自然言語処理 91－2
（1992. 9. 17）

英日機械翻訳システムにおける並列関係の検出

武田 紀子
成蹊大学 工学部

文章を解析する上で、文中に現れる並列関係を正確に見つけることは、構文の曖昧性を解消するためにも大切なことである。英文中に現れる等位接続詞は、その前後の句を並列につなげるが、句の構造が入れ子になっていたり、品詞が確定できないために、等位関係を一意的に決めることが出来ない場合がある。この問題を解決するために、”見かけ上同じようなものは、並列になり易い”ということに着目し、処理をすることとした。ここで、”同じようなもの”とは、同じ語、対になる語からなる句、同じ形をした句、等をいい、これらの判断を、英日機械翻訳システムの解析ルーチンに組み込み、いくつかの例文に対し、実験を行った。

Detection of Coordinate Relations in English－Japanese Machine Translation

Noriko Takeda
Department of Industrial Engneering
Seikei University

To eliminate the ambiguity of a sentence, it is very important to accurately detect the coordinate relations exist in the sentence. It is particularly difficult to decide the coordinate relations in the case when sentence with coordinate conjunctions are nested clauses or the part of speech of a word is not definite.

However, after a few observations, it can be noticed that those clauses with similarity tend to coordinate easily. Here, similarity means those clauses with same or paired-word or having same syntactic structure. The observation was then applied to the Coordinate Analysis Routine of the system and a few experiments have been performed using some sample sentences.

１．はじめに

文章を解析する上で、文中に現れる並列関係を正確に見つけることは、構文の曖昧性を解消するためにも、大切な問題である。語句を対等な関係で、接続する等位接続詞は、接続詞の前後にある同等なもの同士を、等位関係にあるとみなす。ここで、同等なものとは、文法的意味において等しいもの、形態が似ているもの、意味的に、等位関係になるもの等が考えられる。しかし、英文の構文解析の過程で、等位関係の処理を行おうとした場合、文のその他の解析における曖昧性が、解消されていないことが多い。例えば、多品詞語における品詞の特定、前置詞句の修飾関係、多義語における意味の特定、等である。又、句構造が入れ子になっている場合や、一文中に、２つ以上の等位接続詞が含まれている場合等も、各接続詞が等位関係を形成する範囲を決めることは難しい。

ここで、人間が、比較的容易に複雑な等位関係を見つけることができる理由を考えると、

１．文全体を一目見て、全体の構文をおおまかに把握することができる。

２．文中の同じような意味、形、あるいは、対になるような意味、形を、文を見ることにより、容易に発見できる。

が、挙げられる。そして、正しい等位関係を求める過程で、同時に、その他の曖昧性の解消も行っている。

翻訳システムで、あらゆる解析の可能性をとっておき、それをもとに最適な解析結果を得ようとすると、莫大な記憶領域と、処理時間を費やすことになる。そこで、解析の効率を考え、”見かけ上、同じようなもの同士を見つける”ということに重点を置き、次のような方針で、等位接続詞による並列関係の処理を行うこととした。

１．等位関係を形成するもの全体を文法的な１つの構成要素とする。

例．　hardware and software

名詞の並列　・・・　１つの名詞

advanced software engineering and sofisticated software techniques

名詞句の並列　・・・　１つの名詞句

２．この為、各構文要素（e.g. 名詞、名詞句、前置詞句 etc）の処理の最後に、その要素が、それに続く文により生成される構文要素と、等位関係を生成するかどうかを調べる。この時、”同じようなもの同士を見つける”ための規則を割り込ませ、等位関係解析の判断基準に加える。

次に、処理の概要、個々の等位関係解析のための規則、実現方法について述べる。

２．処理の概要

構文解析は、与えられた文法規則をもとに、top-downに行う。そして、各構文要素に対する文法規則の最後に、conj と指定されていると、今解析された要素が、以下の文から生成される要素と、等位関係を形成するかどうかを調べる。文法規則は、リストで記述され、例えば、名詞句の文法は、次のように書かれる。

```
（名詞句　（（all　限定詞　形容詞句
                動詞の分子形　名詞）
           （all　限定詞　動詞の分子形
                          名詞）
                ・・・・
           （the　形容詞））
           conj
                ・・・・                    ）
```

名詞句の処理過程で、conj にくると、文の次の語は、接続詞かどうかを調べ、接続詞の場合は、以下の文から、名詞句が、生成されるかをみる。そして、

名詞句　接続詞　名詞句

と並ぶと、等位関係は、成立するとみなされる。等位関係が成立すると、全体で１つの構文要素（e.g. 名詞句）とし処理をすることとした。これにより、等位関係の解析は、前後の文の解析に影響を及ぼさない。

しかし、例えば、

名詞句１　接続詞　名詞句２

と並んでいる場合でも、文全体を見ると

主語の名詞句　動詞　目的語の名詞句１　接続詞　主語の名詞句２　動詞　・・・

となっていることもあり、この場合は、明らかに名詞句の並列ではなく、文の並列となる。そこで、まず、次のような順序で、等位関係を見つけることとした。

［規則１］

１．文、従文、不定詞句、分子句の並列が成り立つときは、それらの並列とする。

２．接続詞の前後の単語の品詞が同じ時は、単語の並列とする。

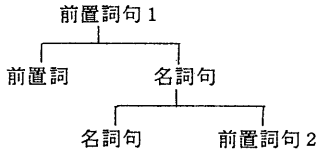
３．接続詞の前後の句（e.g. 名詞句、形容詞句）が同じ場合は、句の並列とする。

しかし、この規則のみでは、以下のような問題点も生じる。

［規則１］の２の適用において、単語の品詞が特定できている場合はよいが、接続詞の後の単語が多品詞

語の場合、この単語の品詞を接続詞の前の単語の品詞と同じものとしてしまうため、構文解析全体に影響を与える間違いをすることがある。

［規則１］の３の適用をすると、接続詞の前の前置詞句の解析結果が、



となった場合、接続詞の後の前置詞句は、常に、前置詞句２と等位関係となるように解析される。

又、コンマを含む３つ以上の要素の並列の中に、別の等位関係が入れ子になって存在する場合の解析も正しくなされない。

そこで、個々の構文要素の等位関係を処理する時に、等位関係成立を制限するような規則を加えることとした。この規則の適用のために、以下のスタックを用意し、解析の軌跡を保存する。

・句のスタック　（スタック１）
　動詞句、名詞句、前置詞句の処理に入った時、処理対象の句の名前を入れる。
・文のスタック　（スタック２）
　句のスタックに入れる時、同時に、解析の対象となる文、つまり、処理が終わっている部分を除いた残りの文を入れる。

句の処理が終わると、この２つのスタックの中味は、取り除かれる。

・等位関係を形成する句のスタック　（スタック３）
　コンマを含む等位関係に入った時、等位関係を形成するために必要とされる句の名前を入れる。

このスタックの中味も、処理が終わると取り除かれる。

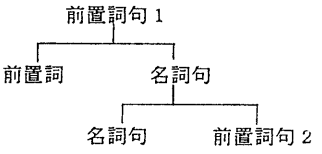
これらの解析の軌跡を利用し、個々の句等の等位関係の解析にあたる。

３．等位関係形成のための制限規則

３．１．入れ子となった句のための規則

（１）前置詞句の処理

等位接続詞の前後の句が、前置詞句の場合は、前置詞句の並列となる。しかし、接続詞の前の文の解析結果が、



の場合、必ずしも直前の前置詞句２と、等位関係を形成するとは限らない。前置詞においては、同じ格、対応する格を生成する句同士が、並列になり易いということを利用し処理することも考えられるが、格は、文全体の解析が終わらないと、決まらないことも多い。そこで、解析の処理効率を考え、以下の規則を加えて解析することとした。

［規則２］
１．同じ前置詞で始まっている前置詞句があれば、それらの並列とする。
２．対になり易い前置詞（toとfrom, inとout etc.）で始まっている前置詞句があれば、それらの並列とする。
３．接続詞の直前の前置詞句と並列とする。

この処理を行うために、解析の軌跡を保存したスタックを利用する。次の例文により、具体的処理過程を説明する。

It can be asked for one line descriptions of commands specified by name, or for all commands whose description contains any of a set of DS.

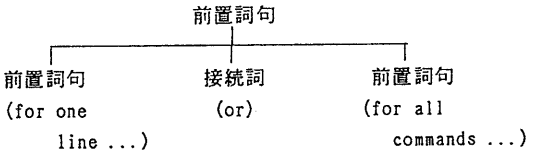
この文では、or の前後の句により、等位関係が形成される。or の直後の for からの文を解析しようとしている時のスタックの状態は、次のとおりである。

句スタック（スタック１）
前置詞句
名詞句
前置詞句
名詞句
前置詞句
動詞句

文スタック（スタック２）
by name, or for all ...
commands specified ...
of commands .....
one line description ..
for one line .....
asked for one line ....

ここで、スタックの状態を見ると、for で始まる前置詞句の処理が登録されている。従って、［規則２］の１が適用され、or の後の for 以下の文からなる前置詞句は、for one line ... から、or までの語からなる前置詞句と並列となるとみなされ、by name から

なる前置詞句の処理は終了し、スタックから除かれる。of command で始まる前置詞句の処理も同様の理由により、終了し、



という解析結果が得られる。

（2）名詞句の処理

名詞句の場合は、文法も複雑で、多様な形態をとり、(to-不定詞句、that句、分子句等も名詞句となり得る)名詞の意味もいろいろある。又、構文上、名詞句は、主語、目的語、補語として現れる。そこで、まず、［規則1］の1、2を適用し、名詞句の等位関係が成立するかどうかを調べる。［規則1］の1、2に当てはまらない時、名詞句の並列に関する解析が始まる。しかし、名詞句を形成する句のなかに、名詞句が入れ子になって存在している場合は、等位関係にある名詞句の範囲を決めるのが難しい。接続詞の前後の全ての名詞句の解析結果を見て、名詞句を生成する名詞が同じもの、同じ意味を持つもの、対になっているものや、同じ形態を持った名詞句を見つけ、それらに等位関係を適用することも考えられるが、効率は悪い。又、同じ意味といった場合は、意味分類の精度に依存するし、同じ形態といっても、同じ形態の定義も一意的に決まらず、ギャップのある場合にはさらに複雑となる。

そこで、次のような規則を加え、名詞句の等位関係の処理を行うこととした。

［規則3］

1．接続詞の後の名詞句を生成する文の先頭の3語と同じ語をその先頭の3語に含む名詞句があれば、その名詞句が、等位関係を形成する。
但し、a, the のような冠詞は除く。

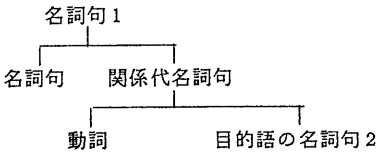
以下の文に、この規則を適用させる。

One might provide a type trunk-module in a program dealing with telephony or a type list of paragraphs for a text processing programs.

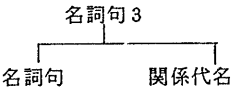
すると、or によって繋がれる名詞句は、a type .. から始まる2つの名詞句と解析される。この処理は、スタック1、スタック2を利用することによってなされる。

2．名詞句が、主名詞を修飾する句を含む場合は、同じ形を持った名詞句を並列させる。

例えば、接続詞の前の名詞句の解析結果が、

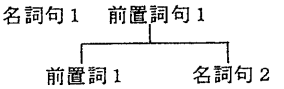


とする。接続詞の後の名詞句の解析結果が、



の時は、名詞句3は、名詞句1と等位関係を持つとし、それ以外は、名詞句2と等位関係を持つとする。

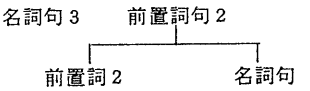
しかし、接続詞の前の文の解析結果が、



となっている場合、この時点では、前置詞句1は、名詞句1にかかるかの決定はなされていないことが多い。ここで、接続詞の後の文が、名詞句（名詞句3と呼ぶ）を生成するとこの名詞句の形と、前の名詞句の形を比べ、同じようなものに等位関係をもたすこととし、次のような規則を加えた。

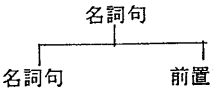
［名詞句＋前置詞句のための規則］

1．名詞句3の解析結果が、



で、前置詞1と、前置詞2が同じ前置詞、或いは、対になっている前置詞の場合、前置詞句1、2は、共に名詞句にかかり、名詞句1＋前置詞句1と、名詞句3＋前置詞句2とが並列となる。

2．3つ以上の名詞句の並列が、コンマをはさみなされている時、間の名詞句の形が、



の場合は、前置詞句1、2は、名詞句にかかり、名詞句＋前置詞句の並列となる。

3．名詞句3の後に、前置詞句がない場合は、名詞句3は、名詞句2と等位関係となる。

4．その他の場合は、文全体を解析した後、次の基準により、解析木を組み直す。

・前置詞句2が、必須格を形成していたり、動詞にかかる場合は、名詞句3は、名詞句2にか

かり、前置詞句２は、動詞にかかる。
・それ以外、前置詞句は、直前の名詞句にかかり、名詞句＋前置詞句の並列となる。

以下に、解析の例を示す。

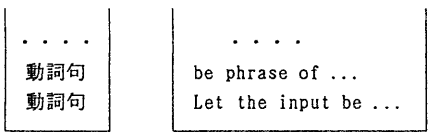
Example include simple classes for symbol
名
table management, stack manipulation and set
詞句　名詞句
manipulation.
名詞句

-F markes directories with a trailing slash
名詞句
and executable file with trailing asterisk.
名詞句

但し、接続詞の前後の名詞句が共に１つの名詞からのみなっている場合は、［規則１］の２が、名詞句の並列の解析の前に適用され、名詞の並列となる。

（３）動詞句の処理
動詞句が入れ子になっている場合も、
・より同じようなものを（スタック１、スタック２を参照することによって見つけられる。）、等位関係で繋げる。
という規則をくわえる。
”同じようなもの”の判断基準は、以下のとおりである。
１．同じ単語
２．同じ時制
３．同じ意味マーカ

例.
Let the input be phrase of a language,
動詞句１
and let the output be approval or disapproval.
動詞句２
この例文で、and 以下の let からを解析しようとしている時のスタックの状態は、次のとおりである。



スタック１　　スタック２

動詞句２は、be 以下の動詞句との並列とはならず、同じ動詞 let から始まる動詞句１と並列となる。
又、同じような動詞句が見つからない場合は、接続詞の直前の動詞句と等位関係にあるとみなされる。

３．２．入れ子となっている等位関係のための規則
等位関係を形成する句の中に、別の等位関係が存在する場合がある。接続詞をはさんだ２つの構成要素の並列の場合は、問題ないが、（例えば、A and B and C は、Aと、B and C の並列）コンマによる３つ以上の並列の中に、下のように、別の等位関係が存在する場合は、最後まで解析をしなければ正しい結果が得られない。

例.
A, B, C and1 D, and2 E
（A,B,C,D,E は、すべて同じ句を生成するものとする）

ここで、文を先頭から解析していくと、and1 D までの解析で、A,B,C,D の並列と解析され、これ全体と、E との並列と解析される(1)。しかし、これは、A, B, C と D との並列と、E との並列と解析することも出来(2)、どちらが正しいかは、A, B, C, D, E の意味、形態を見なければ判断出来ない。

(1) A, B, C and D, and E
(2) A, B, C and D, and E

上のように、and1 の前にコンマがなく、and2 の前にコンマがある時は、（２）の場合の方が多いので、いかのような規則を加えた。

・コンマによる２つ以上の句の並列において、直前にコンマを伴わない接続詞がきたとき、それ以降の文に、”, 接続詞・・・”が含まれ、２つ目の接続詞以降の文が、同じ句を生成している場合は、前の接続詞は、入れ子になった等位関係を含む句を生成するとみなす。

そして、次のような手順で、等位関係を形成する句のスタック（スタック３）を利用して処理することにする。
１．等位関係の処理に入ると、スタックに等位関係を形成する句の名前が入る。
上の例で、A, B,C, D, E は、すべて名詞句とする。A, の解析にきた時、このコンマが並列のためのコンマと判断されると、A が名詞句と解析されているので、スタック３には、名詞句と入る。

スタック３

２．接続詞の解析にきた時、文の後に、別の等位接続詞が含まれて入るかどうかを調べる。含まれている場合は、その接続詞の後の文が、スタック３のｔｏｓに入って句を生成するかどうかをみる。生成している場合は、等位関係が入れ子になっているとみなす。

上の例では、and1 の後に、and2 があり、and2 の後の E は、名詞句を生成するので、and1 によって繋がれるのは、C と D のみと判断される。

３．但し、１、２の手順は、最初に出会った接続詞の前に、コンマがある場合や、単語の並列の場合には適用されない。又、１、２が成立しない場合は、接続詞のあとの句は、その直前の句と等位関係にあるとみなされる。

この手順により、解析された例を下に示す。

The standard stream input/output library

describe here provides a flexible and efficient

for handling character input/output of integer,
名詞句１
floating point numbers, and character strings,
名詞句１の続き
and a simple model for extending it to handle
名詞句２
user-defined type.

名詞句１と名詞句２とが並列となり、character strings は、of から始まる前置詞句を形成する名詞句の一部と解析される。

３．３．多品詞語の処理

一般に、接続詞で繋がれる句の構文要素は、文法的に、同じ形をしている。従って、接続詞の前の文の解析が終わった時点で、後の文の解析結果は、予測される。これは、単語の等位関係処理過程で、多品詞語の品詞の確定に使われる。特に、３こ以上の語の並列においては、曖昧性が解消される可能性も高く、未登録語の品詞確定にも役立つ。しかし、単語の並列を優先させたことにより生じる誤りもある。名詞と動詞の両方を品詞として持つ語は、多いが、これを間違えると、構文全体に影響を及ぼす。そこで、接続詞の直前に、コンマがある場合は、語の接続となる場合は少ないので、”コンマ付きの接続詞は、語の等位関係を形成しない”ということとした。そして、次のような規則も加えた。

・動詞句の処理中、（スタック１を見ることにより判る）接続詞の後の文が、動詞句を生成するなら、動詞句の並列とする。

次の例文に、この規則を適用する。

We will examine recursive programming in this
動詞句１
chapter, and construct the recursive style.
動詞句２

and の直前の語、chapter の品詞は、名詞であり、直後の語、construct は、動詞、名詞の２つの品詞を持つ。しかし、and の前にコンマがあり、後ろの文は、動詞句を生成するので、and は、名詞句の並列を生成せず、construct の品詞は動詞となり、動詞句の等位関係を生成する。

３．４．その他の規則

（１）動詞句の並列の範囲

助動詞を含む動詞句と並列関係にある動詞句に、助動詞がかかるかどうかの判断は、動詞の時制を見ることによってなされる。従って、

・２つ目以降の動詞句を生成する動詞の全ての時制が、助動詞の後の動詞の時制と一致する場合は、助動詞は、全ての動詞句の動詞にかかる。それ以外は、助動詞は、直後の動詞のみにかかる。

という規則を加えた。

例１．

The way in which a data structure can be protected, initialized, accessed, and finally cleaned up are explained.

動詞句を生成している動詞の時制は、すべて過去分子なので、can be は、等位関係にあるすべての動詞句にかかる。

例２．

Predicate length was defined in sec-5.5, and

computes the number of items in a list.

接続詞の後の動詞句の動詞 computes の時制は、過去分子ではないので、was は、defined のみにかかる。

（2）名詞句の並列のための追加規則

名詞句の並列を調べる場合、まず、［規則1］の1を適用させ、接続詞の後の文が文等を生成しないときのみ、名詞句の並列の解析にいっていた。しかし、次のような文の場合には問題が起こる。

We have seen the speed and reliability of systems improve dramatically.

ここで、and 以下の文は、improve を動詞とする文と解析でき、上の文は、2つの文の並列と解析されてしまう。本システムでは、動詞は、Hornby のパープパターンによって、分類されている。そこで、次のような規則を加えた。

・動詞が、パープパターン18、
　動詞＋目的語＋原形不定詞句
又は、パープパターン24、
　動詞＋目的語＋過去分子句
を含む場合は、名詞句＋原形不定詞句（又は、過去分子句）の解析を優先させる。

上の例文に、この規則を適用させると、動詞 see は、パープパターン18を含むので、the speed の後の等位関係の処理では、名詞句の並列を先に調べ、残りの文で、原形不定詞句を生成するかどうかをみる。すると、これは、成立するので、seen のパターンは、18となり、and は、名詞の等位関係を形成する。

We have seen the speed and reliability of systems improve dramatically.

上の規則が成立しない場合は、これまでと同様、［規則1］が、順に適用される。

4．ギャップの扱い

本システムでは、等位接続詞によって繋がれた句は、全体を1つの句として扱われるため、以下のような解析結果が得られる。

例.

The calculator does the operation and prints
　主語　　等位関係にある2つの動詞句
out an answer.
全体を1つの動詞句とする

ls lists and generates statistics for files.
全体を1つの動詞とする

このため、等位関係にある句、語のどちらかが、ギャップを含むとはみなされない。

しかし、次の文のように文の並列において、動詞が省略されている場合には、適切なギャップの処理が必要とされる。

The special macro ¥* stands for the target name with suffix deleted, ¥@ for the full target
　special macro 省略　^　^ stands 省略
name, ¥< for the complete list of prerequisites,
　^　^ special macro , stands 省略
and ¥? for the list of prerequisites that are
　^　^ special macro , stands 省略
out of date.

上の例では、¥@ 等の前のギャップ（special macro）は、構文全体にそれ程影響を与えないが、動詞の省略を考えず、¥@ 以下を名詞句の並列として処理してしまうと、解析結果は、全く異なってしまう。そこで、よく現れる上のような動詞のギャップを扱うために、次のような規則を加えた。

・第1文が、
　主語＋動詞＋前置詞句1
で、それに続いて、名詞句nが並列に並んでいて、名詞句nの形態がそれぞれ、

　　名詞句
　　―――――
名詞句　　前置詞句n

となっており、前置詞句1を生成する前置詞と、前置詞句nを生成する前置詞が、同じ、又は、対になっている時は、動詞が省略された文の並列とみなす。

それ以外は、名詞句nは、前置詞句1を生成する名詞句と並列となる。

5．結果と、考察

以上のような方針で、約200例文の等位接続詞を含む文に対する解析を行った結果の正解率は、約90%であった。ここで、失敗した文のいくつかについて、その原因を述べる。

例１．

They assign to rule order, fact order and
名詞句　名詞句
quiery order.
名詞句

この文は、明らかに、上のように名詞句の並列として解析すべきである。しかし、rule は、動詞と名詞の２つの品詞を持つ。to の後の多品詞語に対しては、動詞が最優先されるという規則が、名詞句の解析の前に適用されるので、並列の処理を行う前に、rule の品詞は、動詞と決められ正しい解析はなされない。このためには、to-不定詞句の処理に入る前に、適切な制限規則を入れなければならないだろう。

例２．

(1)The first set of three characters refers to file access permissions for the user, (2)the next set for the user group and (3)the last set for all others.

これは、ギャップを含む文の並列である。（２）は、名詞句＋前置詞句の形をしているが、実際は、

the next set of three characters refers to file access permissions for the user group

であり、（３）も同様である。この例には、前に述べた簡単な規則では不十分である。（２）、（３）の名詞句が、（１）の主語と同じようであり、前置詞句が、（１）の最後の前置詞句と同じようであることから判断できるが、”同じよう”であるものを見つけるための範囲を決めることは、様々な現れ方をするので非常に困難である。

又、今まで述べた規則にあてはまらないために起こる失敗もある。

例３．

-b ignores trailing blanks and other string
名詞句　名詞句
of blanks to compare equal.

上の文では、下線のような２つの名詞句の並列である。しかし、接続詞の後の名詞句のみが、名詞句＋前置詞句となっているため、of blanks からなる前置詞句は、接続詞の前後の２つの名詞句にかかると解析されてしまう。

これらの失敗例に対しては、より多くの事例を集めて、適切な処理を行わなければならない。特に、例２のような場合には、現れやすいギャップの型を事例を集めることにより、選択し、文法規則として登録すればよいだろう。

又、今回の解析は、”見かけ上同じものは、等位関係になり易い”ということに重点をおいているが、よりよい結果を得るためには、意味的な要素をもっと考慮すればよい。例えば、上の例３のような場合は、接続詞の前後の２つの名詞句の意味と、of の後の名詞句の意味を見れば、of 以下が、どこまでかかるかを判断することができる。

しかし、多義語の意味は、文全体の解析結果を見なければ判断できない場合が多いし、逆に、個々の句の解析ができていなければ、文全体の解析も正しく行われない。又、意味の定義についてもまだ確立されたものはない。従って、解析過程で、局所的に決定できる意味から、より活用するようにすれば、よりよい解析結果が得られるだろう。

従って、今後は、より多くの例文の解析をとおして、”同じような形”の定義をより充実させるとともに、効率良い意味の決定をも考慮したよりよいシステムをめざしていこうと思う。

[謝辞]

本研究を行うにあたり御指導、御助言を頂いた成蹊大学名誉教授、和田　弘氏に感謝致します。

[参考文献]

(1) Lynette Hirschman : Conjunction in Meta-REstriction Grammer, Journal of Logic Programming, Vol. 3, No.4, pp.299-328, 1986
(2) W.A.Woods : An Experimental Parsing System for Trasition Network Grammers, in Natural Language Processing ,Rustin, R.(ed), pp 111-154, Algorithmics Press (1973)
(3) Veronica Dahl, Michael C.McCord : Treating Coordination in Logic Grammars, American Journal of Computational Linguistics, Vol.9, No.2, pp.69-91 (1983)
(4) 徳永　健伸、岩山　真、田中　穂積 : 論理文法におけるギャップの扱い、情報処理学会論文誌、Vol.32, No.11, pp.1355-1365 (1991)
(5) 長尾　真、辻井　潤一、田中　伸佳、石川　雅彦 科学技術論文における並列句とその解析、情報処理学会、自然言語処理研究会、36-4 (1983)
(6) 渡辺　藤一 : 前置詞・接続詞・間投詞、現代英文法講座　第５巻、研究社 (1958)