格構造を使った日本語文の意味解析システムの開発

杉 本 英 二

0. はじめに

コンピュータが人間のコトバを理解するという自然言語理解は，人工知能の重要な課題である。言語学の分野では，表層構造を深層構造で説明する生成文法を中心にした研究が行われて来たが，意味理解の研究には十分ではないと考えられている。これを補うためにフィルモアは格文法（Case Grammer）を提唱し［Fill］，語と語の間の係受けを中心とする日本語の意味分析には格文法が好都合であることが分って来た。

日本語の文法というと学校文法が代表的なようだが，「学校文法は，意味論はまったく考慮していないから，本来『格』のような概念を組込むようにはできていない」［郡司］，「無原則的な，首尾一貫しない活用語尾の特徴づけをもたらした」［寺村2］というので，新たな日本語文法が必要である。そのような文法として，文の生成規則と名詞句に関しては奥津文法［奥津］，用言の活用に関しては寺村文法［寺村］を採用した。文法の意味に係わる部分に関しては，表層格の認定を結合価文法［石綿］に従って行ったが，奥津文法の深層格との対応にはまだ研究が必要である。

自然言語理解のシステム開発には，構文情報，意味情報，文脈情報，談話情報，専門知識，常識などの情報を駆使することになるが，談話や状況や心的情報などの高度な情報の表現はまだ達成されていない［野村1］。一方，形態素

解析，かな漢字変換，構文解析の技術はかなり確立されている［野村2］。その中でも，限定節文法は，構文解析，意味解析，文脈解析とを融合した自然言語処理が可能な文法記述法で，これで書かれた文法には種々の構文解析手法が適用可能であり［田中］，中でもＢＵＰ［松本］という高速の構文解析手法が知られている。

本研究は，奥津文法・寺村文法・結合価文法を限定節文法で記述し，これをBUP システムで高速に処理し，日本語の単位文の意味を深層格のレベルでとらえるシステムを開発した。開発言語はパソコン上の Arity／Prolog を使った。文法記述は26本の生成規則と120本の辞書項目である。結果は良好で「太郎が買った本を花子がよむらしい」を６秒で意味解析した。

１．日本語文法

（名詞句の構造）

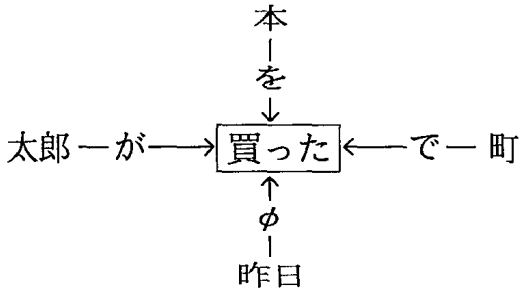
奥津は，生成文法の立場から日本語文の名詞句の構造を明らかにする研究を行った［奥津］。名詞句は，文の中で，動詞の主格や対象格や目的格となる重要な素材である。（１）で，花子が本を読んだことが分るが，その本は太郎が買って来たものであったことも分らなくてはいけない。つまり，（１）は（２）の２つの文から構成されている。これが**同一名詞連体修飾**である。同一名詞というのは，（２ａ）の本と（２ｂ）の本が同一の名詞であることによっている。奥津は，「太郎が買った本」という時，本は修飾する文の「買う」になんらかの**格関係**（ここでは対象格「を」）で係っていると見るのである。

（１）太郎が買った本を花子が読んだ。

（２）ａ太郎が本を買った。
　　ｂ花子が本を読んだ。

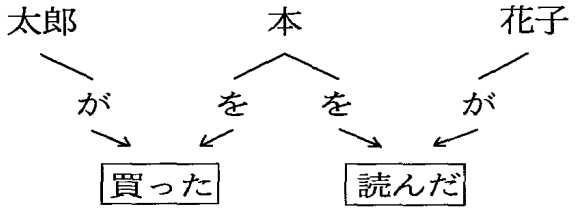
修飾される名詞が同一であることを図を使って示そう。文を，動詞と動詞に係る部分とに分けるとすれば，（３）の文の構造を［図１］のように，動詞を中心に置き，その周りを格助詞を伴う名詞（補語）が取り巻く姿として表すことができるだろう。ただし通常は時間を表す名詞には格助詞を付けないので，空の格助詞φがあるものとする。

（３）　昨日，太郎が町で本を買った



［図１］「昨日，太郎が町で本を買った」の構造図

この図の表現方法を使えば，（２）を［図２］のように表すことができる。



［図２］「太郎が買った本を花子が読んだ」の構造図

しかし（１）は，単に（２）の２つの文が並列的に並んでいるのではない。もし２つの文が単なる並列ならば，（４）の文も成り立つことになろう。だから，２つの文が何らかの構造で繋がっていることをが分るので，コンピュータ処理では，その表現の工夫が必要である。

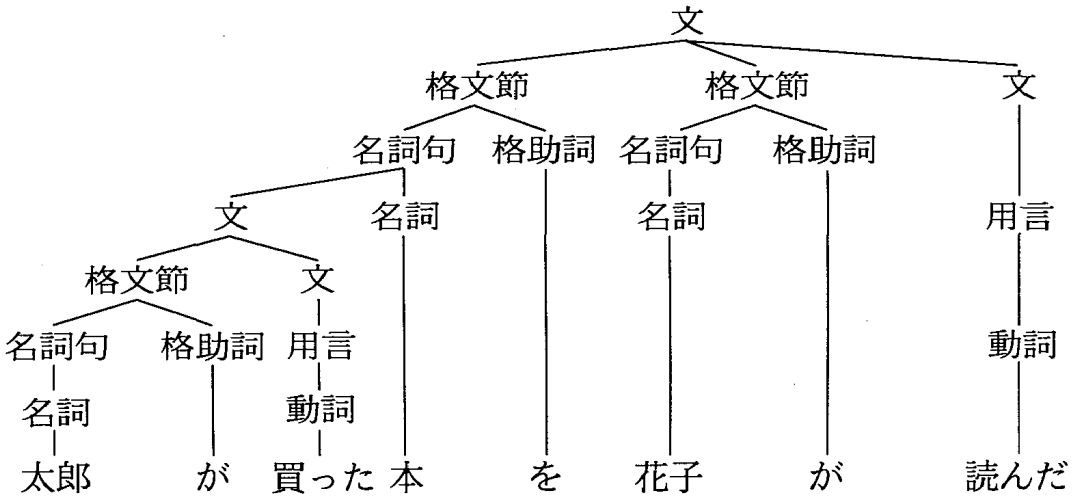
（４）　花子が読んだ本を太郎が買った。

ところで，（１）から（２ａ）のように「本を」の格助詞「ヲ」をどのよう

に導きだしたのであろうか？人間は（1）から（2）の2つの文を連想することが簡単に出来るのであるが，コンピュータにこれをさせることすれば，そのやり口はいったい何であろうか？ここに意味解析の必要性が出てくるのである。

（構文規則）

我々は，文が複数の文節から出来ていることを知っている。特に日本語では，動詞などの用言句が必須となっていて，この用言句に格助詞を伴った**格文節**が複数個用言に先行し，誰が何を何処で等を示す補語で，文の内容を表現する。（1）のように「太郎が買った本」というように，文が名詞を修飾しても，その全体は名詞の性格が保持されている。このようなものも名詞として取り扱うために名詞句を置く。文のこうした構成概念を合理的に表すのに「木構造」が使われる。その例を［図3］に示す。



［図3］「太郎が買った本を花子が読んだ」の構文木

［図3］の木を作る規則を**書き換え規則（生成規則）**といい，例えば（5）のように与えることも出来る。

（5）　a　文　　－＞　用言　｜　格文節　文

　　b　格文節　－＞　名詞句　格助詞

　　c　名詞句　－＞　名詞　｜　文　名詞句

d　用言　　－＞　動詞
e　格助詞　　－＞　が　｜　を
f　名詞　　－＞　太郎　｜　花子　｜　本
g　動詞　　－＞　買った　｜　読んだ

（5）には a から g まで7本の規則がある。各規則は，左辺の記号を右辺の記号で書き換えることが出来るということを表している。1つの記号を複数個の記号のどれかに書き換える時，複数個の記号を並べ，それらの間を縦棒の記号｜で区切る。例えば（5a）は2通りの書き換えが可能である。1つは，文という記号を用言という記号に書き換え可能ということだし，2つは，文という記号を格文節と文という2つの記号の並びに書き換え可能ということである。どちらの書き換え規則を使っても構わないことを表している。以下同様にして次々に書き換えていき，（5f, g）の規則を適用すると，この規則の右辺の記号はこれ以降書き換えが出来なくなる。（5f, g）の右辺の記号を**終端記号**といい，その他の記号を**非終端記号**，もしくは**文法記号**と呼んでいる。初期記号「文」から出発してすべての記号が終端記号になったら，それは生成規則から導出される正しい文であるということである。

次に（5）から（1）が得られることを示そう。まず記号「文」から書き換えを始める。書き換えが行われたら適用された規則の番号を右側に表示する。この表示で使われている左右の記号は，右辺に2つの書き換え規則がある場合，そのどちらかを示すためのものである。

（6）文－＞格文節　文　　a右
－＞名詞句　格助詞　文　　b
－＞文　名詞句　格助詞　文　　c右
－＞格文節　文　名詞　格助詞　文　　a右
－＞名詞句　格助詞　文　名詞句　格助詞　文　　b
－＞名詞　格助詞　文　名詞句　格助詞　文　　c左
－＞太郎　が　文　名詞句　格助詞　文　　f左, e左

－＞太郎　が　用言　　名詞句　格助詞　文　　a左
－＞太郎　が　動詞　　名詞句　格助詞　文　　d
－＞太郎　が　買った　名詞句　格助詞　文　　g左
－＞太郎　が　買った　名詞　　格助詞　文　　c左
－＞太郎　が　買った　本　を　文　　f右，e右
－＞太郎　が　買った　本　を　格文節　文　　a右
－＞太郎　が　買った　本　を　名詞句　格助詞　文　　b
－＞太郎　が　買った　本　を　名詞　　格助詞　文　　c左
－＞太郎　が　買った　本　を　花子　が　文　　f中央，e左
－＞太郎　が　買った　本　を　花子　が　用言　　a左
－＞太郎　が　買った　本　を　花子　が　動詞　　d
－＞太郎　が　買った　本　を　花子　が　読んだ　　g右

このプロセスを逆に進めて，データとして与えられた文から記号「文」を得ることも出来る。［図3］の木を見ると，記号「文」は上にありデータは下にあるので，書き換え規則を左辺から右辺方向に使うことを**トップダウン**，この逆方向を**ボトムアップ**という。

1組の生成規則を与えると，生成規則に従う文の集合を決定できる。問題は，日本語の基本的な文の全てを生成出来て，なおかつ，非終端記号に対応する概念が構文のみならず文が表している意味の面でも合理的，統一的取扱と解釈が出来ることが必要である。さらには，それらの規則が単純で見通しが良いことも重要である。この条件に合致するものとして，奥津の生成文法［奥津］を採用した。

（奥津の生成規則）

（7）a　文　　—＞　中立文　文体素
　　b　中立文　—＞　（文頭詞）（叙述文（文末詞））

c　叙述文　—>　平叙文（判断詞）

d　判断詞　—>　（｛はずだ，はずがない｝時制詞）

（｛らしい　時制詞，かもしれない，だろう｝）

e　平叙文　—>　核文　時制詞

f　核文　—>　格要素のリスト　用言句

g　用言句　—>　（核文）用言

h　格要素　—>　名詞句　格助詞

ただし名詞句は格助詞と組合わせることによって，次の深層格を持つものである：

［時，所，主語，直接目的，間接目的，手段，出発点，目標，対称，共同，引用，期間，移動，目的，理由］

（ただし時間を表す格助詞はないので，空格助詞を持つことにする）

i　名詞句　—>　｛叙述文　名詞句，（補足句）名詞，文｝

j　補足句　—>　｛叙述文，出発格要素｝（｛という，数量名詞｝）

生成規則の表現のためにいくつかの記号を導入した。それを（8）に説明する。

（8）a　（　）で囲まれた項目は，選択項目である。

b　｛　｝で囲まれた項目は，その中の１つを選択する。

c　リストとは，複数個の並びをいう。従って，格要素のリストは，格要素を複数個並べたもので，例えば「昨夜　太郎が　本を」というような並びがそれである。

本研究では，文の基本的な部分に注目し，発話者の心的部分が反映される部分である文体素，文頭詞，文末詞の部分を省略し，次の生成規則（9）を開発の対象とした。ただし，文末詞としての疑問詞や用言の活用で表示される命令

形は，重要であるので開発したシステムでは考慮した。だが今回のプログラムには開発時間の関係で組込まれていない。その為，開発システムでは（9a）が「文　－＞　叙述文」となっている。

（9）a　文　　—＞　叙述文（文末詞）
　　b　叙述文　—＞　平叙文（判断詞）
　　c　判断詞　—＞　{はずだ時制詞，はずがない時制詞，
　　らしい時制詞，かもしれない，だろう}
　　d　平叙文　—＞　核文　時制詞
　　e　核文　　—＞　格要素のリスト　用言句
　　f　用言句　—＞　（核文）用言
　　g　格要素　－＞　名詞句　格助詞
　　h　名詞句　－＞　{叙述文　名詞句，名詞，文}

2．確定節文法と構文解析

（構文解析の問題）

文脈自由文法の構文解析の技法として，ＡＴＮやアーリー法はトップダウン解析であり，ＣＹＫ法やボトムアップ・チャート法はボトムアップ解析である。両者の利点を生かした双方向縦型探索法というのもある。それはボトムアップの各時点において得られた結果を，生成規則の左隅とするような直上の生成規則を予測する方法である。この予測で得られた生成規則の残された目標を，接続するデータのトップダウン解析の予測とするものであるから，ボトムアップとトップダウンが交互に行われる効率的な方法である。

一般に文法記述には文脈自由文法が使われているが，これでは自然言語の中で文脈依存を含む文を使うことが出来ない。何らかの補強を施すことによってこの部分を回避することが必要である。そのために様々の工夫がありえるが，それらの工夫においては，次の問題が指摘されている［田中］。

(10)

a 構文解析速度

b 構文解析過程の制御

c 構文解析と意味解析・文脈解析との融合

d 文法の記述能力

e 文法の記述形式

(10a)の解析速度を犠牲にしないことは当然の要求であろう。田中は，人間の自然言語理解では構文解析，意味解析，文脈解析を同時に融合していると考え，そのように計算機処理でも可能なようにすることが重要だと考え，(10b，c)の2つを挙げている。(10d)は上で指摘した文脈依存の問題であり，(10e)はそのような工夫が文法記述の量の増加や，複雑化にならないようにすることも，ソフトウエア工学として重要な検討事項である。田中の言語研究のこれまでの経験から，文法を確定節文法で記述し，これに補強と意味解析を組合わせた方法が上記の問題解決の最も良い方法であると述べている［田中］ので，今回の開発はこの枠組みに従った。

(確定節文法と Prolog 変換)

確定節文法(DCG)は，(9)で書かれた上記の文法形式をそのまま Prolog のプログラムに変換して実行させるような文法記述法である。【付録1】は(9)の判断詞に関係する部分を省略した文法を確定節文法で書いたものである。この記述を(9)の表現と比較すると，極めて自然で簡単であることが理解されるだろう。各行の右側に(9)の書き換え規則に対応するものがあれば，その項目番号を表示した。プログラムの14行以降は辞書の部分である終端記号を列挙している。辞書項目は，品詞名を左辺に書き，右辺には単語の語彙を［ ］で括って表す。

確定節文法記述に書き直す時，特に工夫した2点を説明する。(9d)では

平叙文は核文と時制詞の2つに分割できることを意味しているが，簡単には文を分割することが出来ない。奥津は時制詞を「る（ru）」のuと「た（ta）」のaにあるとしているが，データとして与える文をローマ字で入力させる訳にはいかない。核文の解析では用言の活用解析を行うことになるので，この時点で時制が決定できる。このプログラムでは，4行目の用言句解析で時制情報も得られるとしている。

（9e）の「核要素のリスト」を実現するためにプログラムでは4，5行目の様にした。これは用言句の前に複数個の格要素を連ねる再帰的な手法である。平叙文を2つの格文節と用言句の並びに書き換えるプロセスを（11）で説明する。書き換えの右側には適用されたプログラムの行数が書かれている。これで，格文節を必要な数だけ用意することが出来る。

(11)	平叙文	-->	核文＋時制	3
		-->	格文節，核文＋時制.	5
		-->	格文節，格文節，核文＋時制.	5
		-->	格文節，格文節，用言句.	4

確定節はPrologに読み込まれる時に，それぞれの述語に2つの変数が追加される。追加には一定のルールがある。それを（12）に例示する。（12a）は変数を順番に付けていく原則を示している。（12b）は辞書項目の変形ルールを示している。（12c）は文脈依存を確定節文法に取り入れることがPrologプログラムとして実現できることを示している。word1 と word 2に挟まれた時だけ，aをword 1　b word 2と書き換えることができるというのは，語彙の照応を文法として定義可能であることを示している。【付録1】の文法はこの様な変形を受けて，Prologプログラムとなる。確定節文法の記述は単純なルールではあるが，Prologプログラムの些細な部分に気を取られることなくコンピュータで動作可能な文法を定めることが出来る。

(12) a1 a --> b, c, d.
a2 a(S1, Sn) : − b(S1, S2), c(S2, S3), d(S3, Sn).

b1 a --> [word].
b2 a([word | S], S).

c1 a --> [word 1], b, [word 2].
c2 a([word 1 | S1], Sn) : − b(S1, [word 2 | Sn]).

(ボトムアップ解析)

たいていの Prolog 処理系には確定節文法の変換ルーティンがあるので，このファイルをそのままコンサルトすれば良い。プログラムの実行は（13）のように解析したい品詞名を述語名とするコマンドを入れる。パラメータはデータとして与える文のリストと，空リストである。ところが（13）の質問を入れても答は出ずに，しばらく待たされた揚げ句，スタックが一杯になったというようなメッセージが出力されて終了する。その原因は，文法が持つ左再帰規則によって無限ループが発生したからである。

左再帰規則というのは(14)に示すように，規則の左辺と同じものが右辺の最も左側にある規則のことである。【付録 1】には(14)のような規則は見当たらないが,それでもこれがあるというのは,間接的に左再帰を形成する組合わせがあるからで，それは 4 行目と 9 行目の 2 つの規則を組合わせると出来る(15)。

(13) ?− 文([太郎,が,町,で,本,を,買った], [ ]).

(14) a --> a, b.

(15) 4 核文＋時制 --> 用言句.
9 用言句 --> 核文＋時制, 用言.

↓

核文＋時制　　--> 核文＋時制,　用言.

一般に確定節文法の構文解析は，トップダウンである。トップダウン解析では左再帰規則があると，上記の様に動作しないのである。従って確定節文法で書かれた文法でも左再帰規則を持たないものであれば，解析は成功する。この確定節文法は欧米での自然言語解析の目的で開発されたものであって，欧米の言語のように言語の構造が右枝分れ［柴谷］である場合にはうまくいくが，左枝分れである日本語［柴谷］では本質的に左再帰構造を持つのでうまくいかない。

これを回避するには，ボトムアップ解析が適切である。この候補として，ＢＵＰシステム［松本］がある。これは確定節文法で書かれた文法をボトムアップを中心にトップダウンを交えながら双方向から解析する手法で，極めて高速である。このプログラムについての論文には方法論の記述があるがプログラムの全体がないが，一般に出版されている出版物の中でただ１つ畝見氏の８５行ほどのプログラムがあった［畝見］。これにはコンパイルエラーに引掛からない７箇所のミスタイプがあった（【付録２】参照）が，本文の説明を良く読みながらデバッグすれば取れるだろう。

ＢＵＰシステムによって，【付録１】のプログラムは順調に動作し結果を出力する。どのような解析をしたのかについては不明であるが，正しい文を与えた場合にはyes，正しくない文の場合にはnoの解答が出る。

3．詳 細 な 構 文 情 報

解析結果がイエスかノーかだけではなくて，正しい文ならその文の構造が分

る訳だから，それを表示したい。また【付録１】の２４行目から以降の様に動詞それぞれについて，現在形，過去形，連用形，命令形，未然形などと種々の活用形を定義することは無駄であるから，これを統一的に処理したい。このような工夫を本章で行う。

（確定節文法で構文情報を取るには）

【付録１】の書き換え規則のそれぞれの品詞に変数を用意し，この変数に構文上の情報を持ち帰らせることを考える。そのため，まず構文情報の表現を与える。規則が（16　a1，b1，c1）で与えられたとすると，その構文情報をリスト表現で（16　a2，b2，c2）の様に書くことにする。リストの先頭を品詞名，残りは品詞を構成する要素の情報のリストとする。

```
(16)   a1      a  -->  b.
       a2      [a，bの情報]

       b1      a  -->  b，c.
       b2      [a，bの情報，cの情報]

       c1      a  --> [word].
       c2      [a，word]
```

この情報をプログラムに反映するには（16）のそれぞれの形式について，確定節文法表現を（17）のようにすれば良い。

```
(17)   a     a( [a，X] )  -->  b(X).
       b     a( [a，X，Y] )  -->  b(X)，c(Y).
       c     a( [a，word] )  -->  [word].
```

この方法で【付録1】を書き直したのが【付録3】のプログラムである。【付録3】の第1行目の述語名がsになっているのは，特別な理由はないが，文のトップをsと置くことが習慣になっているので，「文」の代りにsと表すことにした。このプログラムで読み取りにくいところは，格文節を束ねて格文節リスト情報を作りだすところであろう。これはリスト処理に慣れれば，そう困難はないと思われる。

（寺村文法の動詞活用）

学校文法に従って動詞の活用を実現することは難しいことではない。しかしその活用を使って，意味処理を行うには文の他の要素との関係が重要になる。奥津は，助動詞，補助動詞などが補文をとる動詞であると考えて（7g）の生成規則を設定している。動詞の意味を活用と文中の他の要素との関連でとらえ直すと，学校文法よりは動詞の意味がより統一的になるという研究が寺村によってなされている［寺村1, 2］。この研究によると動詞を語幹の最後の音で，Ⅰ型，Ⅱ型，Ⅲ型という3つの型に分類しそれぞれの活用を次の様に与えている。

［動詞分類］

Ⅰ型　語幹が子音で終るもの

書く，　読む，　取る，　切る，　練る，　要る

kak－　yom－　tor－　kir－　ner－　ir－

特徴：－ru　で終らないもの。

でも－ruの直前の母音がa，u，oで終ればⅠ型

例）　有る，　売る，　取る

ar－　ur－　tor－

Ⅱ型　語幹が母音で終るもの

食べる，　見る，　着る，　寝る，　居る，　買う

tabe－　mi－　ki－　ne－　i－　ka－

特徴：－ru で終る。

Ⅲ型　来る，「する」と「する」の変種

研究する，議論する，恋愛する，失恋する，マークする，ストする

[活用表]

ムード	基本語尾		タ系語尾	
確言	V Ⅰ	－u	V Ⅰ	－ta －da
	Ⅱ	－ru	Ⅱ	－ta
	Ⅲ	－suru（基本形）	Ⅲ	－sita（過去形）
		－kuru		－kita
	A	－i		－katta
概言	V Ⅰ	－o	V Ⅰ	－taro ～－daro
	Ⅱ	－yo	Ⅱ	－taro
	Ⅲ	－siyo（推量意向形）	Ⅲ	－sitaro（過去推量形）
		－koyo		－kitaro
	A	－karo		－kattaro
命令	V Ⅰ	－e		
	Ⅱ	－ro		
	Ⅲ	－siro（命令形）		----------
		－koi		
	A	----------		----------
条件	V Ⅰ	－eba	V Ⅰ	－tara ～ －dara
	Ⅱ	－reba	Ⅱ	－tara
	Ⅲ	－sureba（レバ形）	Ⅲ	－sitara（タラ形）
		－kureba		－kitara
	A	－kereba		－kattara

保留	V　Ⅰ　－i	V　Ⅰ　－te ～ －de
	Ⅱ　－φ	Ⅱ　－te
	Ⅲ　－si　（連用形）	Ⅲ　－site　（テ形）
	－ki	－kite
	A　　－ku	A　　－kute
		Ⅰ　－tari ～ －dari
		V　Ⅱ　－tari
		Ⅲ　－sitari　（タリ形）
		－kitari
		A　　－kattari

注）この表の中で，Vは動詞，Aは形容詞を表す。

（動詞活用の実現方法）

動詞を語幹と語尾に分けて考える。語尾の活用を調べると活用の種類（M, Op）と時制（T）が得られる。Mはムード情報，Opはオプション情報とする。Ⅰ型の動詞には語幹末の子音に従って表層の発音が異なるから，音便などの処理を施す必要がある。このために語幹末の子音情報が必要になる。例えば「買う」の動詞定義を次の様にした。語幹の音韻はkaw－である。

動詞([買う, [M, Op], T]) －－> [買], 活用([w, M, T, Op]).

活用処理はそれぞれ3つのタイプごとに用意し，Ⅲ型には「来る」「する」の2つを用意した。それが活用Ⅰ，活用Ⅱ，活用Ⅲk，活用Ⅲsである。オプション情報が特に無ければその値をnonとした。活用の種類は，例えば「推量意向形」はM＝概言，T＝完了というようにムードと時制の座標で表すことにした。

その他判定詞ダ・デスや助動詞などもあり，文の中で重要な働きを行うが，

今回は動詞の実現のみにした。

（実行例）

以上の2点を【付録1】に新たに付け加わえたのが【付録3】のプログラムである。【付録3】を実行するには，まずＢＵＰシステムをPrologに読み込み，このＢＵＰシステムを使って，【付録3】のプログラムファイルを通常のPrologプログラムに変換しながら読み込む。そのコマンドは（18）である。変換されたプログラムは，そのままPrologシステムにあるので，（19）のように，解析のためのデータ文を与えると，解析結果が変数Ansに帰って来る。結果はリストで帰ってくるので，人間にとっては見にくいが，結果を巧く印刷するプログラムを作成すれば【付録4－1】の結果を得ることが出来る。

（18） ？－ bup（ ’付録3のファイル名’).

（19） ？－ goal（s, Ans,
[太, 郎, が, 買, っ, た, 本, を, 花, 子, が, 読, む, ら, し, い], [ ]).

4．意味解析

構文解析には，構文情報だけでは取ることができない誤った解析結果も含まれる。（19）の実験では4種類の解答のうち，意味として正しいものは1つだけであった。つまり25パーセントの正解率である。その誤った解釈の一部を【付録4－2】に示す。意味を考える人間だったらこうは解釈しないというような制約条件を構文解析の実行時の必要な箇所に組込めれば，明らかに無駄な解答は出てこないであろう。そのような制約条件を入れて，構文解析を助けるのが意味解析である。同時に，文の意味を集約することも出来るので，コンピュータが自然言語を理解する基礎となりえる。

（連用修飾の格となりえる名詞の制約条件）

用言に対して種々な格がある。そのような格として奥津は15種類の格を挙げ

ている（20）[奥津]。表層格の格助詞と深層格に1対1の対応があれば，格の決定が実に簡単に行われるのであるが，そうは簡単にはいかない。（21）に示すように，同じデでも，材料を示す手段格のデ，あるいは場所格のデ，さらには理由格のデなどがあり，紙や公園や宿題の意味や概念を知らないならば，それらが文の中でどうような働きをしているかが分らないであろう。

（20）

格	格助詞，格助詞相当語
時格	に
場所格	で
主格	が
第1対象格	が，を
第2対象格	に
手段格	で
出発点	から
目標格	へ
対称格	と
共同格	と
引用格	と
期間格	間
移動格	ながら
目的格	のために
理由格	で，だから，なので

（21）a　太郎は，紙で　飛行機を作った。

b　太郎は，公園で　飛行機を作った。

c　太郎は，宿題で　飛行機を作った。

従って，連用修飾の名詞が用言とどのような位置関係にあるのかを，格助詞以外に知る手立てが必要である。それが常識である。例えば，動詞「買う」の主語となりえるのは人間以外にはありえないというようなことである。このような観点から名詞の種類を整理する作業をしているのが，依存文法と呼ばれる分野である。石綿等は，用言を「体言＋格助詞」との結合関係でとらえ，それぞれの結合の型を体言の意味と格助詞の種類によって記述した『日本語用言の結合価』を作った［石綿］。まず名詞の意味特徴を表（22）とする。

（22）

名詞特徴の略称	内　　容
abs	抽象概念
act	行為
ani	動物
con	具象物
div	種々
hum	人間
loc	場所
num	数
mat	材料
temps	時
s	文

Nを名詞，Vを動詞として，それぞれの動詞の特徴を整理する。 その一部を（23）に示す。石綿等は，1154種類の用言に対してこの作業をしている。

(23)

a	買う(うらみを)	N(hum)が＋N(abs)を＋V
b	買う	N(hum)が＋N(con)を＋N(hum)に＋V
c		N(hum)が＋N(con)を＋N(hum)から＋V
d	読む	N(hum)が＋N(con)を＋V
e	作る	N(hum)が＋N(con)を＋N(con)で＋V
f		N(hum)が＋N(con)を＋N(mat)で＋V

以上の考えを，構文解析の途中で用言が確定した時に，格として係る名詞句の種類のチェックに適用することにより，常識外のものが格として係わることを制限する方策を組込むことにした。このために名詞の辞書にそれぞれの特徴を考慮して（22）の特徴を組込んだ。さらに（23）に対応する情報を動詞に組込んだ。ただ，動詞定義が冗長にならないように１つの動詞に２つの定義が石綿の研究でなされていても，それを１つに集約した。また，余分に場所格なども追加した。

以上の追加した部分のところにさらに法情報という項目を追加した。これは用言の活用情報をこの用言が作る核文や文の外部に伝えるためのチャンネルである。この情報は２つの目的で使われる。

1）規則（７g）によって，用言を複数個接続することが出来るが，その結合情報として役立てる。

2）用言の命令形は文末詞に反映されることになっているので，その情報を叙述文まで伝えることが必要である。

（意味処理の具体化）

意味情報を伝えるために，３章でしたように新たに１つの変数をそれぞれの述語に追加する。この意味情報を加工するために３章で述べた補強項を使う。次の３つの補強項(24)を使って意味処理を可能にしたプログラムが【付録５】

である。この3つの補強項のプログラムは別に【付録6】で与える。

(24) 1）体言の係受け解析（Ks, SemV, Time, Sem）
2）時間格標識確認（Noun）
3）同一名詞連体修飾（SemH, SemNP, Sem）

「体言の係受け解析」は，格要素リストKs，用言の意味（上記の制約条件の情報）SemV，を使って格の制約条件チェックを行う。この時，深層格としての格は1つの文に1つしかないという「**1文1格原理**」も適用する。このチェックで埋め込み文が複数個連なっても無駄な解析結果を切ることが出来る。また時制情報Timeは格情報の1つとして記録される。

「時間格標識確認」は，名詞の意味情報を受け取ってその特徴が時間の特徴を持っているかを調べる。このチェックによって【付録4】で出力された，時間の格になることが出来ないものまで時間格にするという，無駄を節約できる。

「同一名詞連体修飾」は，1章で説明したような連体修飾を調べるものである。それは，連体修飾する文の意味情報SemHと，修飾を受ける名詞SemNPとを使って，SemNPがSemHの格要素でまだ埋っていないどこかの格要素を満たすことが出来るかどうかを調べる。アルゴリズムは，動詞が持っている格要素のリストを順番に調べる。すでに埋っている格は修飾文の中の格である。従って，空いている格の中で，名詞の特徴が入り込める制限条件を見つければ良い。そのような格が無ければ，現在までの構文解析が間違っていることになる。そうなれば，この解析は放棄し新たな解析を試みればよい。

本来ならデータ構造について説明すべきところであるが，スペースの関係で省略した。これらは【付録5，6】のコメントを参照のこと。

（実行結果）

（25a）の例文の実行結果を（26）に示す。（25b）の結果は【付録7】に

示す。（25a）は1章で述べた同一名詞連体修飾の例である。結果は叙述文の意味を表す動詞と深層格を表示し，これを判断詞で囲んだ。判断が特に示されていなければ non と表示する。名詞，あるいは名詞句はカテゴリーとして特徴を表示をする。また関連情報としてこれらに係る連帯修飾の情報も表示した。関連情報が特になければ non と表示する。また文そのものが引用されている場合には，self と表示した。その場合のカテゴリーは文を表す s である。

(25b)は引用の例であるが，これは2つの解釈が可能な曖昧な文でもある。つまり文頭の「太郎が」が「買う」の主格になったり，「書く」の主格になったりしている。実際の解析結果もそうなった。【付録7】の2つの解析結果の解析速度が5秒と2秒という大きな差があるが，2回目はバックトラックなので，最初の解析よりは速めになっている。なお，これらの結果の表示には【付録8】のプログラムを使った。

(25)　a 「太郎が買った本を花子が読むらしい」
　　b 「太郎が本を買ったと花子に手紙を書いた」

(26)　太郎が買った本を花子が読むらしい　6 sec

【意味解析】

判断詞　　[らしい, [時制, 現在]]

【読む】

【主格】

花子　カテゴリー（hum）　.....花子が読むの主格であることを表示

関連情報＝なし　　.....花子を修飾している情報はない

【対象格】

本　　カテゴリー（con）　.....本が読むの対象格であることを表示

関連情報＝　　.....本を修飾している情報を以下に示す

【埋め込み構造：対象格】　.....本を修飾する埋め込み文である

判断詞　non

【買う】

【主格】

太郎　カテゴリー（hum）...太郎が買うの主格である

関連情報＝なし　.....太郎を修飾する情報はない

【対象格】

＊　.....この対象が同一名詞連体修飾されている

【法情報】

[確言,完了]　.....「買う」ムードは確言

【時制格】

完了　.....「買う」の時制は完了

【法情報】

[確言,現在]　.....「読む」のムードは確言

【時制格】

現在　.....「読む」の時制は現在

（検討，その他）

(25)の例の結果は期待していた以上に良好であったと考えている。また（27）のような名詞に関する関連情報を代入するための変数（Object）を，この名詞クラスのインスタンスと考えてオブジェクト指向を導入することも出来る。そうれば，埋め込み文と主文の間の引用関係をはっきり付けることが出来るだろう。また今後は文をイベントとしてとらえ，このイベントを時間軸あるいは原因と結果の因果関係で管理することが，意味理解の重要な点であることも判明した。意味解析が石綿の結合価だけでは十分でないことは明らかだが，簡便な方法として採用した。これ以上の意味解析は，多くの辞書情報を必要とするので費用と効果を十分考慮する必要がある。

(27)　名詞（［名詞，本］，［本，con，Object］）　-->　［本］.

開発システムはNEC－PC 9801 VMであったが【付録7】に示すように数秒で結果が出ている。最近の高性能パソコンでは1秒以内に収るであろう。開発言語はArity／Prolog ver 5.1である。ラムディスク環境であれば，快適な開発が出来る。ただデバックのツールがボックスモデルのみしかないので，かなり苦しい。

意味処理ルーティンを【付録6】に構文解析のための入出力ルーティンを【付録8】に示した。意味処理については説明が本文で出来なかったが，プログラムのコメントで概要が理解できると思う。意味解析の枠組みは［田中］によっている。今回の開発には，文法は［奥津］［寺村1, 2］，格構造の考え方は［野村2］，解析の枠組みは［田中］，高速解析手法は［松本］［畝見］，意味の制約条件は［石綿］と種々の方々の研究に依存している。係助詞の「ハ」の問題も組込の過程にあるが，単位文の範疇を越えるので今回の開発には完成しなかった。残された課題の大きなものに「の」の付加詞句がある。単位文のトータルな理解システムの完成には，まだまだ遠いというのが実感である。

5. 参考文献

[石綿] 石綿敏雄、荻野孝野：結合価から見た日本文法、「文法と意味Ⅰ」朝倉書店、1983.
[畝見] 畝見達夫：自然言語処理への応用、「Prologとその応用」総研出版、1985.
[奥津] 奥津敬一郎：生成日本文法論、大修館書店、1974.
[柴谷] 柴谷方良ほか：「言語の構造－意味・統語篇」、くろしお出版、1982.
[郡司] 郡司孝男：機械翻訳と言語理論、「機械翻訳」bit別冊、1988.
[田中] 田中穂積：自然言語解析の基礎、産業図書、1989.
[寺村1] 寺村秀夫：日本語のシンタックスと意味Ⅰ、くろしお出版、1982.
[寺村2] 寺村秀夫：日本語のシンタックスと意味Ⅱ、くろしお出版、1984.
[野村1] 野村浩郷：自然言語理解の構造－理解の表現、情報処理、Vol.30, No.10, 1989.
[野村2] 野村浩郷：自然言語処理の基礎技術、電子情報通信学会、1988.
[野村3] 野村浩郷：田中穂積編集、「機械翻訳」bit別冊、共立出版、1988年9月
[松本] Matumoto, Y. et.al., : BUP-A Bottom up Parser Embedded in Prolog, New Generatin Computing, Vol.1, No.2, 1983.
[Fill] Fillmore, C. J., 田中ほか訳：「格文法の原理」、三省堂、1975.

6. 付　　録

【付録1】　奥津文法の確定節文法記述Ⅰ

1	文	--> 叙述文.	a
2	叙述文	--> 平叙文.	b
3	平叙文	--> 核文＋時制.	d
4	核文＋時制	--> 用言句.	e
5	核文＋時制	--> 格文節, 核文＋時制.	e
6	格文節	--> 名詞句, 格助詞.	g
7	格文節	--> 名詞句.	g
8	用言句	--> 用言.	f
9	用言句	--> 核文＋時制, 用言.	f
10	用言	--> 動詞.	
11	名詞句	--> 名詞.	h
12	名詞句	--> 文.	h
13	名詞句	--> 叙述文, 名詞句.	h
14	名詞 --> ［本］.		

```
名詞 --> [手,紙].
名詞 --> [町].
名詞 --> [学,校].
名詞 --> [太,郎].
名詞 --> [花,子].
名詞 --> [朝].
格助詞  --> [を].
格助詞  --> [で].
格助詞  --> [が].
動詞 --> [買う].
動詞 --> [買った].
動詞 --> [読む].
動詞 --> [読んだ].
動詞 --> [書く].
動詞 --> [書いた].
```

【付録２】　畝見氏のＢＵＰプログラムの変更箇所

以下の場所を変更する。

1）３つ目の bup_trans1
　　2行目と3行目の Nt を Nh に

2）４つ目の bup_trans1
　　2行目と3行目の Nt を Nh に

3）５つ目の bup_trans1
　　最後の行 bup_assert_link(Nh, N1) を bup_assert_link(N1, Nh) に

4）parseの２行目 abolish(wf_goal, 4) を abolish(wf_goal/4) に

【付録３】　解析結果を出力する奥津文法の確定節文法記述Ⅱ

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  奥 津 文 法 基 本 部 分  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
s([s, J]) --> 叙述文(J).
```

```
叙述文([叙述文, H, [判断詞, non]]) --> 平叙文(H).
叙述文([叙述文, H, [判断詞, D]])   --> 平叙文(H), 判断詞([判断詞, D]).
平叙文([平叙文, Kernel, Time])     --> 核文＋時制(Kerel, Time).
核文＋時制([核文, [格文節リスト|Ks], V], [時制, T])
                                   --> 核文＋時制２(Ks, V, T).
核文＋時制２([ ], V, T)            --> 用言句([V, T]).
核文＋時制２([K|Ks], V, T)         --> 格文節(K), 核文＋時制２(Ks, V, T).
格文節([格文節, N, [格標識, C]])   --> 名詞句(N), 格助詞([格標識, C]).
格文節([格文節, N, [格標識, φt]]) --> 名詞句(N).
用言句([[用言句, V], T])           --> 用言([V, T]).
用言句([[用言句, [K|V]], T])       --> 核文＋時制([K, _]),
                                       用言([V, T]).
用言([[動詞, V], T])               --> 動詞([V, M, T]).
名詞句([名詞句, N])                --> 名詞(N).
名詞句([名詞句, B])                --> s(B).
名詞句([名詞句, H, N])             --> 叙述文(H), 名詞句(N).

%%%%%%%%%%%%%
%   判 断 詞   %
%%%%%%%%%%%%%
判断詞([判断詞, [H, Time]])        --> はず時制(H, Time).
判断詞([判断詞, [H, Time]])        --> はずがない時制(H, Time).
判断詞([判断詞, [H, Time]])        --> らしい時制(H, Time).
判断詞([判断詞, [H]])              --> かもしれない(H).
判断詞([判断詞, [H]])              --> だろう(H).
はず時制(H,          [時制, 現在]) --> はずだ(H).
はず時制(H,          [時制, 完了]) --> はずだった(H).
はずがない時制(H, [時制, 現在])   --> はずがない(H).
はずがない時制(H, [時制, 完了])   --> はずがなかった(H).
らしい時制(H,        [時制, 現在]) --> らしい(H).
らしい時制(H,        [時制, 完了]) --> らしかった(H).

%%%%%%%%%%%%%
%   名    詞   %
%%%%%%%%%%%%%
名詞([名詞, 本])          --> [本].
名詞([名詞, 手紙])        --> [手, 紙].
```

```
名詞([名詞, 町])          --> [町].
名詞([名詞, 学校])        --> [学,校].
名詞([名詞, 太郎])        --> [太,郎].
名詞([名詞, 花子])        --> [花,子].
名詞([名詞, 朝])          --> [朝].
格助詞([格標識, を])      --> [を].
格助詞([格標識, で])      --> [で].
格助詞([格標識, が])      --> [が].
格助詞([格標識, は])      --> [は].
格助詞([格標識, に])      --> [に].
格助詞([格標識, と])      --> [と].
格助詞([格標識, から])    --> [か,ら].
格助詞([格標識, へ])      --> [へ].

%%%%%%%%%%%%%%
%　判断詞語彙　%
%%%%%%%%%%%%%%
はずだ(はず)                    --> [は,ず,だ].
はずだった(はず)                --> [は,ず,だ,っ,た].
らしい(らしい)                  --> [ら,し,い].
らしかった(らしい)              --> [ら,し,か,っ,た].
かもしれない(かもしれない)      --> [か,も,し,れ,な,い].
だろう(だろう)                  --> [だ,ろ,う].
はずがない(はずがない)          --> [は,ず,が,な,い].
はずがなかった(はずがなかった)  --> [は,ず,が,な,か,っ,た].

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%　　動詞([動詞基本語彙, [ムード, オプション], 時制], 入力リスト, 出力リスト)
%　　ムード: 確言、概言、命令、条件、保留
%　　オプション: non、て、たり　　規定値は non。
%　　時制: 現在、完了
%
%%%%%%%%%%%%
% 動詞Ｉ型
% kak-
動詞([書く, [M, Op], T]) -->　　[書], 活用Ｉ([k, M, T, Op]).
% kaw-
```

```
動詞([買う, [M, Op], T]) -->    [買], 活用Ⅰ([w, M, T, Op]).
% yom-
動詞([読む, [M, Op], T]) -->    [読], 活用Ⅰ([m, M, T, Op]).

%%%%%%%%%%%
% 動詞Ⅱ型
% mi-
動詞([見る, [M, Op], T]) -->    [見], 活用Ⅱ([M, T, Op]).
%%%%%%%%%%%
% 動詞Ⅲ型
動詞([来る, [M, Op], T]) --> [来], 活用Ⅲk([M, T, Op]).
動詞([する, [M, Op], T]) --> 活用Ⅲs([M, T, Op]).
動詞([研究する, [M, Op], T]) --> [研, 究], 活用Ⅲs([M, T, Op]).
動詞([議論する, [M, Op], T]) --> [議, 論], 活用Ⅲs([M, T, Op]).

%%%%%%%%%%%
%  活    用  %
%%%%%%%%%%%
%  活用Ⅰ([語幹最後の子音, ムード, 時制, オプション], 入力, 出力)
活用Ⅰ([A, 確言, 現在, non])    --> 音([A, u]).
活用Ⅰ([A, 概言, 現在, non])    --> 音([A, o]).
活用Ⅰ([A, 命令, 現在, non])    --> 音([A, e]).
活用Ⅰ([A, 条件, 現在, non])    --> 音([A, e]), [ば].
活用Ⅰ([A, 保留, 現在, non])    --> 音([A, i]).
活用Ⅰ([A, 確言, 完了, non])    --> 音便(A), [た].
活用Ⅰ([A, 確言, 完了, non])    --> 音便(A), [だ].
活用Ⅰ([A, 概言, 完了, non])    --> 音便(A), [た,ろ,う].
活用Ⅰ([A, 概言, 完了, non])    --> 音便(A), [だ,ろ,う].
活用Ⅰ([A, 条件, 完了, non])    --> 音便(A), [た,ら].
活用Ⅰ([A, 条件, 完了, non])    --> 音便(A), [だ,ら].
活用Ⅰ([A, 保留, 完了, て])     --> 音便(A), [て].
活用Ⅰ([A, 保留, 完了, て])     --> 音便(A), [で].
活用Ⅰ([A, 保留, 完了, たり])   --> 音便(A), [た,り].
活用Ⅰ([A, 保留, 完了, たり])   --> 音便(A), [だ,り].

活用Ⅱ([確言, 現在, non])  --> [る].
活用Ⅱ([概言, 現在, non])  --> [よ,う].
```

活用Ⅱ([命令, 現在, non]) --> [ろ].
活用Ⅱ([条件, 現在, non]) --> [れ,ば].
活用Ⅱ([保留, 現在, non]) --> [ ].
活用Ⅱ([確言, 完了, non]) --> [た].
活用Ⅱ([概言, 完了, non]) --> [た,ろ,う].
活用Ⅱ([条件, 完了, non]) --> [た,ら].
活用Ⅱ([保留, 完了, て]) --> [て].
活用Ⅱ([保留, 完了, たり]) --> [た,り].

活用Ⅲk([確言, 現在, non]) --> [る].
活用Ⅲk([概言, 現在, non]) --> [よ,う].
活用Ⅲk([命令, 現在, non]) --> [い].
活用Ⅲk([条件, 現在, non]) --> [れ,ば].
活用Ⅲk([保留, 現在, non]) --> [ ].
活用Ⅲk([確言, 完了, non]) --> [た].
活用Ⅲk([概言, 完了, non]) --> [た,ろ, う].
活用Ⅲk([条件, 完了, non]) --> [た,ら].
活用Ⅲk([保留, 完了, て]) --> [て].
活用Ⅲk([保留, 完了, たり]) --> [た,り].

活用Ⅲs([確言, 現在, non]) --> [す,る].
活用Ⅲs([概言, 現在, non]) --> [し,よ,う].
活用Ⅲs([命令, 現在, non]) --> [し,ろ].
活用Ⅲs([条件, 現在, non]) --> [す,れ,ば].
活用Ⅲs([保留, 現在, non]) --> [し].
活用Ⅲs([確言, 完了, non]) --> [し,た].
活用Ⅲs([概言, 完了, non]) --> [し,た,ろ,う].
活用Ⅲs([条件, 完了, non]) --> [し,た,ら].
活用Ⅲs([保留, 完了, て]) --> [し,て].
活用Ⅲs([保留, 完了, たり]) --> [し,た,り].

音便(k) --> [い]. 音便(s) --> [し]. 音便(r) --> [っ]. 音便(w) --> [っ].
音便(t) --> [っ]. 音便(g) --> [い]. 音便(m) --> [ん]. 音便(n) --> [ん].

音([k, i]) --> [き]. 音([s, i]) --> [し]. 音([t, i]) --> [ち].
音([n, i]) --> [に]. 音([h, i]) --> [ひ]. 音([m, i]) --> [み].
音([y, i]) --> [い]. 音([r, i]) --> [り]. 音([w, i]) --> [い].

```
音([k, u]) --> [く].    音([s, u]) --> [す].    音([t, u]) --> [つ].
音([n, u]) --> [ぬ].    音([h, u]) --> [ふ].    音([m, u]) --> [む].
音([y, u]) --> [ゆ].    音([r, u]) --> [る].    音([w, u]) --> [う].
音([k, e]) --> [け].    音([s, e]) --> [せ].    音([t, e]) --> [て].
音([n, e]) --> [ね].    音([h, e]) --> [へ].    音([m, e]) --> [め].
音([y, e]) --> [え].    音([r, e]) --> [れ].    音([w, e]) --> [え].
音([k, o]) --> [こ,う].  音([s, o]) --> [そ,う].  音([t, o]) --> [と,う].
音([n, o]) --> [の,う].  音([h, o]) --> [ほ,う].  音([m, o]) --> [も,う].
音([y, o]) --> [よ,う].  音([r, o]) --> [ろ,う].  音([w, o]) --> [お,う].
% ========================== end of file ==========================
```

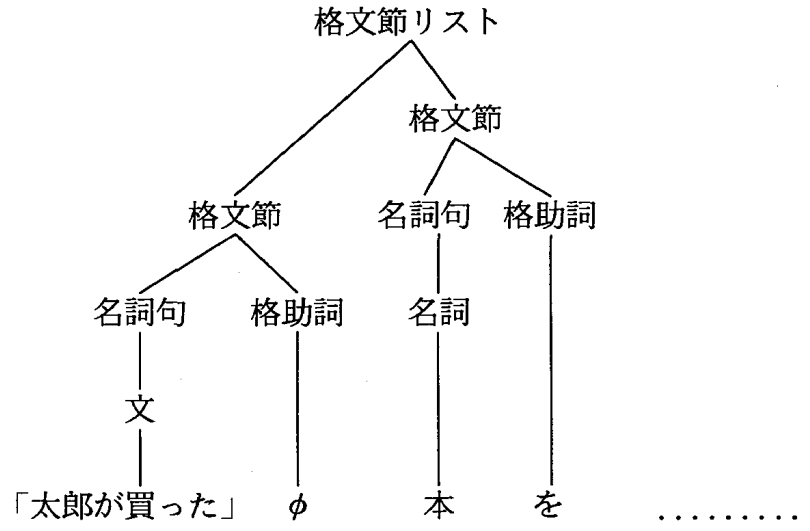
【付録４－１】　「太郎が買った本を花子が読むらしい」の構文解析の結果

```
【構文解析】
  【s】
    【叙述文】
      【平叙文】
        【核文】
          【格文節リスト】
            【格文節】
              【名詞句】
                【叙述文】
                  【平叙文】
                    【核文】
                      【格文節リスト】
                        【格文節】
                          【名詞句】
                            【名詞】
                              太郎
                          【格標識】
                            が
                      【用言句】
                        【動詞】
                          買う
                    【時制】
```

完了
【判断詞】
non
【名詞句】
【名詞】
本
【格標識】
を
【格文節】
【名詞句】
【名詞】
花子
【格標識】
が
【用言句】
【動詞】
読む
【時制】
現在
【判断詞】
【らしい】
【時制】
現在

【付録４－２】　構文情報だけではとれない誤りの構文

【付録３】を用いて「太郎が買った本を花子が読むらしい」を構文解析した結果、３つの誤った構文解析を行った後に、正しい解析をした結果【付録４－１】が得られる。それらの誤りのうち、共通に誤った部分の解析木が以下の木である。この誤りの原因は、空の格助詞φが時間を表す名詞のみに付けられるはずなのに、この意味的制限をする手立てがないためにフリーパスしている所にある。



【付録５】　意味解析を行う奥津文法の確定節文法記述Ⅲ

```
s([s, J], Sem)  -->  叙述文(J, Sem).
叙述文([叙述文, H, [判断詞, non]], [判断詞, non, Sem])
                                    -->  平叙文(H, Sem).
叙述文([叙述文, H, [判断詞, D]],  [判断詞, D, Sem])
                                    -->  平叙文(H, Sem),
                                         判断詞([判断詞, D]).
平叙文([平叙文, Kernel, Time], Sem)    -->  核文＋時制(Kernel, Time, Sem).
核文＋時制([核文, [格文節リスト|Ks], V], [時制, T], Sem)
                                    -->  核文＋時制２(Ks, V, T, SemN, SemV),
                                         {体言の係受け解析(SemN, SemV, T, Sem)}.
核文＋時制２([ ], V, T, [ ], SemV)    -->  用言句([V, T], SemV).
核文＋時制２([K|Ks], V, T, [SemN|SemNs], SemV)
                                    -->  格文節(K, SemN),
                                         核文＋時制２(Ks, V, T, SemNs, SemV).
格文節([格文節, N, [格標識, C]], [C, SemN])
                                    -->  名詞句(N, SemN), 格助詞([格標識, C]).
格文節([格文節, N, [格標識, φt]], [φt, SemN])
                                    -->  名詞句 (N, SemN),
                                         {時間格標識確認(SemN)}.
用言句([[用言句, V], T], [Sem])       -->  用言([V, T], Sem).
用言句([[用言句, [K|V]], T], [SemY|SemK])
```

```
                                        --> 核文＋時制([K, _], SemK),
                                            用言([V, T], SemY).
用言([[動詞,   V], T], Sem)             --> 動詞([V, M, T], Sem).

名詞句([名詞句, N], SemN)               --> 名詞(N, SemN).
名詞句([名詞句, B], [self, s, [[文引用格, Sem]|R] ])
                                        --> s(B, Sem).
名詞句([名詞句, H, N], Sem)             --> 叙述文(H, SemH),
                                            名詞句(N, SemN),
                                            {同一名詞連体修飾(SemH, SemN, Sem)}.
```

＜＜ここに【付録３】の判断詞の１１行を複写する＞＞

```
%%%%%%%%%%%%%%%%%%%%%%%
%        辞 書 項 目        %
%%%%%%%%%%%%%%%%%%%%%%%

名詞([名詞, 本],   [本,   con,   Object]) --> [本].
名詞([名詞, 手紙], [手紙, con,   Object]) --> [手,紙].
名詞([名詞, こと], [事柄, abs,   Object]) --> [こ,と].
名詞([名詞, 町],   [町,   loc,   Object]) --> [町].
名詞([名詞, 学校], [学校, loc,   Object]) --> [学,校].
名詞([名詞, 太郎], [太郎, hum,   Object]) --> [太,郎].
名詞([名詞, 花子], [花子, hum,   Object]) --> [花,子].
名詞([名詞, 朝],   [朝,   temps, Object]) --> [朝].
名詞([名詞, 散歩], [散歩, act,   Object]) --> [散,歩].
名詞([名詞, テニス], [テニス, act, Object]) --> [テ,ニ,ス].
```

＜＜ここに【付録３】の判断詞語彙を複写する＞＞

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%    動詞([動詞基本語彙, [ムード, オプション], 時制], 入力リスト, 出力リスト)
%    ムード:  確言、概言、命令、条件、保留
%    オプション:  non、て、たり        規定値は  non。
%    時制:  現在、完了
%
%%%%%%%%%%%
```

```
% 動詞Ⅰ型
% kak-
動詞([書く, [M, Op], T], [書く, [ [主格,      Act, [が], [hum]],
                                  [対象格,    Ob1, [を], [abs, con]],
                                  [対象2格,   Ob2, [に], [con, hum]],
                                  [場所格,    Loc, [で], [loc]],
                                  [引用格,    Com, [と], [s]],
                                  [法情報,    [M, T], [ ], [ ]]    | R] ])
        --> [書], 活用Ⅰ([k, M, T, Op]).

% kaw-
動詞([買う, [M, Op], T], [買う, [ [主格,      Act, [が], [hum]],
                                  [対象格,    Obj, [を], [con]],
                                  [対象2格,   Ob2, [に], [hum]],
                                  [場所格,    Loc, [で], [loc]],
                                  [対称格,    Syn, [から], [hum]],
                                  [法情報,    [M, T], [ ], [ ]]    | R] ])
        --> [買], 活用Ⅰ([w, M, T, Op]).
% yom-
動詞([読む, [M, Op], T], [読む, [ [主格,      Act, [が], [hum]],
                                  [対象格,    Obj, [を], [con]],
                                  [場所格,    Loc, [で], [loc]],
                                  [共同格,    Syn, [と], [hum]],
                                  [法情報,    [M, T], [ ], [ ]]    |R] ])
        --> [読], 活用Ⅰ([m, M, T, Op]).

%%%%%%%%%%%
% 動詞Ⅱ型
% mi-
動詞([見る, [M, Op], T], [見る, [ [主格,      Act, [が],   [hum]],
                                  [対象格,    Obj, [を],   [con]],
                                  [場所格,    Loc, [で],   [loc]],
                                  [共同格,    Som, [と],   [hum, ani]],
                                  [法情報,    [M, T], [ ], [ ]]  | R]])
        --> [見], 活用Ⅱ([M, T, Op]).
```

```
%%%%%%%%%%%
% 動詞Ⅲ型

動詞([来る, [M, Op], T], [来る, [ [主格,   Act, [が],   [hum, con]],
                                  [出発格, Sta, [から], [loc]],
                                  [目標格, Goa, [へ],   [loc]],
                                  [到着格, End, [に],   [loc]],
                                  [法情報, [M, T], [ ], [ ]]     | R]])
        --> [来], 活用Ⅲk([M, T, Op]).

動詞([する, [M, Op], T], [する, [ [主格,   Act, [が],   [hum]],
                                  [対象格, Verb, [を],   [act]],
                                  [法情報, [M, T], [ ], [ ]]   | R]])
        --> 活用Ⅲs([M, T, Op]).

動詞([研究する, [M, Op], T], [研究する,
                              [ [主格,   Act, [が],   [hum]],
                                [対象格, Obj, [を],   [abs, con]],
                                [法情報, [M, T], [ ], [ ]]   | R]])
        --> [研, 究], 活用Ⅲs([M, T, Op]).

動詞([議論する, [M, Op], T], [議論する,
                              [ [主格,   Act, [が],   [hum]],
                                [対象格, Obj, [を],   [abs] ],
                                [共同格, Com, [と],   [hum]],
                                [法情報, [M, T], [ ], [ ]]   | R) ] )
        --> [議, 論], 活用Ⅲs([M, T, Op]).

<<ここに【付録3】の活用と音便、音の部分を複写する>>

% ========================== end of file ==========================
```

【付録6】　意味処理ルーティン(sem.ari)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  意   味   処   理   シ   ス   テ   ム  %
%                              By. E. Sugimoto     %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  体 言 の 係 受 け 解 析  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                 +    +    +    -
% 体言の係受け解析（Ks, SemV, Time, Sem）
%
%     SemNs = [ [が, [太郎, hum, _]],
%              [を, [本,  con, _]] ]
%
%     SemV  = [ [買う, [ [主格,     Act, [が],     [hum]],
%                     [対象格,   Obj, [を],     [ani, con]],
%                     [対象2格,  Ob2, [に],     [hum, ani]],
%                     [場所格,   Loc, [で],     [loc]],
%                     [対称格,   Syn, [から],   [hum]]  |  R] ]]
%     Time  = 完了
%
%     Sem   = [ [買う, [ [主格,     [太郎, hum, _], [が],    [hum]],
%                     [対象格,   [本,  con, _], [を],    [ani, con]],
%                     [対象2格,  Ob2,          [に],    [hum, ani]],
%                     [場所格,   Loc,          [で],    [loc]],
%                     [対称格,   Syn,          [から],  [hum]],
%                     [時制格,   完了,         [ ],     [ ]] | R] ]]
%

  体言の係受け解析（Ks, [ [Wordv, CList] | SemVn], Time, [Wordv, CList]） :-
            unifyList(Ks, CList),
            appendDList(CList, [時制格, Time, [ ], [ ]]), !.
% SemNs のすべての格要素について，SemVのCListとの対応を調べる


unifyList([ ], _).
unifyList([K|Next], CList) :-
```

```
                unifyL2(K, CList, CList),
                unifyList(Next, CList).

% CListにある格要素の鋳型リストに、Kの格標識を持つ鋳型の有無をそれぞれ調べる。
% 検索は，memberを使って調べる。格標識が一致したら、さらにカテゴリーの対応も
%   チェックする。格標識とカテゴリーが満たされるなら、ＯＫだ。
% 鋳型がなくなったら，格要素をを追加する。
%
unifyL2(E, Var, CList) : -
                var(Var), !,
                格要素の追加(Var, E, CLis t).

unifyL2([_格標識, [Word, Cat, WSem]],
        [[_格機能名, [Word, Cat, WSem], _格標識リスト, CatL] | _], CList) : -
                member(_格標識, _格標識リスト), !,
                member(Cat, CatL).

unifyL2(E, [_ | Next], CList) : -
                unifyL2(E, Next, CList).

% 格要素を動詞オブジェクトに追加する時に、時間格と場所格については
%   それぞれのカテゴリーがそれぞれの格条件を満たすか調べ、さらに、
%   すでにこれらの格機能が満たされていないことを確認する。
%   カテゴリーがこの２つの条件を満たさない時は、追加格という名前で記録する。
格要素の追加([[時間格, [Word, temp, Sem], [ ], [temp]] | _],
                                        [φT, [Word, temp, Sem]], _) : - !.
格要素の追加([[場所格, [Word, loc, Sem], [で], [loc]] | _],
                                        [で, [Word, loc, Sem]], CList) : -
                notmember格機能（場所格, CList).

格要素の追加([[時間格, [Word, temp, Sem], [に], [temp]] | _],
                                        [に, [Word, temp, Sem]], CList) : -
                notmember格機能（時間格, CList).
  格要素の追加([[追加格, [Word, C, Sem], [K], [C]], [K, [Word, C, Sem]] | _],
                                        CList) : -
                notmember格機能（場所格, CList).
```

```
notmember格機能(K, Var) : - var(Var), !.
notmember格機能(K, [ [K|_] | _]) : - !, fail.
notmember格機能(K, [ _ | Next ]) : -
                notmember格機能(K, Next).

%%%%%%%%%%%%%%%%%%%%%%%%%%
%    時 間 格 標 識 確 認    %
%%%%%%%%%%%%%%%%%%%%%%%%%%
時間格標識確認([Noun, temps, _]).

%%%%%%%%%%%%%%%%%%%%%%%%%%
%      同一名詞連体修飾      %
%%%%%%%%%%%%%%%%%%%%%%%%%%
%                        +     +     -
%    同一名詞連体修飾(SemH, SemNP, Sem).
%
%    SemH  :  [判断詞, D, _核文意味]
%    SemNP :  [Word, Cat, _名詞句の修飾リスト]
%    Sem   :  [Word, Cat, NewSemList]
%    NewSemList  :  (太郎が買った本)の意味の例
%            [本, con, [埋め込み構造: 対象格, [判断詞, non,
%                    [買う, [ [主格,  [太郎, hum, _], [が], [hum]]
%                             [対象格, *,      [を], [con]]
%                             [時制格,   完了, [ ],      [ ]]
%                             |  R]
%                        ]]]]
%
%    同一名詞連体修飾(SemH, SemNP, Sem)のアルゴリズム
%
%        SemH =  [判断詞, non,
%                [買う, [ [主格,       [太郎, hum, _], [が], [hum]]
%                         [対象格,      _,             [を], [con, ani]]
%                         [対象２格,    _,             [に], [hum, ani]]
%                         [場所格,      _,             [で] , [loc]]
%                         [対称格,      _,             [から], [hum]] |  R]]  ]
%
%        SemN =  [本, con, [S1, S2]]
```

```
%                    S1, S2 は、既に本に付けられていた修飾意味
%
%      だから、SemN のカテゴリー con を取りだし、格機能の条件カテゴリーリスト
%      の中で、con を持つものを上から順番に調べる。一致するものがあれば、この
%      格機能名を取りだし、格機能のオブジェクトを * とする。
%      取りだされた格機能名は、「埋め込み構造」とで
%
%            埋め込み構造 : 対象格
%
%      を作る。したがって、以下の様になる。
%
%       [本, con, [ [埋め込み構造:対象格,
%                    [判断詞, non,
%                     [買う, [ [主格,      [太郎, hum, _], [が], [hum]]
%                              [対象格,     *,             [を], [con, ani]]
%                              [対象2格,    _,             [に], [hum, ani]]
%                              [場所格,     _,             [で], [loc]]
%                              [対称格,     _,             [から], [hum]]
%                              | R]
%                    ]]],
%                  S1, S2 ] ]
%

同一名詞連体修飾([判断詞, D, [Verb, CList]], [N, C, L], [N, C, [Sem | L]]) :-
        格機能探索([N, C], _格機能, CList),
        Sem = [埋め込み構造 : _格機能, [判断詞, D, [Verb, CList]]], !.

格機能探索(_, _, X) :- var(X), !, fail.
格機能探索([N, Cat], _格機能, [[ _格機能, *, Ks, Cats] | _]) :-
        member(Cat, Cats), !.
格機能探索 (NS, _格機能, [_ | Next]) :-
        格機能探索(NS, _格機能, Next).

% ------------------------------
appendDList(X, Y) :-
        var(X), !,
        X = [Y | _].
```

```
appendDList([A | X], Y)   : -
        appendDList(X, Y).

member(X, [X | _])   : -   !.
member(X, [_ | Y])   : -   member(X, Y).
% ========================== end of file ==========================
```

【付録７】　意味解析による意味構造

（２６ｂ）「太郎が本を買ったと花子に手紙を書いた」の例

(1)　太郎が本を買ったと花子に手紙を書いた　　５ｓｅｃ

【意味解析】

判断詞　non

【書く】

【主格】　　.....「太郎が手紙を花子に書いた」

太郎　カテゴリー(hum)

関連情報＝なし

【対象格】

手紙　カテゴリー(con)

関連情報＝なし

【対象２格】

花子　カテゴリー(hum)

関連情報＝なし

【引用格】　　.....何といって書いたかというと

self　カテゴリー(ｓ)

関連情報＝

【文引用格】

判断詞　non

【買う】　　.....「本を買った」と

【対象格】

本　カテゴリー(con)

関連情報＝なし

【法情報】

[確言,完了]

【時制格】
完了
【法情報】
[確言,完了]
【時制格】
完了

(2)　太郎が本を買ったと花子に手紙を書いた　　2sec
【意味解析】
判断詞　non
【書く】
【対象格】　　.....「誰かが手紙を花子に書いた」
手紙　カテゴリー(con)　　... 主格「誰」は不明である。
関連情報＝なし
【対象2格】
花子　カテゴリー(hum)
関連情報＝なし
【引用格】
self　カテゴリー(s)
関連情報＝
【文引用格】　　.....何といって書いたかというと
判断詞　non
【買う】
【主格】
太郎　カテゴリー(hum)　.....「太郎が本を買った」と
関連情報＝なし
【対象格】
本　カテゴリー(con)
関連情報＝なし
【法情報】
[確言,完了]
【時制格】
完了
【法情報】
[確言,完了]
【時制格】
完了

【付録８】 構文解析のための入出力ルーティン（print. ari）

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                             %
%   構 文 解 析 入 出 力 ル ー テ ィ ン       %
%                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

go(DCGFile) :- [ubup], [sem], bup(DCGFile). %ファイル名ubupはBUPシステムのファイル名
                                  % semファイル名は付録6のファイル名 sem. ari
% 最初にこのファイルをprologにコンサルトして、次の実行を行う。
% 実行は ?- go('DCGで書かれたファイル名').
%        ?- start.
% あとは、指示にしたがってデータ文を入力すると、解析結果が表示され、
% また一方bup. outファイルに表示されたものと同じ内容が出力される。
%
start  :- create(H, 'bup. out'), s0(H).
open(F)  :- create(H, F), s0(H).
s0(H)  :- abolish(handle/1),
           asserta(handle(H)),
           date(date(Y, M, D)),
           tab(H, 29), write(H, '構 文 解 析 結 果     '),
           write(H, Y), write(H, '年'),
           write(H, M), write(H, '月'),
           write(H, D), write(H, '日'),
           tab(H, 2),
           printTime, nl(H), nl(H).
close  :- handle(H), close(H).
end    :- close, halt.

read   :- abolish(ans/3),
          num_cls,
          タイトル出力,
          read(Data),
          入力文字の処理(Data, ListOfChar),
          解析と結果出力(Data, ListOfChar).
```

```
タイトル出力　:-
        write('*****日本語解析のデータ文を入力して下さい*****'), nl,
        write('**　　データは、　$　と　$　で囲んで入力して下さい。**'), nl,
        write('**　そして最後にピリオッド．と改行を入力します。　**'), nl, nl.

%%%%%%%%%%%%%%%%%%%%
%　　入力文字の処理　　%
%%%%%%%%%%%%%%%%%%%%
入力文字の処理(Data, ListOfChar)　:-
          tr1(Data, Y),
          tr2(Y, ListOfChar), !.

%tr1(+文字列, -文字のリスト).
%　目的：文字列をバラしてリストにする
tr1(X, Y)　:-
        string_length_k(X, N),
        tr11(X, N, 0, Y).
%tr11(+文字列, +文字の個数, +処理対象の文字位置, -文字のリスト)
tr11(X, L, L, [])　:-　!.
tr11(X, N, L, [C | Y])　:-
            nth_char_k (L, X, C), L2 is L + 1, tr11(X, N, L2, Y).

%tr2(+文字のアスキーコードのリスト, -文字のリスト).
%　　目的：漢字コードを漢字に変換する
tr2([], [])　:-　!.
tr2([Car | Cdr], [Atom | AList])　:-　name_k(Atom, [Car]), tr2(Cdr, AList).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%　　解　析　と　結　果　出　力　　%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%　　入力データ：　ListOfChar
%　　解析結果　：　Ans
%
%　　解析とその結果は、入力されたデータに対して、可能性のあるものを全て試し
%　　それらを表示する。
%　　従って、ここで全ての組合わせをテストする為に、fail がある。
```

```
%     しかし、「解析と結果出力」の実行結果は、必ず成功で終る。
%
解析と結果出力(Data, InData)  :-
        abolish(timef/1), time(ST), assert(timef(ST)),
        parse(s, Ans, InData),        %%% <<ＢＵＰシステムの呼出
        num(Num),
        time(Time), retract(timef(SetTime)), assert(timef(Time)),
        assertz(ans(Num, Ans, [SetTime, Time])),
        fail.
解析と結果出力(Data, _)  :-
        ans(Num, Ans, Times),
        prints([Num, Data, Times], 0, Ans),
        fail.
解析と結果出力(Data, _)  :-
        file_title,
        ans(Num, Ans, Times),
        printf([Num, Data, Times], 0, Ans),
        fail.
解析と結果出力(_, _).

prints([Num, Data, Times], N, [Syntax, Semantics])  :-
        write(' (' ), write(Num), write(') '),
        write(Data),
        経過時間(Times, T),
        tab(2), write(T), write('sec'), nl, nl,
        print【 】 (0, 構文解析), nl,
        print 構文構造(0, N, [Syntax]), nl, nl,
        print【 】(0, 構文解析), nl,
        print 意味構造 (0, N, Semantics), nl, nl, !.

printf([Num, Data, Times], N, [Syntax, Semantics])  :-
        handle(H),
        write(H, '(' ), write(H, Num), write(H, ') ' ),
        write(H, Data),
        経過時間(Times, T),
        tab(H, 2), write(H, T), write(H, 'sec'), nl(H), nl(H),
        print【 】(H, 構文解析), nl(H),
```

```
        print構文構造(H, N, [Syntax]), nl(H), nl(H),
        print【 】(H, 意味解析), nl(H),
        print意味構造 (H, N, Semantics), nl(H), nl(H), !.

%%%%%%%%%%%%%%%%%%
%      print構文構造      %
%%%%%%%%%%%%%%%%%%
print構文構造(H, N, X)  :-
            var(X), !,
            tab(H, N),
            write(H, '?'),
            nl(H).
print構文構造(H, _, [ ])  :-  !.
print構文構造(H, N, X)  :-
            atom(X), !,
            tab(H, N),
            write(H, X),
            nl(H).
print構文構造(H, N, [X | Y])  :-
            var(X), !,
            tab(H, N), print【 】(H, '?'), nl(H),
            M is N+3,
            printn(H, M, Y).
print構文構造(H, N, [X | Y])  :-
            atom(X), !,
            tab(H, N), print【 】(H, X), nl(H),
            M is N+3,
            printn(H, M, Y).
print構文構造(H, N, [X | Y])  :-
            M is N+3,
            print構文構造(H, M, X), print構文構造(H, M, Y).

printn(H, _, [ ]).
printn(H, N, [X | Y])  :-
            print構文構造(H, N, X), printn(H, N, Y).
```

```
%%%%%%%%%%%%%%%%%%%%
%        print 意味構造        %
%%%%%%%%%%%%%%%%%%%%
print 意味構造(H, N, [判断詞, D, S])  :-
                tab(H, N), write(H, '判断詞'), tab(H, 2), write(H, D), nl(H),
                M is N + 3,
                print 平叙文意味(H, M, S).
print 平叙文意味(H, N, [Verb, CaseList])  :-
                tab(H, N), print【】(H, Verb), nl(H),
                M is N + 3,
                print 格List(H, M, CaseList).

print 格List(_, _, Var)  :- var(Var), !.
print 格List(H, N, [[KF, Object, KL, Cats] | Next 格List])  :-
                var(Object), !,
                print 格List(H, N, Next 格List).
print 格List(H, N, [[KF, Object, KL, Cats] | Next 格List])  :-
                tab(H, N), print【】(H, KF), nl(H),
                M is N + 3,
                printObject(H, M, Object),
                print 格List(H, N, Next 格List).

printObject(H, N, T)  :-
                atom(T), !,
                tab(H, N), write(H, T), nl(H).
printObject(H, N, [Name, Cat, Sem])  :-
                !,
                tab(H, N), write(H, Name), tab(H, 2),
                write(H, 'カテゴリー( '), write (H, Cat), write(H, ')'), nl(H),
                tab(H, N),
                (var(Sem), !,
                        write(H, '関連情報＝なし'), nl(H)
                  ;
                        write(H, '関連情報='), nl(H),
                        print 関連情報(H, N, Sem) ).
printObject(H, N, T)  :-
                tab(H, N), write(H, T), nl(H).
```

```
print関連情報(_, _, Var)  :- var(Var), !.
print関連情報(H, N, Atom)  :-
                atom(Atom),
                tab(H, N), write(H, Atom), nl(H).
print関連情報(H, N, [E|Next])  :-
                printSemElement(H, N, E),
                print関連情報(H, N, Next).

printSemElement(H, N, [K, _意味])  :-
                tab(H, N), print【】(H, K), nl(H),
                M is N + 3,
                print意味構造(H, M, __意味).

print【】(H, X)  :-
                write (H, '【'), write(H, X), write(H, '】').

%--------------utilities --------------------
num(N)  :- retract(numdata(N)), !, M is N+1, assert(numdata(M)).
num(1)  :- assert(numdata(2)).
num_cls  :- abolish(numdata/1).
printTime  :-  time(time(H, M, S, _)),
                handle(F),
                write(F, H), write(F, ' - '),
                write(F, M), write(F, ' -'),
                write(F, S).
file_title  :- handle(F),
                tab(F, 67), printTime, nl(F).
経過時間([From, To], T)  :-
                From  =..[time, H1, M1, S1|_],
                To    =..[time, H2, M2, S2|_],
                X is 360*H1+60*M1+S1,
                Y is 360*H2+60*M2+S2,
                T is Y - X.
%=======================end of file ============================
```