自然言語処理 67－1
（1988. 7. 22）

素性構造の単一化過程の視覚化手法

逸見 研一 小暮 潔
㈱ATR自動翻訳電話研究所

辞書記述の開発は、処理の試行によりその妥当性の検証を行いながら進められる。しかしこの方法によれば、満足な結果が得られないことで、関係する記述のどこかに問題があることが判るだけである。したがって、問題の解決のためには、丁度プログラムをデバッグするときのように、実行時のトレース情報などから、問題点を辞書記述のレベルまで還元する操作が必要となる。本稿では、このように複雑な素性構造の単一化過程の挙動を把握追跡する際に、素性構造をいかに提示すべきかについて論じる。

How to Represent Feature Structure Unification Process

Ken-ichi ITSUMI Kiyoshi KOGURE
ATR Interpreting Telephony Research Laboratories
Twin 21 Bldg. MID Tower 2-1-61 Shiromi Higashi-ku Osaka 540 Japan

New method to determine spacial layout of labeled directed acyclic graph is developed. This paper describes outline of this method. Development of lexical descriptions are made out together with testing validity of temselves by practical using. But, such test tell us only whether testing descriptions are O.K. or not. If some parts were wrong, one should try to find out what are wrong and how to correct them like debugg of program from trace informations. This paper describes, in order to such debugg of lexical descriptions, what informations should be represented and how to present them.

1. はじめに

機械翻訳システムなどの自然言語処理システムに必要な言語情報、すなわち、一般的な文法情報や語彙情報などは、一般に、その妥当性の検証を行いながら蓄積･整備される。したがって、この過程を効率的に行うことが、このようなシステムを構築する上での重要な要素となる。

ある語彙に関する記述の妥当性を評価しようとする場合、最も直接的な方法は、実際にその記述を用いて、目的とする処理を試行してみることである。この手段によれば、語彙記述に問題があった場合に、満足な結果が得られないことによって、その処理に関係した辞書記述のどこかに不適切な記述が存在するということが判るだけである。だから、対処法を見出すために、ちょうどプログラムのデバッグと同じ様に、問題点を、トレース情報などから、語彙記述のレベルにまで還元することが必要となってくる。

本稿では、ATRで開発されている端末間対話翻訳システムNADINEにおける解析用辞書の開発作業を対象として、その効率化に関して議論するとともに、そこで用いられている素性構造を視覚化する手法について述べる。

2. 単一化に基づく発話の解析

2.1 NADINEの単一化パーサー

ATRで開発している端末間対話翻訳システムNADINEは、自動翻訳電話のプロトタイプとしての、端末間でのキーボードを介しての日英二ヵ国語間の対話のための機械翻訳システムである。

NADINEの日本語解析部は、入力として、解析の対象である文字列と、単一化文法の枠組みにしたがって記述された文法規則及び語彙記述をとり、Earleyのアルゴリズムにしたがって解析を逐行し、統語的･意味的に受理された構造各々の素性構造表現の集合を返す。文法的な枠組みは、語彙中心的で、Head-Driven Phrase Structure Grammar [HPSG] の枠組みに沿っている。そして、Japanese Phrase Structure Grammar [JPSG] の主要な素性を採用し、これに加えて、話し言葉を記述するために、文の階層性を示す素性や語用論的な素性などを新たに導入している。

2.2 単一化文法の利点

単一化文法の枠組みを用いることの利点として、次の2つが挙げられる。

(1) ある語の担う統語論的･意味論的･語用論的情報などの多面的な情報を、素性構造の形式を用いて統一的かつ部分的に記述することができる。

(2) 自然言語の解析、変換、生成、などの多様な処理を、素性構造間の単一化演算という論理的に明解な形式で構成することができる。

こうした利点から、単一化文法の枠組みにしたがうことにより、意味の構成的な解析など高度で複雑な処理を比較的見通しよく行うことが期待できる。

2.3 単一化文法の枠組みでの語彙記述

単一化文法の枠組みでは、文や名詞句などの句構造、さらにそれを構成する各単語の担う情報は、いわゆる素性の束である素性構造で表現される。NADINEの解析用辞書における記法にしたがった「情報」という語彙項目の記述例を[Fig.1] に、示す。

```
(deflex 情報 N
  [[head [[pos n]]]
   [subcat { }]
   [slash { }]
   [semf [[anim -]]]
   [sem [[parm ?x]
         [restr [[reln 情報-1]
                 [obje ?x]]]]]])
```

[Fig.1 語彙項目に関する記述の例]

2.4 単一化文法の枠組みでの文法規則の表現

単一化文法の枠組みでは、文法規則は、規則左辺の記号の構成を表現するCFG規則と、そのCFG規則両辺の記号間に満たされるべき制約によって表される。NADINEの解析用文法の記法にしたがった、名詞と後置詞の連結から後置詞句を構成する文法規則の例を[Fig.2] に示す。

この記述において、第一行defruleの後に続くのがCFG規則で、この後に続く素性構造に関する等式によって、規則が適用される際にCFG規則両辺の記号間に満たされるべき制約が記述される。パス表現間の等式によって、素性構造中の値が指定される。パス表現は、句構造を示す非負の整数とそれに続く素性名の列により構成される。非負の整数は、0ならば規則の左辺の親構造に対応する素性構造、自然数nの時は、規則右辺のn番目の娘構造に対応する素性構造を示し、それに続く素性名の列は、その素性構造の根から、参照する素性値に至るまでにたどられる素性名を示す。そして、等式により

二つの素性がトークンとして同一であることが示される。

例えば、

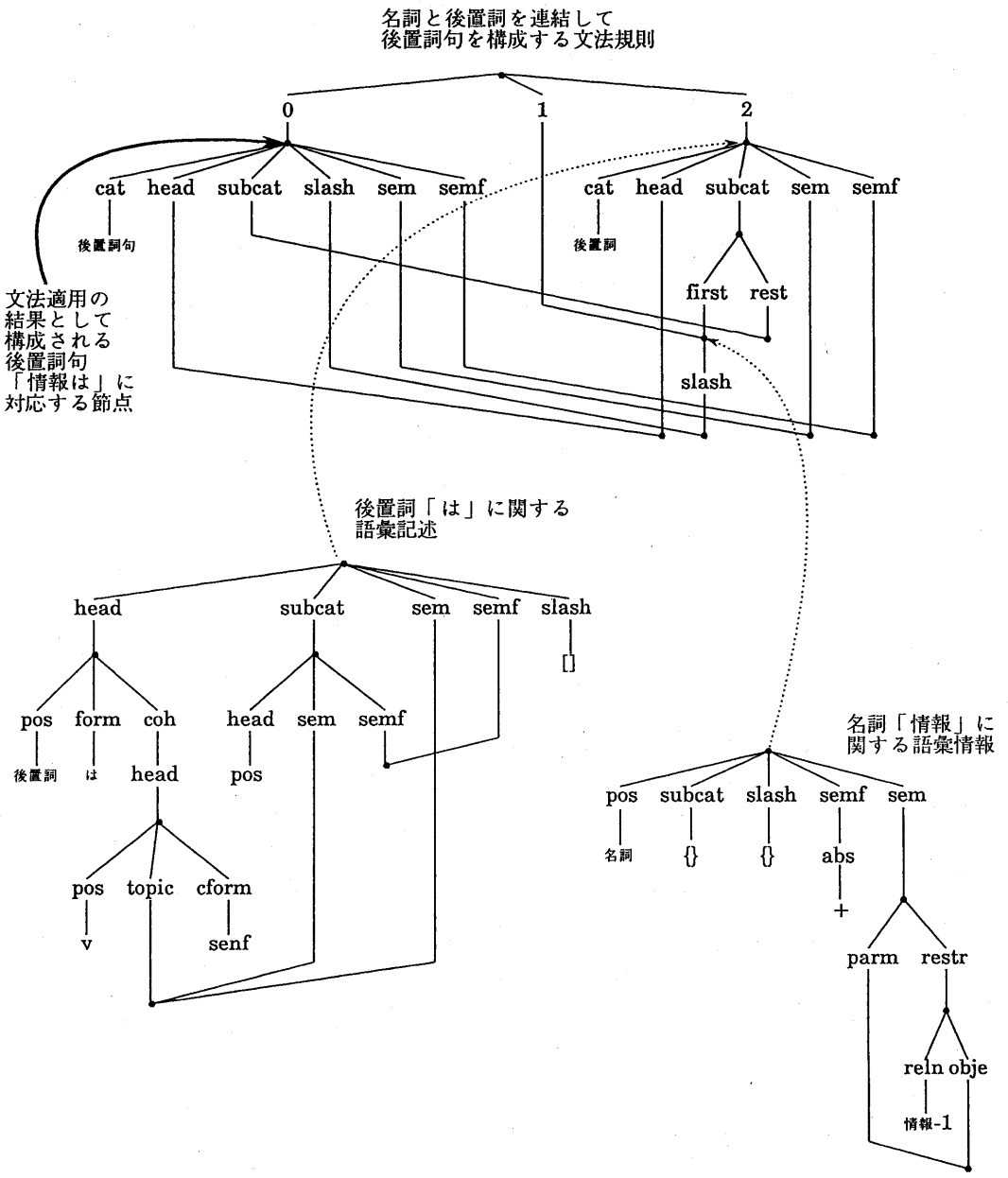
(<0 head> == <2 head>)

は、このCFG規則によって後置詞句が構成されるとき、でき上がる後置詞句のhead素性の値は、後置詞のhead素性の値と等しくなければならないという制約を表している。

```
(defrule 後置詞句 ⇒ (名詞 後置詞)
  (<0 head>    == <2 head>)
  (<0 subcat>  == <2 subcat rest>)
  (<0 slash>   == <1 slash>)
  (<1>         == <2 subcat first>)
  (<0 sem>     == <2 sem>)
  (<0 semf>    == <2 semf>))
```

[Fig.2 文法記述の例]

第1行が後置詞句→名詞　後置詞なるCFG規則を表し、後に続く等式の列は、規則が適用されるとき、規則両辺の記号間に満たされるべき制約を表す。



[Fig.3 語彙中心の単一化文法の枠組みでの文法規則の適用]

こうした文法適用時の制約もまた一つの素性構造で表現することができる。システム内部では、[Fig.2]にあげたような文法規則は、[Fig.3]の上部に示したような同値な素性構造に変換して取り扱われる。

2.5 文法規則の適用

単一化文法の枠組みの上では、文法規則の適用は、規則右辺の記号に対応する素性の値それぞれに対して、対応する素性構造を単一化していくことにより逐行される。このとき単一化の順番は任意でよい。もし、途中で、こうした単一化に失敗したならば、その文法規則の適用は失敗ということになる。規則右辺のすべての記号に対応する素性の値に対して、この単一化の操作が無事終了すれば、この文法適用は成功し、このときの、規則右辺の記号に対応する素性の値が、文法適用の結果構成された記号に対応する素性構造となる。例として、[Fig.1]に挙げたの名詞「情報」と後置詞「は」から、[Fig.2]の文法規則にしたがう連結によって、後置詞句がつくられる例を[Fig.3]に示す。

3. 解析用辞書の記述

3.1解析用辞書記述過程

現在、我々のNADINEシステムでは、解析用の辞書記述を整備する際、記述する語彙項目を含む典型的な文の集合–想定入力文を用意し、[Fig.4]に示したような手順を踏んで、解析用辞書の開発作業を行っている。

(1) 語彙項目の初期記述

文法的な諸原則に基づき、類縁の語の語彙記述などを参考にして、語彙項目に関する情報を素性構造により記述する。この際、テンプレートなどを使用して記述を容易にしている。

(2) 句構造の決定

文法の原則にしたがって、想定入力文の句構造を決定する。

(3) 解析結果の素性構造の初期想定

想定入力文と、選んだ句構造から、解析結果として望ましい素性構造を想定する。ここでは、特に、変換以降に引き渡される意味記述の整合性に重点が置かれる。

(4) 構文解析の試行

当面の語彙記述の妥当性を検証するために、その記述を用いて、想定入力文の解析を実際に行ってみる。

解析が途中で中座して失敗するならば、その原因を究明し、それを語彙記述のレベルまで還元して適当な修正を加える。このために、想定入力文の再度の解析、一部の解析などを行う。このようなデバッグ時の解析に際しては、規則によって単一化される素性構造とその結果のトレース情報などが用いられる。

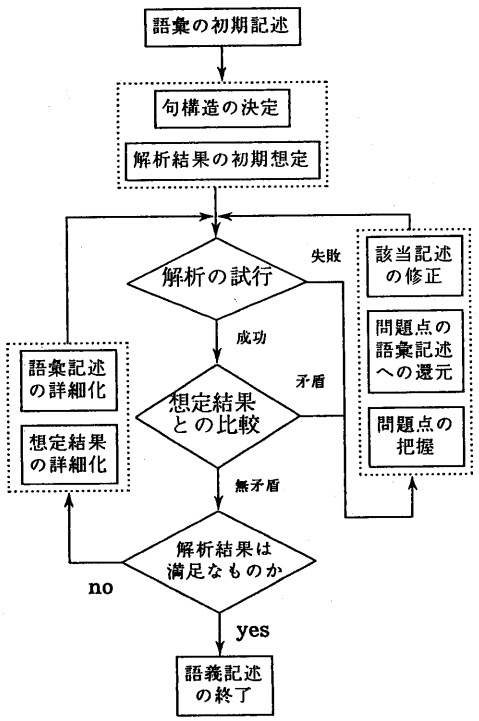
(5) 試行解析結果と想定解析結果との比較

試行解析結果と想定解析結果を比較する。解析結果として予想された素性構造が得られない場合には、解析が中座する場合と同様にして、それを語彙記述のレベルまで還元して適当な修正を加える。

3.2語彙記述作業の問題点

以上で述べた辞書記述の過程は、解析の試行による記述の妥当性の検証を中心にして進められている。

ところで、語彙記述作業における処理の試行は、どちらかというと、辞書記述の十分性を確認するための手段としてよりはむしろ、それぞれの語彙記述に関してそれが妥当であるための条件の探索の道具として用いられている。だが、解析の試行によっては、語彙記述に何か不備があるときに、構文解析の失敗や、得られた解析結果の異常か



[Fig.4 語義記述開発作業の流れ]

ら、問題点の存在を知ることができるだけであり、問題解決のための情報、つまり、語彙記述のどこに問題があって、それをどう直せばよいか、という肝心なことが、この方法では直接示されない。そこで、プログラムをデバッグするときのように問題点を特定の語彙記述に還元する作業が必要になってくる。

ここに挙げた問題はすべて、たかだか語彙記述の十分性を確認する機能しか有していない、処理の試行という手段でもって、語彙記述に関する必要条件の探索を行うことの無理に起因するものと思われる。だから、処理対象文を構成する各単語について、その語彙記述に関する必要条件を明らかにする道具を用意することによって、このような問題を解決することができるものと思われる。

これとは別に(1)、(3)、(5)のプロセスでは、素性構造の比較が、大きな役割を果たしている。たとえば、想定された素性構造と解析で得られた素性構造の比較であり、あるいは、2つの素性構造の単一化可能性の判断である。素性構造を、その構造の相違が一目瞭然と成るように図式化することによって、この種の作業の効率を、改善することが期待できる。

4. 素性構造の視覚化技術

素性構造は有向グラフの構造をとっている。そして、上述のような目的には、グラフ構造として、図式化したほうが有利である。グラフとして図式化するためには、その構成要素であるノードやアークの空間的配置を決定しなければならない。以下では、素性構造をいかに図式化するかについて議論する。

4.1図化表現の形式

描画自体について論じる前に、対象を、いかなる幾何学的形式にしたがって配置するのかについて論じておく。構造を『見易く』表示するためには、構造のメンタルイメージに適合した幾何学的形式を、うまく選択する必要がある。

木構造の表示の際の表現形式として、多くの場合、樹状図が選ばれる。樹状図は、木構造のメンタルイメージそのものの形をしているうえ、幾何学的形式が明瞭な図形であり、木構造の表現形式として理想的なものであるといえる。しかし、素性構造については、木構造に対する樹状図のような、直接的かつ良好な表現の形式は見いだされていない。

そこで、素性構造に対して、その節点すべてと、それらを結ぶ最少の辺で構成される木構造を考える。これをもとの素性構造の極大木と呼ぶ。ここでは、この素性構造の極大木に不足の辺を補ったものを、素性構造の図化表現形式とすることにする

4.2幾何学的観点からの『見易さ』

まず、『見易さ』を特徴ずける要因について述べる。樹状図について、これを『見易く』表示するための要件として、次のようなものが挙げられる。

(1)木の構造の顕示に関するもの

(1.1)階層性の顕示

木構造は階層化されているべきである。すなわち、すべてのノードは階層ごとに分離して、階層ごとに平行線上に配置されるべきである。

(1.2)縁戚関係の顕示

節点間の縁戚関係を明示するために、親ノードは、子ノードの重心線上に来るように配置されるべきである。また。子ノード同士は、同世代のそうでないノードに比べて、近接して配置されるべきである。

(1.3)出力デバイスの有限性への配慮

出力デバイスの大きさや解像度には限界があるので、『見易さ』を損なわない限り、できるだけ詰めて配置できるよう調整すべきである。

(1.4)部分構造の占める領域の顕示

表象のみならず各部分構造の占める領域のメンタルイメージも交わらない様にすべきである。

(2)二節点間の経路の顕示に関する要因

(2.1)線交差

アークは、線交差が最小であるように引かれるべきである。さらに、隣接する平行なアークは、適度に分離されるべきである。

(2.2)アークの直線性

アークは可能な限り直線に引かれるべきである。

(3)構造の比較のしやすさに関するもの

(3.1)相同性の顕示

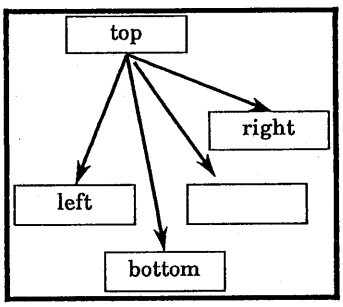
相同な構造[部分構造]は同じ図式として描かれるべきである。また相違な構造は、構造の相違が一目瞭然となるよう描かれるべきである。

4.3部分構造を表示するのに必要な領域の評価

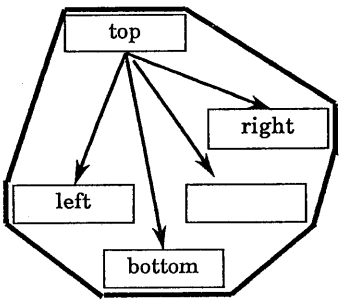
各部分構造について、その空間的配置を決定するためには、その構造の占める領域を的確に評価する必要がある。

人間には、グラフなどの概形を、円や長方形などの規範的な図形の組合せ、あるいは、最小被覆凸領域として認識する傾向のあることが、いくつかの観察から推測できる。このことから、グラフの占める領域のメンタルイメージとは[Fig.5]の①や②に示した様なものであると考えられる。②は、常に、①に包含された、①より小さな図形となる。だから、②に基ずいて空間配置を行ったほうが、グラフを詰めて描くことができる。したがって、出力デバイスの大きさが限られていることを考慮すると、空間配置の結果に関しては、②のほうが①より優れた方法であるということができる。

① 外接長方形



②最小被覆凸領域



[Fig.5 部分木の概形のメンタルイメージ]
(太枠で示してある)

だが、対話型システムのインターフェースとしての利用を考えたときには、何よりも高速性が重要であり、いかに美しい空間配置が得られようとも、あまり計算量の多い方法には魅力がない。

あとで詳述するが、空間的配置の決定のための主要な手続きとして、こうした概形を求める手続きと、複数の概形から、それらすべてを含む概形を求める手続きとがある。グラフの概形の選び方によって、こうした手続きに要する計算量は大きく違ってくる。Fig.6-3の①に挙げた長方形の概形を用いる方法はこうした計算が容易であるために、広く用いられている。一方で、②の概形を用いる方法は、空間配置結果の良さにもかかわらず、この計算が著しく高価であるために用いられていない。

概形を最小外接凸多角形で表現する方法の問題点の一つは、図形の最小外接凸多角形の頂点を求める計算が高くつくことである。ところで、すべての頂点を求めるためには、$O(n^2)$の計算が必要となるが、たとえば最も上の点のような、常に最小外接凸多角形の頂点となることが判っている特別な点を求めるだけならば$O(n)$の計算で済ませることができる。図形の最も上の点、最も下の点、最も右の点、最も左の点を結んだものを[Fig. 6]に示す。このような乱暴な近似で得られる概形は、極めて容易に求めることができ、かつ、きわめて少数のパラメーターで表現することができるから、あとの取扱いも容易となるが、残念ながら、もとの図形を常に内包しているとは限らない。そこで、このようなラフな概形を空間配置に使うためには、次のような補正が必要となってくる。

> ①概形の四つの辺それぞれについて、最も著しい修整を要するはみだしを調べる。
> この操作は$O(n)$のオーダーの計算で済ませることができる。
>
> ②概形の四つの辺それぞれを、最も著しいはみだしに対して補正する。
> この操作に必要な計算量は、nによらない。

こうした操作によって、$O(n)$のオーダーの計算量で、最小外接凸多角形に近い図形を得ることができる。

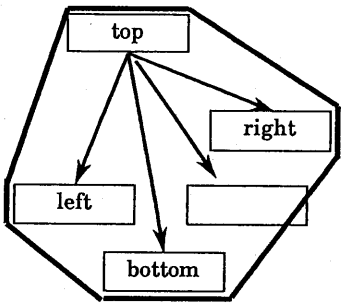
4.4部分構造の配置の決定

[4.2節]で述べた『見易さ』を生み出す基準を守りつつ、素性構造の図式化表現を行う手順の概要は次のようになる。

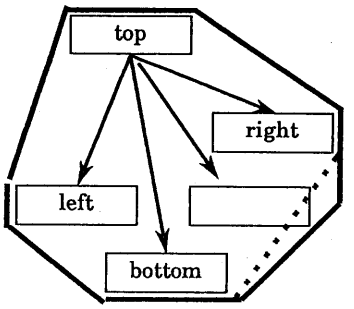
(4.3.1)素性構造からその図化表現形式を求める

素性構造は、素性名をアーク、素性値をノードとするDAGで表現される。このようなDAGの極大木の一つを、その図化表現形式として選ぶ。

③特別の点を結んで得られる図形



④③を補正して得られる図形



[Fig.6 ラフな概形の補正]
(太枠で部分木の概形を囲んである)

(4.3.2)素性構造の図化表現形式を階層化する

根からノードまでの経路上のアークの数の最大値を、そのノードの属する階層と定めることによって、グラフを構成する任意のノードについて、一意に、その所属する階層を定めることができる。

この操作によって、素性構造の図化表現形式を階層グラフに還元することができる。こうして決定した階層は、もとの素性構造において、何か意味があるわけではないが、『見易さ』のための基準(1.1)を満たすために行われる。

(4.3.3)素性構造の図化表現形式を標準化

同一の構造のグラフに対しては常に同一の図形が得られないならば、複数のグラフを比較検討するときなどに不都合となる。

一般のグラフの同形判定は、効率的なアルゴリズムが存在しないとされている。しかし、素性構造に由来するDAGは、すべてのノードについて、そこから発するアークに、それぞれユニークなラベルがついているから、このことを利用して、すべてのノードについて、そこから発するアークを辞書式順序などの一定の秩序に配列することによって、容易に、表示の一意性を実現することができる。この操作を構造の標準化と呼ぶ。

(4.3.4)アークの線交差を最小化

多階層グラフのリンクの交差を最小化する問題には、効率的なアルゴリズムが存在しないことが知られている。ヒューリスティックなアルゴリズムとして、重心法[7]が知られおり、ここではこれにしたがう。

(4.3.5)空間的配置の決定

素性構造の図化表現形式の各部分構造に対して、前節でふれたような『見易さ』に関する図形的特徴にしたがって、その空間的配置を決定する。空間的配置の決定は、極大木の各ノードについて末梢のノードから、自分以下の構造のしめる大きさを計算し、子ノードの相対位置を記録することによって達成される。

5.素性構造の視覚化による辞書記述の支援

辞書記述作業の中心となるのは、記述の妥当性を検証する作業であるが、処理の試行をその手段とすることには、[3.2節]に挙げたような問題がある。そこで著者は、語彙記述を妥当に進めるめの方法として、意図した結果を得るための、個々の語彙項目に関する制約を明らかにし、この制約のもとで、語彙記述を逐行する方法を提案している。この方法では、ある結果を実現するために、個々の語彙記述に課せられる制約は、その記述が、ある素性構造と単一化可能である、というかたちで与えられる。

[Fig.1]のようなマトリックス表現では、二つの素性構造が単一化可能であるかどうかを視察により判断するのは容易ではない。だが、[Fig.7]に示したように、条件として提示された構造をグラフの形式に図式化し、対応する語彙記述をこの図式の上で記述するようにすれば、容易にに、その構造との単一化可能性を維持することができる。なぜならば、その図式の上を擦ったり、新たなものを書き加える限りは、単一化可能性を損なうことはなく、この明確なガイドラインを超えなければ、単一化可能性を損なう記述をなし得ないからである。

このことを利用して、[Fig.7]に示すように、制約である素性構造をグラフの形式に図式化し、該当する語彙の記述を、この上を擦るようにして逐行することにより、それと単一化可能という制限のもとでの語義記述を、容易に逐行することが期待できる。

謝辞

研究の機会を与えてくださるとともに、適切な指針を与えてくださった㈱ATR自動翻訳電話研究所 樽松 明 社長、同言語処理研究室 相沢 輝昭 室長に感謝する。また、熱心に議論に参加してくださった言語処理研究室の諸氏に感謝する。

参考文献

[1] Gunji, T. "Japanese Phrase Structure Grammar", Reidel, 1987

[2] Karttunen,L., "D-PATR A Development Environment for Unification-Based Grammars" CSLI Report No. CSLI-86-61, 1986

[3] 小暮、野村「翻訳実験のためのインタラクティブな支援環境」情報処理学会研究報告85-NL-51-5、1985

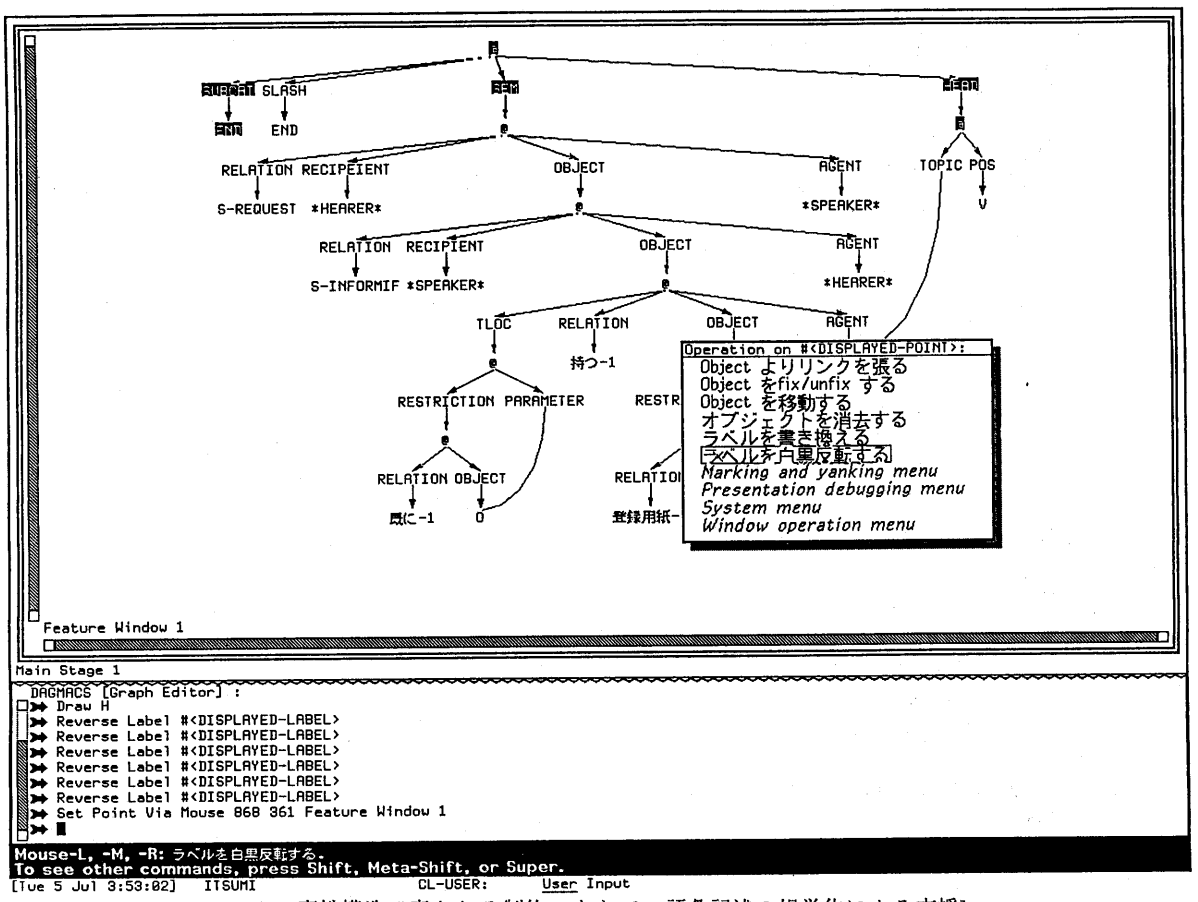
[4] Kogure, K. et. al., "A Method of Analyzing Japanese Speech Act Types", in the Proc. of the 2nd International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages, 1988

[5] 中村、長尾「マルチウインドウを利用した機械翻訳のための文法開発ツール」第29回情処全大、4N-2、1984

[6] Maeda, H. et al. "Parsing Japanese Honorifics in Unification-Based Grammar", in the Proc. of the 26th Annual Meeting of the Association for Computational Linguistics, 1988

[7] Sugiyama,K., Tagawa,S. & Toda,M.(1979): Effective Representations of hierarchical structures, RR-8,IIAS-SIS, Fujitsu Ltd.,

[8] 杉山、「図形言語とスケッチ･エキスパート」情報処理学会研究報告 86-FI-3



[Fig.7素性構造で表される制約のもとでの語彙記述の視覚化による支援]

図式で表現された制約の上で語彙記述を逐行することにより、制約をあらわす素性構造と単一化可能であるという条件を遵守することが極めて容易となる。