

TR-731

選言部の共通情報検査を行なう単一化手法

新納 浩幸、鈴木 浩之、  
飯塚 泰樹（松下）

January, 1992

© 1992, ICOT

**ICOT**

Mita Kokusai Bldg. 21F  
4-28 Mita 1-Chome  
Minato-ku Tokyo 108 Japan

(03)3456-3191 ~ 5  
Telex ICOT J32964

---

**Institute for New Generation Computer Technology**

## 選言部の共通情報検査を行なう単一化手法

新納 浩幸                      鈴木 浩之                      飯塚 泰樹  
shinnou@trl.mei.co.jp    suzuki@trl.mei.co.jp    iizuka@trl.mei.co.jp

松下電器産業株式会社 情報通信東京研究所 基礎研究部  
川崎市多摩区東三田 3-10-1 ... TEL (044)911-6351

単一化に基づく自然言語処理においては、選言として現れる複数の素性構造をどのような形で表記し、その選言的素性構造をどのように単一化するかが重要な問題である。一般には、選言の展開をできるだけ押えることで、単一化を効率化できるため、その観点にたった表記法と単一化処理が提案されてきている。ここでは、Eisele の選言表記とその単一化処理を改良する。改良点は選言部分に更に、その選言の各素性構造が共通してもつ情報のいくつかを持たせることである。選言部分の単一化の際に、まずその共通部分に対して部分的に単一化を行ない、選言を展開しても単一化が失敗する場合を察知する。これによって選言の展開を減らすことができる。また、有効性をみるために Eisele の手法と本手法を実装し、その処理時間を比較した。

### 1 はじめに

近年の文法記述形式として LFG、HPSG など単一化を基本としたものが数多く提案されている。これらの文法では語や句を素性構造で表現し、文法自体も素性構造に変換し、解析処理では、文脈自由型のまとめ上げ規則と文法と語や句の単一化によって、文の構造を解析しようとする。

基本的に上記の方式はすべての曖昧性を導き、用意してある文法の条件を満たさない構造を順次削ってゆき、最終的に残ったものを文の解釈とする方式である。

しかし、この枠組では多数の意味を持つ語を扱ったり、キメの細かい解析結果を出そうとすると、ある部分列に対応する構文意味構造の曖昧性が爆発的に増えてしまい、現実的には処理が不可能になってしまう。

そこで、複数の素性構造を選言を用いることによって、1つの素性構造として表し、それをそのまま単一化の処理にかけることができれば、解析効率が大幅に向上する。また、辞書、文法の記述容易性、記述量、人間による認可性も向上する。

このような背景をもとに選言的素性構造の表記法とその単一化手法の研究が Kasper、Eisele らによって行なわれてきた [Kasper 87][Eisele 88]。

Kasper の手法と Eisele の手法のどちらが優れているかは記述文法や入力文に依存するため一概

には述べられるものではないが<sup>1</sup>、近年の研究は、Kasper の手法をベースとしたものが多い [菅野 89][永田 90][中山 91]。しかし、一方で、選言の扱いについては親ノードに関係のない情報を考慮してゆかない手法が有効であることが知られてきた [Carter 90][Nakano 91]。この手法は Kasper の手法を改良する際には大きな効率化になるが、Eisele の手法では単一化時に参照されないノードを保持していても効率化には特に影響しないので、この点を考慮しなくて良いという長所がある。また選言的素性構造を扱う解析では、解析中に同一部分列に対して同一のカテゴリを持つ句が複数導かれた時に、それらを設定した選言的表記に変換する処理が必要になる。Kasper の表記法では、この処理が比較的複雑になるが Eisele の表記法では単純である。また Eisele の手法は、Kasper の手法よりも使用するメモリが大きい分、速度の面ですぐれていることが指摘されている。

これらの点から Eisele の手法の方が Kasper の手法よりも若干実用的であると考え、ここでは Eisele の手法をベースに Kasper の手法の長所を取り入れる研究を行った。

<sup>1</sup> 選言的素性構造の単一化処理は、選言標準形の充足可能性の判定問題を内部に含むために NP 完全となり、どのように工夫した選言的記述方法を考案しても、最悪の場合には、計算量は選言要素の数の指数関数の大きさになる。

基本的に Eisele の選言表記法とその単一化処理を改良する。改良のポイントは、単一化に基づく解析では解析中の単一化のほとんどが失敗することから、計算のオーバーヘッドが多少生じて、単一化処理の早い段階、つまり選言の展開があまり行なわれないうちに、単一化の失敗を察知することである。

この考えを基に、Eisele の選言表記部分に更に、その選言の各素性構造が共通して持つ情報のいくつかを括り出せるよう拡張した。これによって、選言部分の単一化の際に、まずその共通部分に対して部分的に単一化を行ない、このときの単一化が失敗すると、その選言を展開しても単一化が失敗するために、選言の展開を行わずに単一化の失敗を察知することができる。

本手法の選言表記法では Kasper の表記法のように選言部分の各素性構造の奥深い共通情報を括り出すこともできるし、また選言部分をさらにある属性の値に注目して分類した選言としても表すことができる。

共通部分の単一化は失敗するかどうかの検査だけに利用される。このため、すべての共通部分の情報を取り出す必要はなく、失敗するかどうかの検査にだけ有効であるような情報だけを括り出せば良い。更に、計算コストの高い単一化は途中で打ち切ったり、必ず成功する単一化は予め避けるなどの効率化も可能である。

反面、本手法では、共通部分の単一化が成功した場合、その結果は以後の解析で利用されない。このため二重の計算を行なう場合も存在するが、この計算のオーバーヘッドのコストは、本手法により削減された計算処理のコストに比べ少ないと予測される。

以下、2 節で Eisele の提案した選言表記法とその問題点を説明し、3 節で改良した選言表記法とその単一化を説明し、4 節で本手法の実験、5 節でまとめを行なう。

## 2 選言記述形式 ENF とその問題点

### 2.1 選言記述形式 ENF

以下に Eisele の提案した選言的素性構造の記述形式 ENF (Efficiency oriented Normal Form) の形式を示す。ENF 形式の定義は再帰的な定義になっており、 $\Phi \in ENF$  は  $\Phi$  が ENF 形式であることを示す。

#### ☆ ENF 形式

以下に示す各々の式は ENF 形式である。

- (1)  $NIL$
- (2)  $TOP$
- (3)  $a$  where  $a \in A$
- (4)  $l_1 : \Phi_1 \wedge \dots \wedge l_n : \Phi_n$   
where  $\Phi_i \in ENF \setminus \{TOP\}$   
 $l_i \in L, l_i \neq l_j$  for  $i \neq j$
- (5)  $\Phi \vee \Psi$  where  $\Phi, \Psi \in ENF \setminus \{TOP\}$
- (6)  $\langle p \rangle$  where  $p \in L^*$

上記のシンタクスにおいて、 $A$  はアトム集合、 $L$  はラベル (属性名) の集合である。直観的に、上記のシンタクスは以下の様に捉えられる。

- (1)  $NIL$  はすべての素性構造と単一化可能なもの。  $A \wedge NIL = A, NIL \wedge A = A$  が成立する。
- (2)  $TOP$  はすべての素性構造と単一化不可能なもの。  $A \wedge TOP = TOP, A \wedge TOP = TOP$  が成立する。単一化の失敗の結果値として利用できる。
- (3)  $a$  はアトム。ある属性の値である。
- (4)  $l_1 : \Phi_1 \wedge \dots \wedge l_n : \Phi_n$  は属性名  $l_i$  の値が  $\Phi_i$  であるような素性構造を示す。
- (5)  $\Phi \vee \Psi$  が選言の表記であり、 $\Phi$  あるいは  $\Psi$  が成り立つことを示している。
- (6)  $\langle p \rangle$  はパス表現 (path equivalence) である。素性構造を有効グラフと考えて、 $p$  で示されるラベル (属性名) を辿った値と同じ値を持つことを表している。

以下に ENF 式の例をあげる。

$$\Phi = agr : \langle subj \ agr \rangle \wedge case : (nom \vee acc)$$

上記の  $\Phi$  を一般の素性構造で表せば、

$$\begin{bmatrix} subj & [agr \ [X]] \\ agr & [X] \\ case & nom \end{bmatrix}$$

または、

$$\begin{bmatrix} subj & [agr \ [Y]] \\ agr & [Y] \\ case & acc \end{bmatrix}$$

という選言で表せる。

$$\Phi_1 = \begin{bmatrix} a & 1 \\ b & 2 \\ c & \begin{bmatrix} d & 3 \\ e & 4 \end{bmatrix} \end{bmatrix}, \Phi_2 = \begin{bmatrix} a & 2 \\ b & 2 \\ c & \begin{bmatrix} d & 3 \\ e & 5 \end{bmatrix} \end{bmatrix}, \Phi_3 = \begin{bmatrix} a & 1 \\ b & 1 \\ c & \begin{bmatrix} d & 3 \\ e & 6 \end{bmatrix} \end{bmatrix}$$

図 1

## 2.2 ENF の問題点

ENF の選言表記方法は値選言 (value disjunction) と呼ばれる表記法であり、選言を素性構造の奥深くに押し込めることが可能なので、解析が進むに従って参照されなくなった選言を展開することがなくなるという利点がある。

しかし値選言の表記法では選言部分が複数箇所に現れると、選言をフラットに並べた形にしか記述できないという欠点を持つ。

例えば図 1 で示される素性構造を考える。各素性構造を ENF 式で表すと以下のようになる。

$$\Phi_1 = (a : 1) \wedge (b : 2) \wedge (c : d : 3) \wedge (c : e : 4)$$

$$\Phi_2 = (a : 2) \wedge (b : 2) \wedge (c : d : 3) \wedge (c : e : 5)$$

$$\Phi_3 = (a : 1) \wedge (b : 1) \wedge (c : d : 3) \wedge (c : e : 6)$$

ここで図 1 の各素性構造の選言を ENF 式で表そうとすると、(1 式) のように選言をフラットに並べた形にしか記述できない。

$$\Phi_1 \vee \Phi_2 \vee \Phi_3 \quad (1 \text{ 式})$$

これは Kasper の手法で採用されている一般選言 (general disjunction) のように

$$A \wedge (B \vee C)$$

の形式を、ENF が持っていないからである。

ここで (1 式) で表される選言的素性構造と以下の (2 式) で示される素性構造が単一化される場合を考える。

$$c : d : 6 \quad (2 \text{ 式})$$

ENF 上の単一化では、選言は展開されて、図 1 に示される各  $\Phi_i$  が (2 式) で示される素性構造 (図 2) と単一化される。その結果、各々の単一化は失敗し、全体として単一化は失敗する。

$$\begin{bmatrix} c & \begin{bmatrix} d & 6 \end{bmatrix} \end{bmatrix}$$

図 2

一方、各  $\Phi_i$  には、

$$c : d : 3 \quad (3 \text{ 式})$$

という共通式を持っている。このため一般選言のように共通式を括り出せる場合、(1 式) と (2 式) の単一化では、(3 式) と (2 式) とを最初に単一化することで、選言の展開を行なう以前に単一化結果 (この場合 TOP) が得られる。

上記のように ENF 形式では、選言部分の共通情報を括り出せないために、一般選言を用いた単一化のように選言の展開を行なう以前に単一化の失敗を得ることができないという問題がある。

## 3 拡張 ENF とその単一化手法

### 3.1 拡張 ENF

以下に本論文の提案する拡張 ENF 形式を示す。

拡張 ENF 形式の定義も ENF 形式同様、再帰的な定義になっている。

#### ★ 拡張 ENF 形式

以下に示す各々の式は拡張 ENF 形式である。

(1) *NIL*

(2) *TOP*

(3) *a* where *a* ∈ *A*

(4)  $l_1 : \Phi_1 \wedge \dots \wedge l_n : \Phi_n$   
where  $\Phi_i \in \text{拡張 ENF} \setminus \{TOP\}$   
 $l_i \in L, l_i \neq l_j \text{ for } i \neq j$

(5)  $\Phi \vee \Psi$  where  $\Phi, \Psi \in \text{拡張 ENF} \setminus \{TOP\}$

(6)  $\langle p \rangle$  where  $p \in L^*$

(7)  $\langle \Theta ; \Phi \vee \Psi \rangle$  .

s.t.  $\Theta \in \text{拡張 ENF} \setminus \{TOP\}$

$\Phi, \Psi \in \text{拡張 ENF} \setminus \{TOP\}$

$\Theta \sqsubseteq \Phi, \Theta \sqsubseteq \Psi$

記号  $\sqsubseteq$  は、拡張 ENF 式を論理式とみた場合の包含関係である。または、拡張 ENF 式を素性構造と見た場合の包摂関係 (subsumption relation) である [今村 91]。

ENF と拡張 ENF の違いは、本質的に (7) 式の有無である。以下、ここでは (7) 式を 拡張 ENF 式 と呼ぶことにする。この式の意味は、概ね、 $\Theta$  で表される素性構造が、 $\Phi$  で表される素性構造と  $\Psi$  で表される素性構造の共通情報となっていることを示す。

例として図 1 の各素性構造の選言を拡張 ENF 式で表すと以下のようになる。

$$\langle (c:d:3) ; \Phi_1 \vee \Phi_2 \vee \Phi_3 \rangle \quad (4 \text{ 式})$$

ここで、拡張 ENF 式は共通情報である  $\Theta$  が選言部分の要素である  $\Phi$  と  $\Psi$  の最大の共通情報である必要はないという長所を持つ。これにより  $\Phi$  と  $\Psi$  の共通情報のうち解析で有用となるものだけを  $\Theta$  に付加すればよい。

### 3.2 拡張 ENF に対する単一化手法

拡張 ENF に対する単一化は、Eisele の与えた ENF に対する単一化を拡張することで得られる。

まず、ENF に対する単一化を説明する。ENF の単一化の方法は、通常の素性構造の単一化を素直に拡張した形になっている<sup>2</sup>。基本的に引数である 2 つの素性構造の属性名を比べ、一方のみに現れる属性名に対しては、その属性名と属性値を単一化結果の素性構造に付加し、共通して現れる属性名に対しては、その属性値を再帰的に単一化し、その結果と属性名を単一化結果の素性構造に付加する。

例えば、図 3 で示される素性構造  $\Phi$  と  $\Psi$  の単一化を考える。

$$\Phi = \begin{bmatrix} a & \phi \\ b & \phi_b \end{bmatrix} \quad \Psi = \begin{bmatrix} a & \psi \\ c & \psi_c \end{bmatrix}$$

図 3

片方だけに現れる属性名として  $b$  ( $\Phi$  にのみ現れる) と  $c$  ( $\Psi$  のみに現れる) がある。それらの属性名と属性値を単一化結果の素性構造に付加する (図 4)。

<sup>2</sup> 属性の値が選言でなければ、そのまま通常の単一化処理として使える。

$$\begin{bmatrix} b & \phi_b \\ c & \psi_c \end{bmatrix}$$

図 4

次に共通して現れる属性名には  $a$  がある。 $\Phi$  と  $\Psi$  の  $a$  の属性値  $\phi$  と  $\psi$  を単一化して、その単一化結果 (ここでは  $\theta$  としておく) と属性名  $a$  を図 4 の素性構造に付加し、 $\Phi$  と  $\Psi$  の単一化結果が得られる (図 5)。

$$\begin{bmatrix} a & \theta \\ b & \phi_b \\ c & \psi_c \end{bmatrix}$$

図 5

次に選言の扱いを説明する。ENF では、選言はある属性の値として現れる。単一化の際に一方が選言であった場合、それを展開することで単一化が行なわれる。例えば、図 3 の  $\Phi$  の  $a$  の属性値  $\phi$  が、(5 式) の選言の形をしている場合、各々の  $\phi_i$  が  $\psi$  と単一化される。

$$\phi = \phi_1 \vee \phi_2 \vee \dots \vee \phi_n \quad (5 \text{ 式})$$

$\phi_i$  が  $\psi$  と単一化された結果を  $\theta_i$  とすると、(6 式) で表される ENF が  $\phi$  と  $\psi$  の単一化結果となる。

$$\theta_1 \vee \theta_2 \vee \dots \vee \theta_n \quad (6 \text{ 式})$$

ENF の単一化手法の特徴はパス表現の扱いにあるが、本論文の拡張部分には直接関係しないので、ここでは概説だけを行なう。

パス表現の単一化の場合、ある属性値を破壊的に書き換えなくてはならない。このため素性構造のコピーが必要なり、計算コストが高くなる。そこで、ENF の単一化では、単一化の核になる関数が、局所的結果と、大域的結果の 2 つを返値としている。

直観的には、局所的な単一化結果は、ある属性名の下で独立に成り立つ構造、大域的な単一化結果は、パス表現の単一化によって生じた副作用的に影響を与える条件と考えられる。ENF の単一化の戦略は、局所的な単一化結果を順次求めてゆき、そこで単一化を失敗させようとする。大域的結果は、最後までためこんでゆき最後にそれらから適切な構造を作ろうとする。局所的な単一化結果は比較的、軽い処理

で求められるために、結果的に早期に単一化を失敗させることになり、全体の効率が良くなっている。

次に拡張 ENF に対する単一化を説明する。基本的に ENF の単一化処理をベースにしている。大域的な単一化結果を求める手法は ENF の単一化と同じであり、局所的な単一化結果を求める際に、拡張 ENF 式が扱えるよう拡張している。

拡張 ENF では、選言の表し方が ENF の選言と拡張 ENF 式の 2通りの表し方がある。例えば、(5式) の  $\phi$  が(7式) のような拡張 ENF 式で表されたとする。

$$\phi = (\phi_{common} ; \phi_1 \vee \phi_2 \vee \dots \vee \phi_n) \quad (7式)$$

ここで、 $\phi_{common}$  は、各  $\phi_i$  の共通情報である。

$\phi$  と  $\psi$  の単一化では、まず  $\phi_{common}$  と  $\psi$  の単一化が行なわれる。この結果が TOP (単一化の失敗) であった場合には、 $\phi$  と  $\psi$  の単一化結果として TOP を返し、TOP 以外の場合は、ENF の選言の単一化と同様、 $\phi_1 \vee \phi_2 \vee \dots \vee \phi_n$  と  $\psi$  を単一化し、その結果を  $\phi$  と  $\psi$  の単一化結果とする。ただし、 $\phi_1 \vee \phi_2 \vee \dots \vee \phi_n$  と  $\psi$  の単一化結果が再び選言になった場合には、その選言から拡張 ENF 式を作成し、それを  $\phi$  と  $\psi$  の単一化結果とする。

選言から拡張 ENF 式を作成する際に、共通情報を括り出す必要がある。この処理は計算コストが大きい、ここでは、 $\phi_{common}$  が再び共通情報となる性質を利用している。

$\phi_{common}$  と  $\psi$  の単一化は失敗するかどうかの検査だけに利用され、成功した場合の単一化結果は以後の解析で利用されない。つまり検査のための単一化が成功すると、その検査にかかった計算が無駄になってしまう。このため、ここでは基本的に計算コストの低い処理だけの検査を行ない、計算コストの高い処理は検査を中断し、本来の単一化に処理を移している。例えば、 $\phi_{common}$  あるいは  $\psi$  がバース表現の場合、単一化が失敗するかどうかの検査の処理は計算コストが高いので検査は中断し、本来の単一化に処理を移す。

最後に、 $\phi$  と  $\psi$  がともに拡張 ENF 式の場合の単一化を示す。ここで、 $\phi$  は(7式)  $\psi$  は(8式)で表されているとする。

$$\psi = (\psi_{common} ; \psi_1 \vee \psi_2 \vee \dots \vee \psi_n) \quad (8式)$$

まず、共通情報部分どうし ( $\phi_{common}$  と  $\psi_{common}$ ) の単一化を行なう。この単一化が失敗した場合は、 $\phi$  と  $\psi$  の単一化結果として TOP を返す。共通部分

の単一化が失敗しなかった場合、 $\phi_{common}$  と各  $\psi_i$  を単一化し、各  $\psi_i$  の中から単一化が失敗しなかったもののだけの選言を作る(9式)。

$$\psi_{i_1} \vee \psi_{i_2} \vee \dots \vee \psi_{i_m} \quad (9式)$$

同様に、 $\psi_{common}$  と各  $\phi_j$  を単一化し、各  $\phi_j$  の中から単一化が失敗しなかったもののだけの選言を作る(10式)。

$$\phi_{j_1} \vee \phi_{j_2} \vee \dots \vee \phi_{j_k} \quad (10式)$$

次に、(9式)と(10式)の単一化を行ない、その結果を  $\phi$  と  $\psi$  の単一化結果とする。ただし、(9式)と(10式)の単一化結果が再び選言になった場合には、その選言から拡張 ENF 式を作成し、それを  $\phi$  と  $\psi$  の単一化結果とする。この際、 $\phi_{common}$  と  $\psi_{common}$  が再び共通情報となる性質を利用する。

### 3.3 共通情報の括り出し処理

本手法が従来手法 (ENF とその単一化処理) よりも効率化されるかどうかは、選言の中からどの程度有効な共通情報を括り出せるかに依存するが、一般にこの処理を解析中にダイナミックに行なうことは困難である。

これは共通情報を見つけるのが困難であるだけでなく、より有効な表記を行なう場合、ある属性によるグループ化も考慮しなくてはならないからである。

例えば(4式)は属性  $b$  に注目して(11式)のように拡張 ENF 式を入れ子にした式に変形できる。

$$\langle (c:d:3) ; \langle (b:2) ; \phi_1 \vee \phi_2 \rangle \vee \phi_3 \rangle \quad (11式)$$

属性  $a$  に注目して変形を行なえば(12式)の形になる。

$$\langle (c:d:3) ; \langle (a:1) ; \phi_1 \vee \phi_3 \rangle \vee \phi_2 \rangle \quad (12式)$$

(11式)と(12式)のどちらが、解析にとってより有効な変形であるかは、括り出せる選言要素の数や属性の持つ文法上の意味に依存するために、それらを総合的に評価した拡張 ENF 式を作成するのは困難である。

本来、ENF の表記法とその単一化は、この困難性を避けることができるために効率的になっている。基本的に本手法もこの点は踏襲し、選言がフラットに並んだ形からは、特に共通情報を取り出さない。

ただし、共通情報との単一化の検査を通過して得られた選言では、その共通情報は再び選言の各要素の共通情報となっているため<sup>3</sup>共通情報として括り出すことができる。

以上の議論は解析中に行なわれる動的な処理であるが、ある単語の辞書情報に対応する素性構造や文法に対応する素性構造は固定されているため、予め、より効率的な拡張 ENF 形式に変換しておくことができる。

例えば、 $\langle S \rangle \Leftrightarrow \langle Np \rangle \langle Vp \rangle$  という書き換え規則の持つ素性構造が、いくつかの選言で表される場合を考えてみる。文法の書き換え規則が起こった場合の処理を考えると、「まず  $\langle Vp \rangle$  のテンズで場合分けして、次に  $\langle Np \rangle$  が代名詞かどうかで場合分けして、……」といった処理になる場合が多い。これはプログラムの場合は、if-then 文が入れ子構造になっている場合に相当する。

文法の処理上の意味が分かれば、if-then 文の条件や、if-then 文の入れ子構造の順序を考慮することで、解析の効率が上がるような記述が可能である。

拡張 ENF 式において、ある属性に注目したグループ分けをすることは、この if-then 文の入れ子構造を作ることにはほぼ対応する(図6参照)。つまり拡張 ENF 式は、if-then 文の入れ子構造のマクロと見ることができるため、文法の処理上の意味を考えて、効率的な記述をすることが可能になる。

```
if Attri = a then [処理A]
else [処理B]
```

↓ ↑

```
< ( Attri:a ) ; [処理A] > ∨ [処理B]
```

図6

辞書情報に対する素性構造も曖昧性解消に有効であるような属性名に注目して選言をグループ分けしておくこともできる(図7)。

```
< ( Attri:a ) ; [Aグループ辞書情報] >
∨ < ( Attri:b ) ; [Bグループ辞書情報] >
```

図7

<sup>3</sup>これは単一化の非破壊性から明らかである。

## 4 実験

本手法の有効性を計るために、従来の ENF とその単一化を用いた解析システムと、ここで提案した拡張 ENF とその単一化を用いた解析システムを作成し処理時間を比較した。パーサは CKY パーサ[野村 88]を利用した。

例文

- (a) My father wants to play tennis with me
- (b) He was a pilot when he was young
- (c) I saw a man with a telescope on the park

|        | 従来の手法  | 本手法   |
|--------|--------|-------|
| 例文 (a) | 10.850 | 5.767 |
| 例文 (b) | 5.117  | 3.017 |
| 例文 (c) | 11.800 | 7.400 |

実行時間(秒)

実験は、SPARC station 1 の KCL 上で、それぞれの例文を10回解析し平均時間を測定した。用いた辞書では、1単語に対して平均2つの選言を持っている。文法は20個程度の書き換え規則を持つ簡単なものであるが、平均2つの選言を持っている。

上記の実験では、従来手法から2倍弱の効率化が計れた。

文法記述の工夫は文法処理上の意味を考えて、予め、最適な拡張 ENF 式に変換しておいた。実験で得られた効率化は、ほとんどこの部分が担っていた。

## 5 おわりに

本報告では、選言的素性構造の単一化処理の効率化のために、選言部分に更に、その選言の各素性構造が共通してもつ情報のいくつかを持たせる拡張 ENF 式を導入することで、Eisele の選言記述形式 ENF を拡張し、その上の単一化処理を作成した。また簡単な実験を行なって、その有効性を確かめた。

本手法が従来手法(ENF とその単一化処理)よりも効率化されるかどうかは、計算コストの低い処理によって、どの程度まで有効な共通情報を括り出せるかに依存する。

一般に解析中の動的な処理で共通情報を括り出す処理は計算コストが高いため、この点からの効率化は難しいが、辞書記述や文法は固定的なものであるため、予め有効な形に記述しておくことができ効率化できる。

解析中に動的に行なう共通情報の括り出しの処理も、文法に依存した処理を行なうことにより、計算コストを低くすることもできるし、有効な情報を括り出すこともできる。例えば、名詞句では数の情報、格の情報、品詞の細分類情報などが曖昧性解消に役立つと考えて、その属性だけに注目したり、素性構造の奥深くの情報は曖昧性解消には有益ではないと考え、共通情報の探索を素性構造のある深さ以上は行なわないなどの処理が導入できる。

結果的に、文法に依存した解析エンジン部を作成すること、文法をプログラムという意識で記述することが効率化には有効であるということが言える。ここでは、それを受け入れる記述形式とその処理を提案したと見ることもできる。

ただし、これは文法をプログラム(手続き)的に見る必要があり、本来単一化文法の持つ宣言的記述の長所を欠く。今後は宣言的に記述された文法から手続き的な文法に変換する処理を考察してゆきたい。

なお、本研究は財団法人新世代コンピュータ技術開発機構からの受託(発注番号 6303 号)により行なわれたものである。

## 参考文献

- [Carter 90] Carter, D.: "Efficient Disjunctive Unification for Bottom-Up Parsing", *COLING-90*, pp 70-75 (1990)
- [Eisele 88] Eisele, A. and Dörre, J.: "Unification of Disjunctive Feature Descriptions", *ACL-88*, pp 286-291 (1988)
- [今村 91] 今村誠:「素性構造の単一化」, 情報処理 Vol32 No.10, pp1070-1078(1991)
- [菅野 89] 菅野祐司, 長尾健司:「曖昧さの効率的処理のための構文解析手法について」, *NL74-6*, 41-48 (1989)
- [Kasper 87] Kasper, R.: "A Unification Method for Disjunctive Feature Descriptions", *ACL-87*, pp 235-242 (1988)
- [Maxwell 89] Maxwell, J.T. and Kaplan, R.: "An Overview of Disjunctive Constraint Satisfaction" *Proceedings of the International Workshop on Parsing Technologies*, pp 18-27 (1989)
- [永田 90] 永田昌明, 小暮潔:「文法規則の構造共有による選言的素性構造単一化の効率化」, 情報処理第40回全国大会, 2E-5 (1990)
- [Nakano 91] Nakano, M.: "Constraint Projection: An Efficient Treatment of Disjunctive Feature Descriptions", *LPC-91*, pp219-228 (1991)
- [中野 91] 中野幹生:「Constraint Projection - 制約射影による構文解析 -」, *NL83-12*, pp89-96 (1991)
- [中山 91] 中山圭介, 松本裕治, 長尾真:「選言的素性構造を用いた自然言語処理の曖昧性の表現とその解消方法」, 情報処理第43回全国大会, 3G-2 (1991)
- [野村 88] 野村浩郷:『自然言語の基礎技術』, 電子通信学会編, コロナ社 (1988)