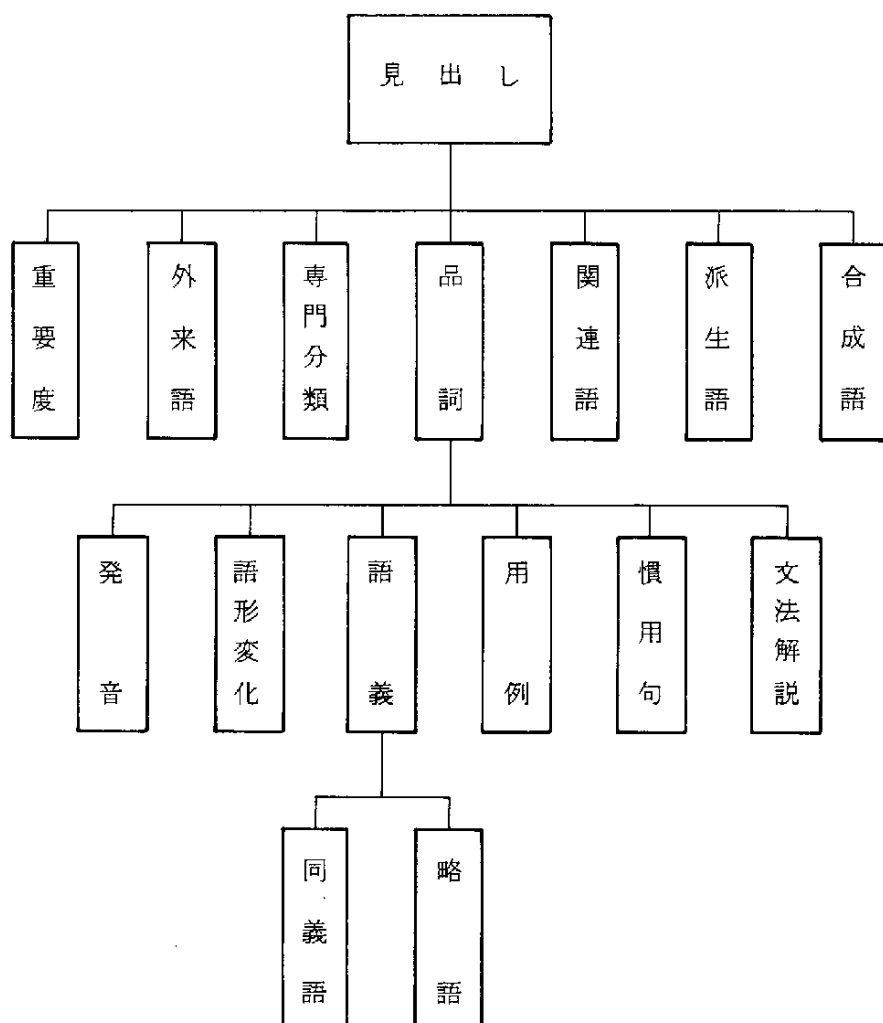


図 6-6 新コンサイス英和辞典の記述項目

SKIP NO. & OUTPUT NO. =

- 204 -

6.3.2 Data Verifier / Compiler の構成

Data Verifier / Compiler は、入力された文字を 1 文字 1 文字走査し、次にとるべき行動を決定してゆくことを基本としており、有限状態をオートマトンになっている。システムの利用者は、まずオートマトンで入力データの持つ構造を定義する。オートマトンをデータ・シンタックスの記述に用いる利点は次のようである。

1. 入力データの構造をルールの形式で規定でき、わかり易い。
2. 直接データ変換用のプログラムを書くのと比較して、後の保守性・拡張性が良い（特に、辞書のように例外的な記述の多いデータを処理するためには、データを処理してゆく過程で、新たにルールを追加・変更する必要が生じることが多いので、この性質は不可決である）。
3. データのシンタックスを記述するもの 1 つの手法として、書き換え規則が考えられる。しかしながら、エラーの検出・データ変換等を柔軟に行なうためには、ルール記述の段階で、システムの動きを利用者が把握している必要があり、この点で書き換え規則は不十分であり、より procedural なルール記述のできるオートマトン・モデルの方が適していると考えられる。

6.3.3 Data Verifier / Compiler の仕様と構成

システムは、先に述べたとおりオートマトンを基本としており、入力ファイルの文字を 1 文字（2 バイト）ずつ走査し、適用コードに一致した場合、指定された状態に遷移し、入力ファイルのポインタを指定に従って進める。このとき同時にルール中に指定された動作（アクション）を実行する。システムの目的がデータを定型化することにあるので、このルール中に記述されたアクションでは、バッファやファイルへのデータの書き出しを行なうことになる。ここでは、特に、バッファ、ファイル、アクション、適用コー

ードの概念について説明する。

i) ファイルおよびバッファ

ファイルは、定型化データを出力するためのものである。ファイルに対応して、出力するデータをオートマトン中で蓄積してゆく。バッファという概念がある。バッファのサイズは自由に指定することができ、ルール中のアクションによって、いくつかのバッファを併合して別の大きなバッファに移したり、バッファの一部をとり出して他のバッファに書き移すこと等が指定できる。

また、利用者が指定するバッファ・サイズ以上にデータの書き込みが行なわれた場合には、現在のステートの如何にかかわらず、各バッファごとに指定されたオーバーフロー・ステートに自動的に遷移する。この機能は、データが可変長になる場合、短い固定長のデータをいくつかつなぎあわせて表現したり、またある一定長以上にはならないことが予め判っている場合のエラーチェックを行なう場合に使われる。

ii) アクション

定型化したデータを、入力データから抽出するために、現在走査している入力文字を特定のバッファに転送したり、また前記のバッファの併合や、バッファからファイルへのデータの書き出しを指定するのが、アクションである。現在用意されているアクションには以下のものがある。

- (1) 入力文字のバッファへの転送
- (2) バッファ内容のファイルへの書き出し
- (3) 複数のバッファの併合
- (4) バッファの初期化
- (5) カウンタ操作

iii) コードの定義

一般に、アルファベット以外の文字を処理しようとするとき、標準化されていない文字や制御コードをプログラム中で参照する必要がある。特に、

英和辞典中には字体シフトを含む様々な制御コード（タグを考えても良い）や数種のカッコ記号等が使われており、データ定型化の手がかりになっている。我々の **Data Verifier / Compiler** では、ルール集合の定義を行なう前に、まずそのルール集合中で使われるコードを定義する。コード定義は次のように行なう。

SMALL-ALPH = 9057-9082

HATSUON = 9057-9082, 9093-9095, ……

BOLD = 10273

コード定義は、2 バイトのコードを10進数としてみたものを使う（本システムは、基本的には2バイト・コードの体系で記述されたデータを対象としている。将来、1バイト・コードと2バイト・コードの混在したデータを取扱う場合には、文字の走査方法について拡張が必要となる）。上記例では、アルファベット小文字という文字集合を10進数で9057から9082までのコード、発音記号として9057から9082までと、9093から9095までのコード、ボールド体シフト記号として、10273のコードを指定している。ルールの定義は、このコード定義を使って行なわれる。

iv) ルールの定義

ルールには、そのルールが適用できるかどうかコードによって指定されている。例えば、MIDASH1 というステートに付けられた7個のルールを図6-8に示す。

第1番目のルールは、ノーマル・ルールと呼ばれ、入力文字がコードHYOに合致した時に適用される。その結果、コントロールはステートSTATISへ移る。以下6個のルールは、ストリング・ルールと呼ばれるものであり、指定されたコードが入力データ中でその順で並んでいる場合にのみ適用される。各ルールの下に、先頭が+記号で続いているのが、そのルールに指定されたアクションである。

Data Verifier / Compiler に与えるルールのシンタックスを付録に、またシステムの構成図を図 6-9 に示す。図 6-9 に示すように、このシステムはコンパイル部と実行部とからなっている。コンパイル部は、利用者の与えたコード定義・ルール定義・バッファ定義等を読み込み、各コード・ステート・バッファを数字で表現する内部表現形に変換する。実行部は、この内部表現形を参照して、実際に入力ファイルから 1 文字ずつ文字を読み込み、データのチェックと定型化の作業を行なってゆくことになる。

```

00340 MIDASI (HYD)      STATIS
00341      <HIFN,CR,BOLD>      MIDASI
00342      <CR,LKATD> STATIS 0
00350      <HYD,BRANK,ALFBI> REIG01 0
00360          + ADD -1 CTR(MIDCTR)
00370      <HYD,LKACE,EI,RKACE,BOLD> MIDASI
00380          + WRITE MOJI(EI) BUF(MIDASI)
00390      <HYD,LKACE,BEI,RKACE,BOLD> MIDASI
00400          + WRITE MOJI(BEIST) BUF(MIDASI)
00410      <HYD,LKACO,BOLD> MIDASI
00420          + WRITE MOJI(LKACO) BUF(MIDASI)
00430      <HYD,RKACO,BOLD> MIDASI
00440          + WRITE MOJI(RKACO) BUF(MIDASI)
00450      <HYD,RKACO> MIDASI
00460          + WRITE MOJI(RKACO) BUF(MIDASI)
00461      <GOU,UP1> MIDASI
00462          + WRITE *-1 BUF(MIDASI)
00463          + WRITE * BUF(MIDASI)
00464      <GOU,UP2> MIDASI
00465          + WRITE *-1 BUF(MIDASI)
00466          + WRITE * BUF(MIDASI)
00470      (ITALIK) MIDASI
00480          + WRITE MOJI(MK) BUF(MARK1)
00490      (ALFBIS,ALFBTL,PRIOB,HIFN,FF) MIDWRT 0
00491      (SPCOD1) MIDWRT 0
00500      (COMMA,BRANK,LKACO,AND,APOS,GOSEI,RKACO,SLASH) MIDWRT 0
00501      <CRUBON,SUJI> MIDASI
00502          + WRITE *-1 BUF(MIDASI)
00503          + WRITE * BUF(MIDASI)
00510      (*)      ERR31 0

```

図 6-8 ルール記述の例

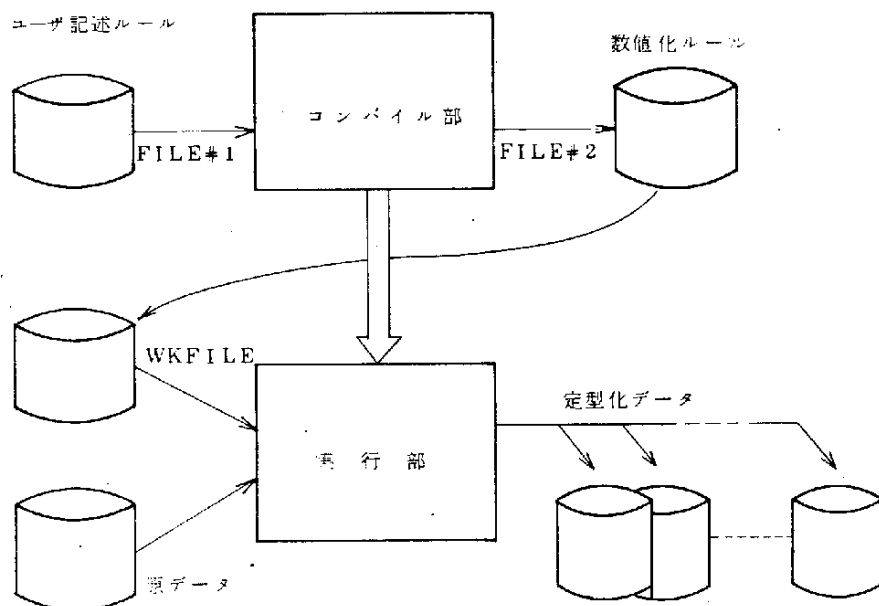


図 6-9 Data Verifier/Compiler の構成

6.3.4 実際の使用例 — 英和辞典の場合

実際にこのシステムを用いて英和辞典をデータ・ベース化するルールを作成した。入力となった原データは前述の図 6-7 に示した。また、定型化後のデータのフォーマットを付録に示した。

初めに、データの認識は字体変更記号等で簡単に行なえるかのような書き方をしたが、字体とデータ項目属性は一対一対応をしている訳ではないから、実際はそれほど容易なことではない。このため、ルール記述は 1300 行（ルール数は約 500）を越え、遷移ルールだけでも 1000 行を要する複雑なものとなっている。

これだけ複雑なルールを用いても、まだ構造化は完全になっていない。それは原データ作成時におけるタイプ・ミス、および、より本質的には用例・慣用句等の記述等において、出版された辞書自体がかなり不統一な記述形式をとっているためである。現在、これを修正するためのエディタを作成中で

ある。このエディタには、修正したデータを即座にオートマトンにかけ、エラーチェックを行なわせる機能があり、英和辞典のデータベース化以外にも、データベース構成のための、汎用エディタ（データ、エントリシステム）とに利用される予定である。

現在は新コンサイス英和辞典中のイニシャルが 'F' のデータだけを定型化して、京大大型計算機センタのディスク内に格納している。表 6-3 にはそれぞれの記憶容量を示している。これらの合計は 3 メガバイトを超えており、'F' からはじまる単語は英和辞典全体のほぼ 1/22 を占めていること、入力エラーにより定型化されなかった項目の量を考えると、この辞書を全部記憶すると 70 メガバイトを越すと考えられる。

しかしながら、現在の記憶形式においては、取扱いの容易さのために、すべてのデータ項目を定長のレコードで表現していること、またエンコーディング等のデータ圧縮手法を全く使っていないことなどから、この容量自体は約半分程度（40 メガ弱）に減少させることが可能である。

表 6-3 定型化されたデータの容量（'F' の項）

見出し部	689 (Kbytes)
関連語部	13
派生語部	52
合成語	221
発音	247
語形変化	13
語義	1287
用例	195
慣用句	286
文法解説	13
同義・対義	13
略語	13

6.3.5 英和辞典データベースの持つべき機能とデータ構造

国語辞典データベースにおいては、日本語固有の表記体系から生じる問題を主として考えてきた。ここでは、英和辞典データベースを例題として、辞書的なデータを種々の観点から柔軟に活用しようとした時に生じる問題について考える。

英和辞典データベースでは、辞書の通常の利用方法である見出し語のスペルによる検索だけでなく、慣用句・用例による検索、発音（例えば、ある特定の発音列を含む単語）、語形変化をうけた変化形による検索など、さまざまな可能性が考えられる。ここでは、この中でもとくに重要な単語による検索を基本とした機能について考える。現在我々の作成した英和辞典データベースに用意されている検索機能は以下のようなものである。

i) 見出し語のスペルによる検索

ii) 語の変化形による検索

探そうとする語が語形変化している場合、もしそれが規則変化に従っているものならば、一般には辞書の見出しとしては登録されていないので、英語の語形変化を取扱うプログラムによって原形に変換し、その後(i)の場合と同じように検索を行なう。不規則変化ならば辞書に登録されていても、簡単に「～の過去形」という程度の説明しか与えられていない。より詳しい情報を知りたければ原形で書き直す必要があるため、現在の英和辞典データベースでは不規則変化した語から直接原形の語につけられた情報が得られる構造になっている。もちろん変化形が見出し語として特別な意味を持っている場合には、それも同時に得られる。

iii) 慣用句・用例での検索

人間が慣用句または用例の意味を調べる場合、それを構成する単語の1つをキーワードとして辞書引きし、その慣用句・用例の中で一致するもの、または良く似ているものを探す。もしなければ、入力文（句）中の別の単語をキーワードに選び、再び辞書引きをし、見つかるまで繰り返す。この

ように用例・慣用句による辞書引きは基本的には(i)の単語スペルによる検索を繰り返すことによって実現できる。データベース中においても、単語ごとの用例・慣用句のアクセスを繰り返すことによって必要とする慣用句・用例の意味を検索することができる。

6.3.6 英和辞典データベースの作成

1) ダイレクトアクセス・データセットへの変換

英和辞典データベースは、現在PL/1を使って実現されており、各データ項目はダイレクトアクセスファイルREGIONAL(1)の形で蓄積されている。PL/1のREGIONAL(1)では固定長非ブロック化データセットしか許されていないが、記憶領域の効率を上げ、ファイルアクセスの回数を減らすため、数レコードを1ブロックにして書き込んでいる。

2) 見出しテーブルの作成

単語のスペルによって辞書を引くという単純な操作を行なうためにも、出版された辞書イメージのままでは実現できない。出版された辞書の見出し部分には次のような問題となる記述がある。

- i) ' . ' で示される音節区切りが各単語ごとに記述されている。
- ii) 同字異義語の場合、添数字で区別がつけられている。
- iii) 2つ以上のスペルを持つ単語があり、それらは' , ' (コンマ)により区別されている。
- iv) スペルの書き方に簡略化がある。

(例) fa•vor, 《英》— vour …favor, favour
fa•vo(u)r•able …favorable, favourable

v) 複数のスペルを持つ単語には英語と米語の区別がつけてあるものがある。

これらの記述があるために、この見出し部と入力スペルのマッチングを検索時に行なうことは、処理時間の点から不可能である。また、これ以外に、不規則変化形については、原形についての辞書記述中に埋もれて記述

されていることが多いため、これらも通常の見出し語を同様に取り出しておく必要がある。辞書中の見出し部を処理し、複数綴り・不規則変化等を取り出した形にしたものを見出しテーブルと呼ぶ。見出しテーブルと見出し部の関係を図6-10に示す。見出しテーブルの1項目は分解・整理されたスペルと、見出し部へのポインタからなる。見出し部からは、その単語に関するすべての情報が検索できるようになっている。見出しテーブルは、アルファベット順にソートされている。

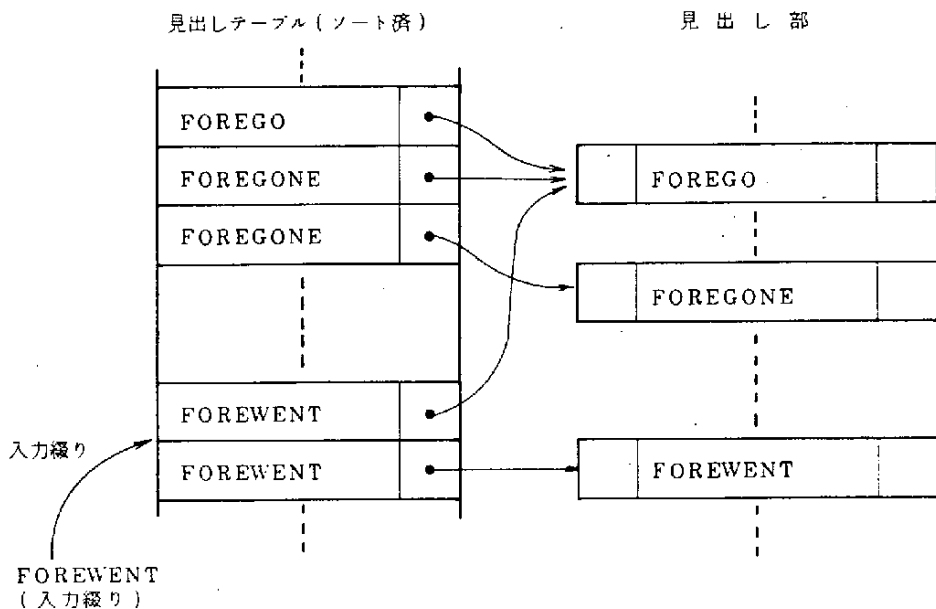


図6-10 見出しテーブルと見出し部

3) インデックステーブル

見出しテーブルの項目数は、約70,000であるため、これを主記憶上に常駐させておくことは実際問題として不可能である。そこで、見出しテーブルのインデックスを主記憶上に作り、ファイルへのアクセスは1回ですませることにする。

インデックステーブルは、見出しテーブルを50項目ごとに区切り、その先頭の項目番号と最後尾のスペルを入れたものである。インデックステーブルと比較することにより(2分探索法で行なう)、入力されたスペルの、見出しテーブル中での存在可能な範囲が完全に決定される。

以上をまとめると、図6-11のような構成となる。入力されたスペルによりインデックステーブル→見出しテーブルの順に検索する。もし見出しテーブル中に見つからなければ、検索を終了する。見つかった場合は見出し部をアクセスし、この見出し部から、発音・用例・品詞・活用形などのさらに必要な下位情報がとり出される。

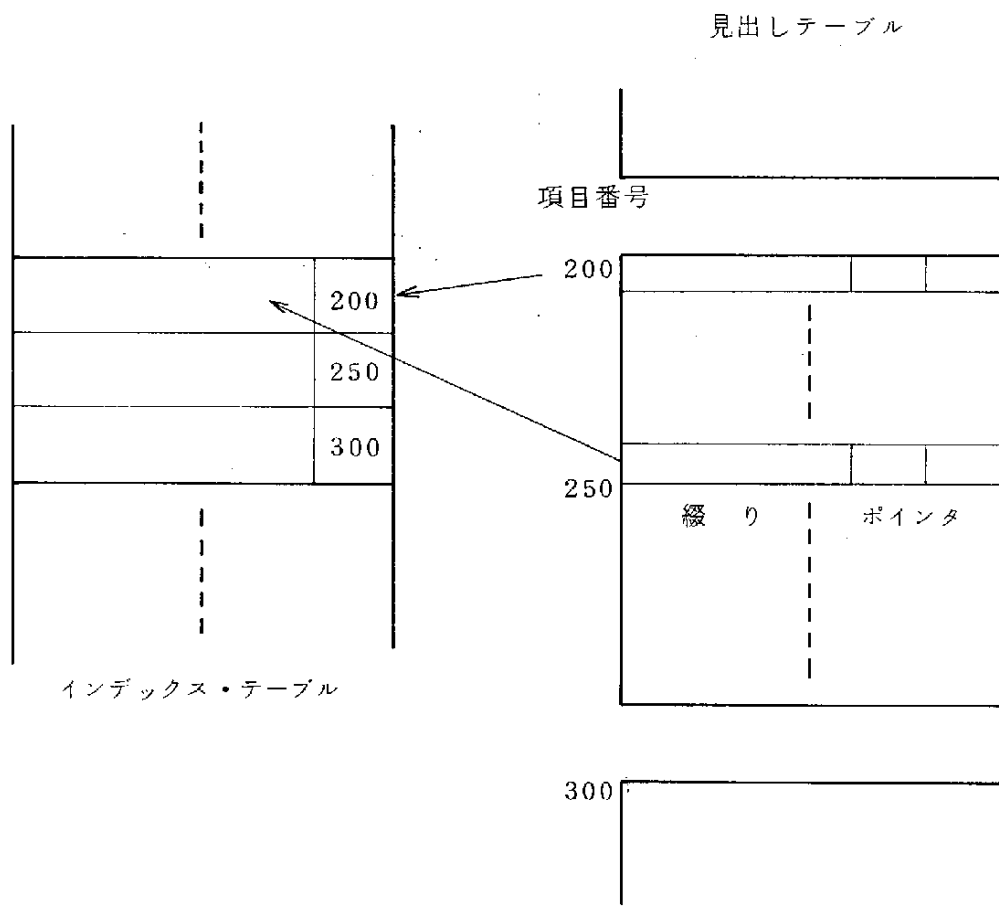


図6-11 英和辞典のインデックス構造

6.3.7 辞書引きの手順

このデータベースの実際の検索手順と、変化形の原形への変換について述べる。

1) 検索の手順

実際の辞書引きにおける検索手順を以下に示す。

- i) スペース，ピリオド，ハイフンの除去
- ii) インデックステーブルの検索 — 2分割法を用いてインデックステーブルを検索する。
- iii) 見出しテーブルの検索 — インデックスにより決定された範囲（50項目）を順に検索する。
- iv) 見つかった場合には端末に表示する。さらに入力スペルを見出しテーブル中のスペルと比較して，一致するものは全て表示する。
- v) 見つからない場合，規則変化の復元をして予想される原形全てについて(ii)～(iv)を繰り返す。

2) 規則変化の復元

規則変化は以下のように語尾が変化するだけであるから，比較的簡単に原形に復元できる。見出しテーブルの記憶容量を小さくするため，規則変化は登録せず，検索時にプログラムで復元する。

- i) —s 名詞の複数形または動詞の三人称単数現在形
- ii) —ed 動詞の過去形および過去分詞
- iii) —ing 動詞の現在分詞
- iv) —er 形容詞または副詞の比較級
- v) —est 形容詞または副詞の最上級
- vi) —ness 形容詞の名詞化
- vii) —ly 形容詞の副詞化

ここで，(vi)と(vii)は正確に言えば語形変化(inflection)ではない。しかし，「コンサイス英和辞典」では，これらは独立の見出しを与えられ

ず、派生語として項目の辞書記述中に埋め込まれていることも多いため、便宜的にこの種の派生形も原形に復元している。

原形の復元は、ここに挙げた語尾を単純にとり去るだけでなく、考え得るすべての原形を出力し、辞書引きを行なう。

また、語尾により原形の品詞が限定される場合には、原形に戻す際に品詞のタイプを決定しておき、この品詞タイプと見出し部に記述された品詞コードが一致するものだけを表示する。

検索結果の表示は、見出し、重要度、品詞、関連語、派生語、合成語、発音、変化形、語義および同義語、用例、慣用句のそれぞれの項目にわけて行なわれる。このうちの一部の情報だけを表示して欲しいという利用者のために、ディスプレイ、コントロールのオプションがあり、一部の項目について表示しないこともできる。

実際に辞書引きした結果を図6-12に示す。

```
INPUT SPELL (, DISP_CNTRL (KANREN, MASEI, GOSEI, HATSUON, HENKA, GOGI, YOREI, SEIKU))
```

```
FOREWENT
```

```
foregone 図1
```

品詞	他
発音	fɔːrgəʊ / fɔː -
語変	ɪ - wɛnt [-wɛnt] ; ɪ - gone [-gɔːn / -gɔːn]
語義	(古)に先だつ、より前に行く、(precede)。

```
forewent
```

品詞	***
発音	-wɛnt
語義	foregone 図1, 図2の過去形。

図6-12 辞書引き例(語形変化) - foregone (処理時間 49msec)

6.3.8 慣用句・用例による辞書引き

6.3.5で述べたように、慣用句・用例による辞書引きも、それを構成する各単語についての辞書引きを繰り返すことにより、データベースの変更および追加なしに行なうことができる。

基本的には、利用者による入力文(句)中に含まれる単語を、もっとも多く含む辞書中の慣用句・用例が取り出される。この際、入力文(句)中の単語および用例・慣用句中の単語は、いずれも語形変化している可能性があるので、それらは原形に復元してから一致がとられる。また、同じ単語が含まれていても語順が異なる場合には一致とはみなされない。慣用句・用例によって辞書引きを行なった例を図6-13に示す。

```
INPUT SPELL C, DISP_CNTL (KANREN,HASEI,GOSEI,HATSUON,HENKA,GOGI,YOREI,SEIKUI)
```

```
SHE KEPT HER FINGERS CROSSED
```

```
keep [have] one's fin-gers crossed
```

意味 | [cross one's ~sをして] じっと事態の好転[成洗]を祈る。

例1 She kept her fingers crossed (検索時間 178msec)

```
INPUT SPELL C, DISP_CNTL (KANREN,HASEI,GOSEI,HATSUON,HENKA,GOGI,YOREI,SEIKUI)
```

```
HE HAS TWO FACES
```

```
have two face s
```

意味 | 二心(ふたごころ)をいなく、裏表がある；[ことばが]二様(によう)に」れる。

例2 He has two faces.(検索時間 235msec)

図6-13 慣用句・用例による辞書引き

6.3.9 記憶容量に関する評価

前述のとおり、英和辞典を現在のレコード・フォーマットが全部データベース化してファイルに入れたとすると、70メガバイトを越える。これは各データ項目のフォーマットが固定長であり、それぞれのファイルのレコード長は、最も長いものに合わせられているからである。これを改善するためには、より短い固定長レコードのリンクによって不定長データを表現することが考えられる。現在、このようにデータ量を減少させるための記憶方式を検討中であるが、試算によれば現在の容量の半分以下30メガバイト程度になる予定である。

6.3.10 速度に関する評価

前にあげた辞書引き例には、すべて検索に要したCPU時間を添えている。これによると、検索して表示が終了するまでの時間は、ほぼ表示されたデータの量に比例していると考えられる。

見出し部へのアクセスは、インデックステーブルの使用により、非常に高速である。入力スベルが見出しテーブル内に登録されている時に見出し部へのポインタを得る時間と、登録されていないことを知る時間はほとんど同じである。登録されていないスベルを入力した場合には約4 msecで'NOT FOUND'が表示される。登録されている場合についても、約4 msecを差し引いた残りが、見出し部を含めて各項目別ファイルのアクセス時間と、その表示の時間に費されているといえることができる。

慣用句・用例による辞書引きの場合は、内部で種々の処理を行なっているため、かなり時間がかかる。現在'F'から始まるデータしか入っていないのでこれだけの時間ですんでいるが、辞書が全部入った時には、各単語で引けるようになるため、辞書引きの時間は2～5, 6倍(構成する単語の種数および個数による)はかかると考えられる。

この対策としては、慣用句、用例を整理し、単語分解したファイルを作っ

ておくことが考えられる。これらは全て原形に戻されていて、入力スペルを原形に戻すだけで簡単にマッチがとれる。あるいは、辞書引き専用のハードウェアを実現することが考えられよう。

第6章 付録①「定型化データのフォーマット」

(1) 見出しレコード

見出し、重要度等の項目とこの見出しに従属する種々セグメントを管理するテーブルを持つ。1見出しにつき1件切り出されている。

KEY		見出し	重要度	外来語		共通表示	管理テーブル	
大	小			区分	同義語		その1	その2
		40	2	2	20	20	e	f

a. 見出し

辞書中の見出しを切り出したもの

b. 重要度

辞書中の重要度を切り出したもの

' _ _ ' ... 非重要語

' * _ ' ... 重要語

' * * ' ... 最重要語

c. 外来語

辞書中の外来語情報を切り出したもの

区分…

{ 上位1文字一半ば英語化されているものは '1'
 { 下位1文字一言語名

{ フランス = '1'
 { ドイツ = '2'
 { イタリア = '3'
 { ラテン = '4'
 { ポルトガル = '5'
 { スペイン = '6'
 { 日本 = '7'

同義語…英語による同義語

d. 共通表示

品詞レベルで切り出されるセグメントに共通してかかる慣用表示や専門語指示情報を切り出したもの

〔例〕

bludge〔発音〕〔オーストラリア俗〕 vi., vt. 説明部

a., ad. 説明部

上記において'〔オーストラリア俗〕'は、品詞レベルで切り出されるそれぞれの説明部にかかるので、この慣用表示部を共通表示フィールドに切り出した。

e. 管理テーブル その1

見出しレベルで切り出されるセグメントを管理するためのテーブルである。

関連語 リンクKEY		派生語 リンクKEY		合成語 リンクKEY	
1st	件数	1st	件数	1st	件数

f. 管理テーブル その2

見出し／品詞レベルで切り出されるセグメントを管理するためのテーブルである。

品 詞	その 1		その 2		その 3		その 4	
	発音 リンクKEY	語形変化 リンクKEY	語義 リンクKEY	用例 リンクKEY	成句・慣用句 リンクKEY	語法・文法 リンクKEY		
	1st 件数	1st 件数	1st 件数	1st 件数	1st 件数	1st 件数	1st 件数	

テーブル数…その 1～その 5 まであるテーブルのうち使われているテーブル個数を示す。

品詞……………辞書中の品詞略号をコード化したもの

(2) 関連語レコード

辞書中の '▷' で示される関連語を切り出したもので 1 見出しにつき無しか、あっても 1 件である。

KEY		見出し リンク KEY	関 連 語	意 味
大	小			
			70	50

(3) 派生語レコード

辞書中の [名] , [形] , [動] , [副] で示される単語, 及びボールド体で記されている '～ly' , '～・ness' 等の単語を切り出したものであり, 1 見出しにつき複数件切り出されていることがある。

KEY		見出し リンク KEY	派 生 語	意 味	品 詞
大	小				
			50	50	

品詞……………派生語の品詞を示す

{ 名詞 = '1'
 副詞 = '2'
 形容詞 = '3'
 動詞 = '4'

(4) 合成語レコード

辞書中のボールド体で記されている合成語を記したものであり、1見出しにつき複数件切り出されていることがある。

KEY		見出し リンク KEY	合 成 語	意 味
大	小			
			50	100

(5) 発音レコード

辞書中の〔 〕で囲まれた発音をそのまま切り出したものである。

辞書中にサブ見出しがあり、品詞が変わることにより発音が違ってくるときは、1見出しにつき複数件あることがある。

KEY		見出し リンク KEY	品詞 ポイ ンタ	発音
大	小			
50				

(6) 語形変化レコード

辞書中の()で囲まれた語形変化をそのまま切り出したものである。

辞書中にサブ見出しがあり、品詞が変わることにより語形変化が違ってくるときは、1見出しにつき複数件あることがある。

KEY		見出し リンク KEY	品 詞 ポイ ンタ	語 形 変 化
大	小			
50				

(7) 語義レコード

辞書中の語義を語義% 単位に切り出したものであり1見出しにつき複数件あり得る。

KEY		見出し リンク KEY	品詞 ポイン タ	同義・対義 参考語 1st リンク 件数	未使用	話	義
大	小						
					2	150	

(8) 用例レコード

辞書中の ' 2 ' 記号で始まる用例を1文単位で切り出したもので1見出しにつき複数件あり得る。

KEY		見出し リンク KEY	品詞 ポイン タ	用 例	意 味
大	小				
				80	80

(9) 成句・慣用句レコード

辞書中のボールドイタリック体で記されている成句・慣用句を1文単位で切り出したもので1見出しにつき複数件あり得る。

KEY		見出し リンク KEY	品詞 ポイン タ	成句・慣用句	意 味
大	小				
				50	150

(10) 語法・文法解説レコード

辞書中の「→」で示される語法・文法解説を切り出したものであり、語法・文法の解説は見出し／品詞レベルで記述されているため複数件あり得る。

KEY		見出し リンク KEY	品詞 ポイン タ	語 法 ・ 文 法 解 説
大	小			

200

(11) 同義・対義・参考語レコード

辞書の語義の中に含まれる同義・対義・参考語を切り出したもので1見出しにつき複数件あり得る。

KEY		語 義 リンク KEY	見出し リンク KEY	区 分	同 義 ・ 対 義 ・ 参 考 語
大	小				

50

区分…同義・対義・参考語 区分

同義語 = ' 1 '

対義語 = ' 2 '

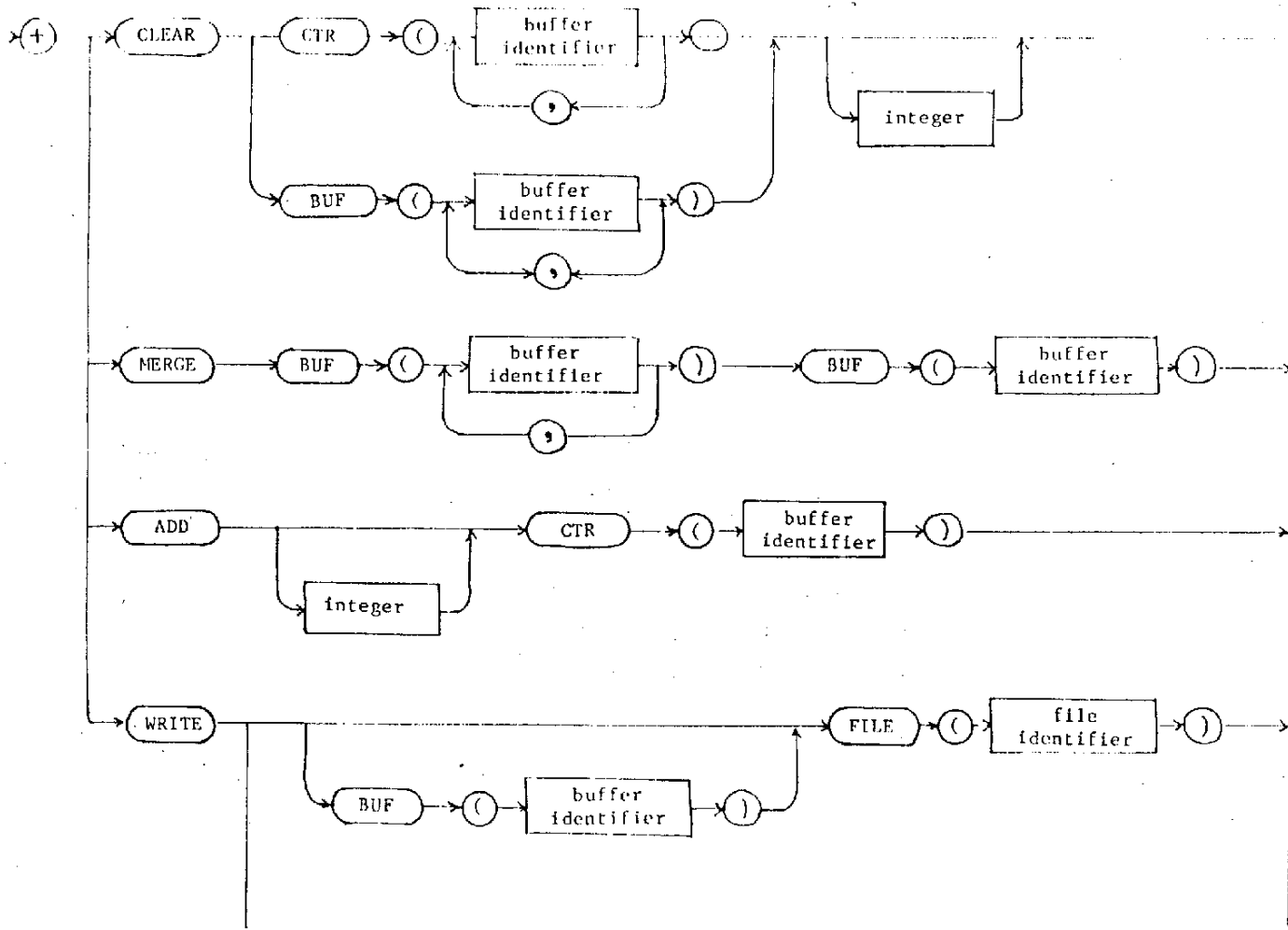
参考語 = ' 3 '

(12) エラーレコード

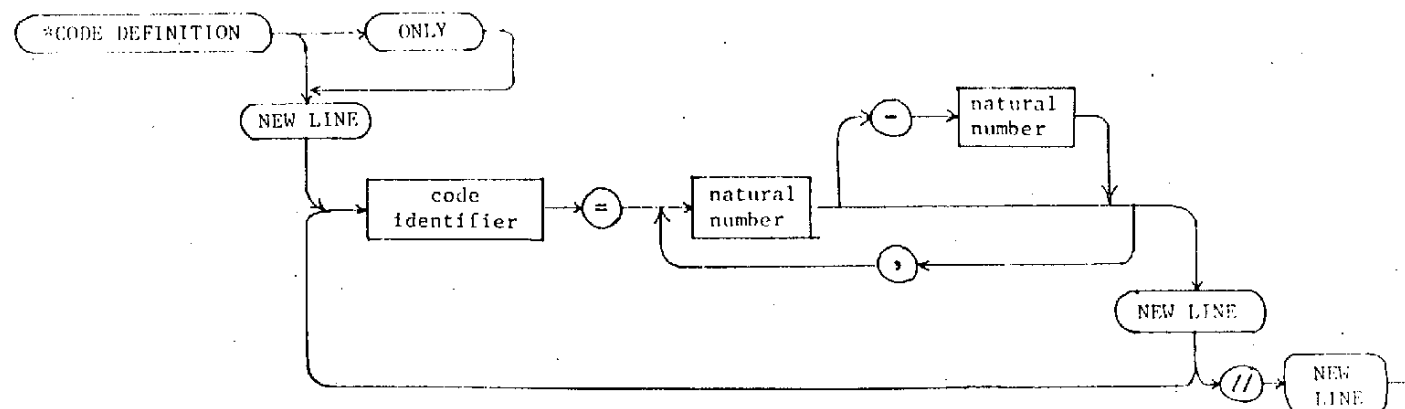
エラーポインタ		エラー原因	未 使 用
・	・		

10 2

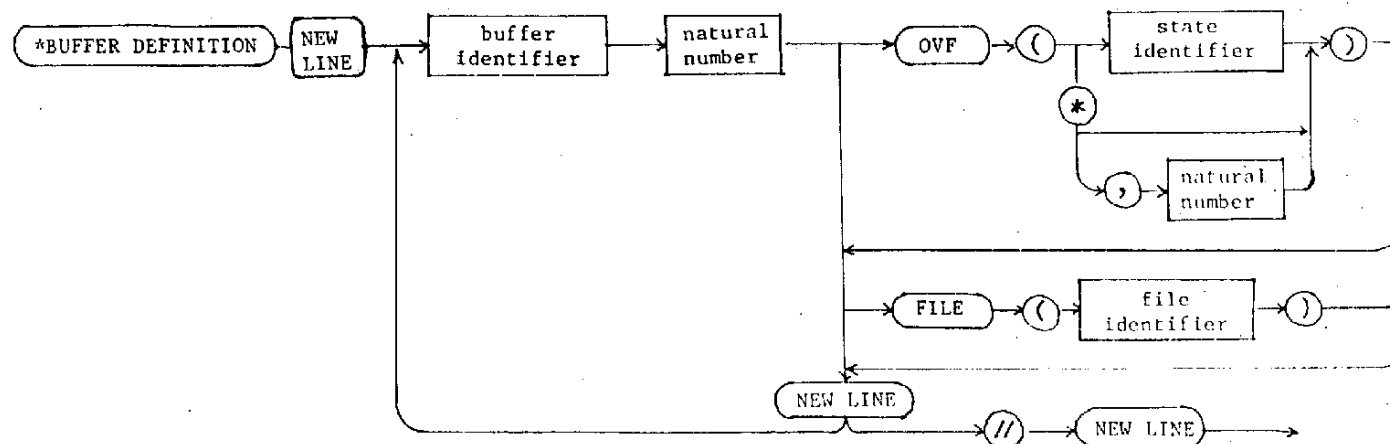
文字ポインタ (4 Bytes Fied Binary)
レコード№…エラーとなったオリジナルデータのレコード№
(= 前述 1・2 の「シーケンス№」参照)



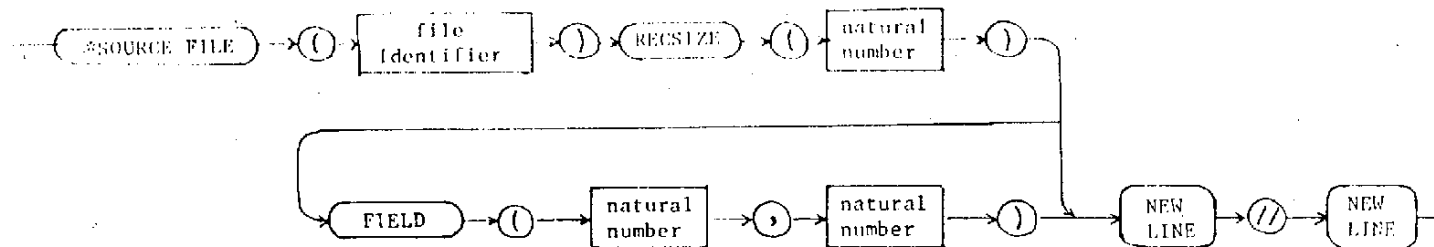
code definition



buffer definition



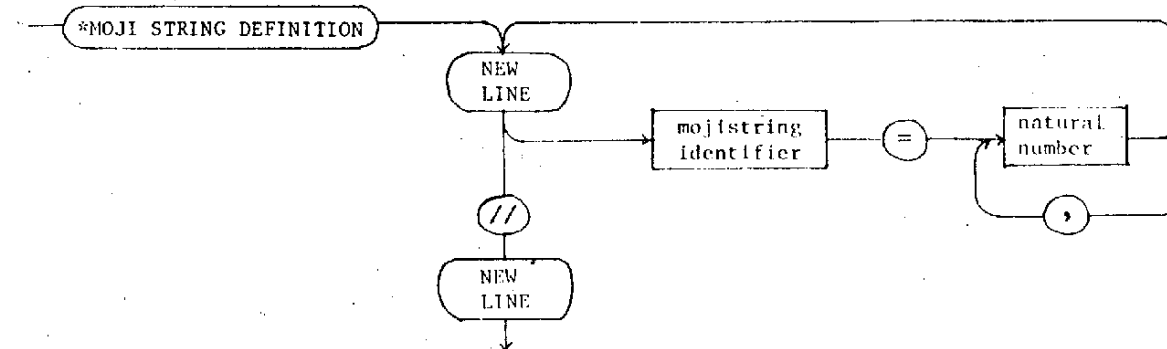
source file definition



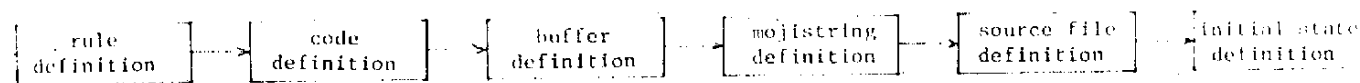
initial state definition



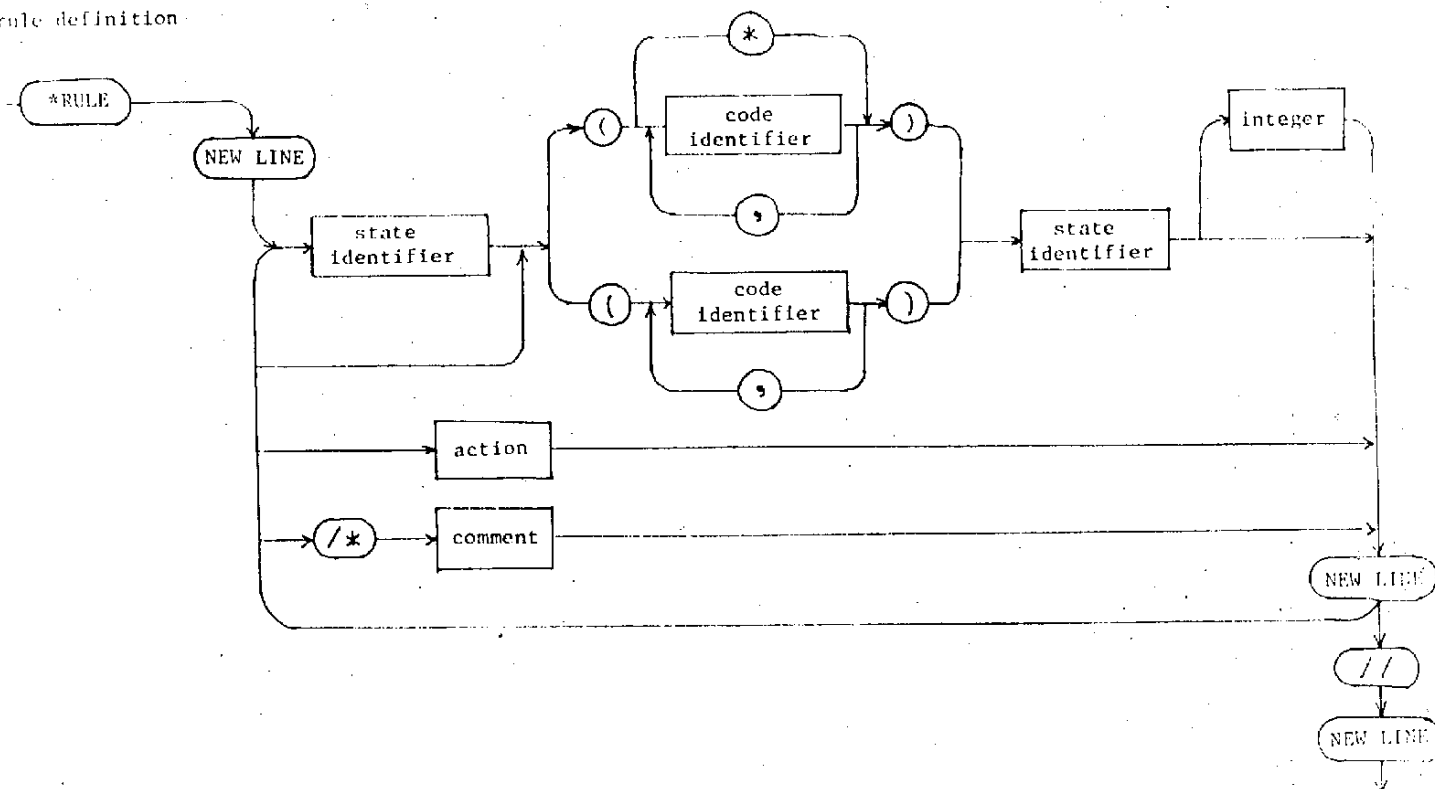
mojistring definition

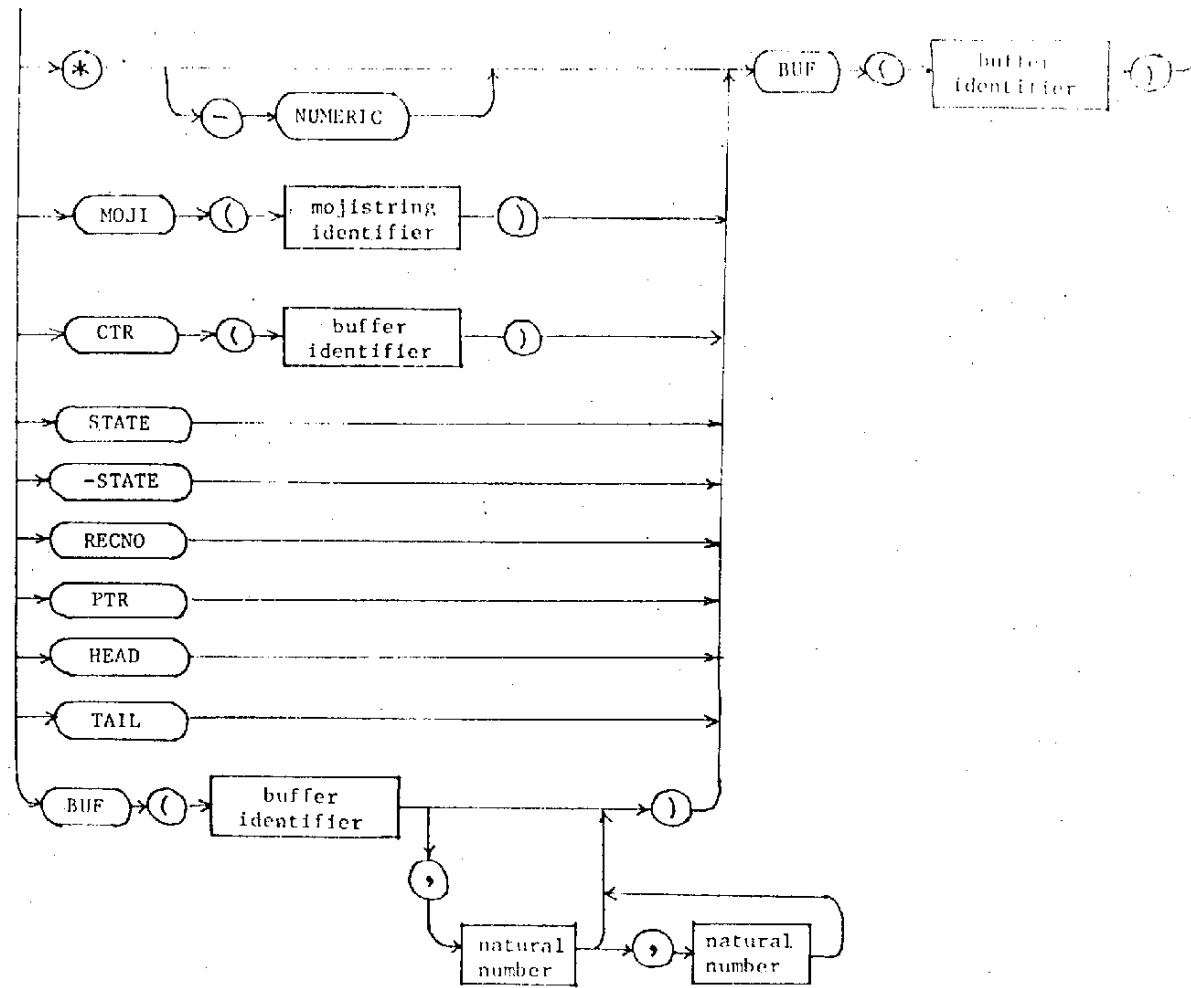


TOTAL SYSTEM



rule definition





— 禁 無 断 転 載 —

昭和 55 年 3 月 発 行

発行所 財団法人 日本情報処理開発協会

東京都港区芝公園 3-5-8

機械振興会館内

TEL (434) 8211(大代表)

印刷所 山 陽 株 式 会 社

東京都港区虎ノ門 1-9-5

TEL (591) 0248

