# 意味理解のための日本語構文解析*

## —— 係り受け関係の表示 ——

久原 健一**
田辺 利文***
吉村 賢治****
首藤 公昭****

# Syntactic Analysis for Japanese Language Understanding

## — Displaying Dependency Structure —

Ken-ichi KUHARA, Toshifumi TANABE, Kenji YOSHIMURA and Kosho SHUDO

In Natural Language Processing, there are lots of syntactic trees corresponding to an input sentence. It is important how to choose the correct one among these syntactic trees. In general, the result of syntactic analysis is ordinarily described as a list structure, which sometimes makes intuitive human recognition harder. This paper presents a syntactic analyzer which displays Dependency Structures in Japanese sentences to make human recognition easier. We also suggest a new framework for the syntactic analysis, which is designed to be a base of forthcoming semantical analysis of Japanese sentences.

***Key Words***: Natural Language Processing, Syntactic Analysis, Dependency Structure

### 1. はじめに

高精度の仮名漢字変換システムや大規模な検索エンジンなどの登場にみられるように，計算機上での日本語処理技術は飛躍的な進歩を遂げつつある．しかし，言語の意味理解に基づく知的な処理を計算機に行なわせるためには，自然言語の持つ膨大な曖昧さを解消することが必要条件であり，曖昧さの解消が自然言語処理における問題点の１つにもなっている．

自然言語処理における一般的な処理の順序は，最初に文の構成単位を認定する分かち書き処理（形態素解析），続けて，文の構成単位の係り受け関係を認定する構文解析が行なわれる．構文解析においても曖昧さの問題は存在し，処理結果の質にも大きく影響することから，複数の構文解析結果（構文構造）から最適な構文構造を正しく選択することが重要となる．構文解析システムの構築段階においては，構文構造が正しいかどうかの判定を人間が行なう場合，構文解析システムが出力する構文構造中の係り受け関係が直感的に分かりやすいことが望まれる．しかし一般的に，構文構造はリスト構造で表示されている場合が多く，これは必ずしも人間にとって都合のいいことではない．

本論文では，論文(1)における形式で分かち書きされた日本語文を入力として構文解析を行い，入力文に対する構文構造を見やすく２次元的にディスプレイ上に出力するプログラムについて述べる．このプログラムにより，複数の構文構造から最適な構文構造を選択する作業が行ないやすくなり構文解析システムの構築がよりスムーズ

* 平成15年５月31日受付
** 電子工学専攻博士課程前期
*** 電子情報工学科
**** 工学研究科情報・制御システム工学専攻

に行なうことができる．

## 2. 係り受け構造と k-marker

### 2.1 拡張文節

本研究は，論文[2]で示した拡張文節を基本としている．拡張文節とは，橋本文法でいう文節の概念を意味処理の観点から拡張したものである．我々は，拡張文節内部の文法構造については遷移ネットワーク（正規文法）による体系的なルール化を行なっている[2]～[4]．以下では，拡張文節を単に文節と呼ぶ．

### 2.2 k-marker

日本語文は，基本的に文節からなる有限列とみなすことができるが，文節の並ぶ順序には任意性が強い．そこで，語順の制約を記述するのに適した CFG を文法モデルとするのではなく，文節（の自立部）の間に観測される「係り受け」の関係のパタンによって，日本語文の構文構造を捉えることにする．論文[5]では，日本語の係り受け構造を記述するための形式として k-marker および dominant を次の(1)，(2)に示すように定義している．

(1) u が文節の e の f.d.であれば，u は e の k-marker である．このとき，u の dominant は u 自身である．
(DOM(u)＝u)

(2) $u_1, u_2$ が，それぞれ，文節の２つの列 $E_1=e_1^1e_2^1e_3^1\ldots e_m^1$, $E_2=e_1^2e_2^2e_3^2\ldots e_n^2$ の k-marker であり，DOM（$u_1$）を f.d.として持つ文節 $e_m^1$ が，DOM（$u_2$）を f.d.として持つ文節 $e_n^2$ に関係 $\alpha$ で係るならば，$[u_1u_2]_\alpha$ は，文節の列 $E_1E_2$ の k-marker である．このとき，$[u_1u_2]_\alpha$ の dominant は $u_2$ の dominant に等しい．
(DOM（DOM($[u_1u_2]_\alpha$)＝DOM($u_2$))

但し，f.d.（functional descriptor）とは文節の解析結果を表す記号のことであり，dominant は言語学でいう head（主辞）に相当する．依存関係の種類を特に指定する必要が無い場合には $\alpha$ は省略される．

通常，日本語文の係り受け構造を規定する方法として，１）係り受けに交差がないこと，２）文末以外の各文節が後続の唯一の文節に係ること，３）各文節は，同一の資格で係りと受けの関係に寄与しなければならないこと，などの制約事項を列挙することが行われるが，文節列に対して１)～３)が満たされることと，その k-marker が存在することとは等価であり，ここでは k-marker の考え方を採用する．x, y が共に k-marker であるとき，[x y] は，DOM(x) が DOM(y) に係ることを意味する k-marker である．

### 2.3 文節の判定

本論文における文節は，１個以上の自立語に０個以上の付属語が後接したものである．句読点は，自立語でも付属語でもないものとし，文の構成要素としてみなしていない．関係表現，助述表現，接尾語的表現は付属語としている．関係表現とは概念語間の関係を指示する格助詞，接続助詞，およびそれらに相当する連語であり，助述表現とは，時刻，相，話者の態度，判断否定など，広義の様相情報を与える助動詞，終助詞およびそれらに相当する連語であり，接尾語的表現とは，複合表現の造語成分のうち，比較的多数の表現に連接でき，形態・意味上の機能が明確な表現をいう．関係表現，助述表現，接尾語的表現の具体的例は論文[1]を参照されたい．

### 2.4 カテゴリークラス

カテゴリークラスとは関係表現，助述表現，接尾語的表現に付与された意味分類であり，主に文節内での連接機能の差異によって通常の橋本文法における品詞を精密化したものである．本研究室において開発した付属語辞書中での４桁のコード，見出し語の分類コードである．

### 2.5 係り受け判定

ある文節 $e_i$ が後続の文節 $e_j$ に係るか否かの判定条件を定義する．f.d. は文節の解析結果を表すものであるが，解析するとは文節自身が受けの場合は何型であり，後続のどれかの文節に係る場合は何型にかかるかというものである．

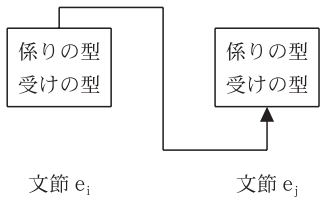


図1

#### 2.5.1 受けの文節の型

文節の型は N 型（名詞型）と P 型（述語型）の２種類に分類している．文節中で最も末尾に近い自立語，または文節中で最も末尾に近い接尾語的表現により受けの文節の型が決まる．副詞，連体詞は係り受け関係において受けの機能がないものとみなし，受けの文節の型は無い．文節中の最も末尾に近い自立語が体言であれば N 型，用言であるなら P 型とする．接尾語的表現の場合はカテゴリークラスにおいて左から３番目の英文字が n であれば N 型，p であれば P 型とする．例えば，論文[1]の形式で表現される形態素解析済み文字列が「氾濫(M20)/する(Snp2)/とは(RPP1)/」である場合，M20は名詞，Snp2は接尾語的表現，RPP1は関係表現であり，この場合は受けの文節の型は P 型となる．

#### 2.5.2 係りの文節の型

末尾が関係表現である文節はその関係表現のカテゴリークラスにおける左から３番目の英文字が示す型を受けと

する後続の文節に係る．末尾が用言，活用可能な助述表現，活用可能な接尾語的表現の文節はそれらの活用形により，連体形であれば受けがN型の文節，連用形であれば受けがP型の文節に係る．また，副詞は後続のP型文節，連体詞は後続のN型文節に係る．例えば，論文[1]の形式で表現される形態素解析済み文節が「IBM(M13)/の(RNN1)」の場合，M13は名詞，RNN1は関係表現であるが，この場合は後続の受けがN型の文節に係る．

## 3. パーシング・アルゴリズム

### 3.1 データ構造及び，パーステーブル

入力された文節の列における各部分列に対する解析結果は，〈x y〉形式のデータ構造で与えられる．ここでxは文節列の係り受け構造に関する記述（k-marker），yは文節列中の特定の文節のf.d.である．

入力文節列を $e_1e_2e_3\ldots e_n$ とするとき，解析部分列 $e_pe_{p+1}\ldots e_r(1\leqq p\leqq r\leqq n)$ に対する解析結果は，パーステーブル T(p, r)に登録される．従って，入力全体に対する解析結果はT(1, n)のデータ構造で与えられる．

### 3.2 アルゴリズム

対応するパーステーブルに係り受け構造(k-marker)を格納する．文頭から文節をleft-to-rightに1個ずつ読み込み，解析を行い，対応するパーステーブルにk-markerを格納する．left-to-rightで読み込んだ文節は，その時点で受けの文節として固定し，すでに読み込まれている文節をright-to-leftに走査し，係りの資格をもつ文節を検出する．検出できた場合は係り受け関係が成立していることを意味するため，k-markerを統合したものをパーステーブルに格納する．その後，検出された文節に統御されている句をright-to-leftに走査し，句の結合を行う．アルゴリズムの詳細については論文[5]を参照されたい．

## 4. プログラム作成環境

係り受け構造表示プログラムを作成するにあたり，今回はMicrosoft Visual C＋＋を用いた．

Microsoft Visual C＋＋では，実行形態に応じて，DOS上で実行させる形式(Win32 Console Application)とWindowsアプリケーションとして実行させる形式がある．今回は出来る限り美しく構造を表示するため，キャラクタ文字を使わずに済むWindowsアプリケーション作成を選択した．これらの方法の詳細については，文献[6]～[8]等を参照されたい．

## 5. 実行例

まず起動画面を図2に示す．論文[1]の形式における形態素解析済み文字列をエディットボックスに入力する．エディットボックスに形態素解析済み文字列を入力した状態で解析ボタンを押すと係り受け構造が表示される．図3は日本語文「ここで飛鳥田らしさを出そうという考えがあったのでしょう。」に対する拡張文節分かち書き結果を入力した時の係り受け関係を示している．出力グループボックス内の3つのラジオボタンのどれか1つをクリックするとそれぞれの解析結果を表示できる．図4は入力E－文節ラジオボタンをクリックしたときの画面であり，入力された形態素列をE－文節に構成した結果とf.d.を示している．図5はパーステーブル内のk-markerを示している．


**図2**　起動画面

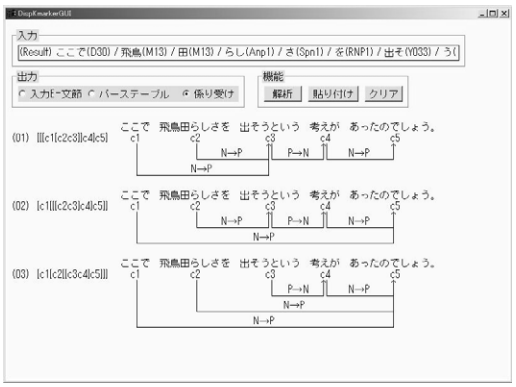


**図3**　係り受け関係表示

## 6. おわりに

本論文では，拡張文節で分かち書きされた日本語文を入力とした構文解析法を述べ，構文構造を見やすく2次元的にディスプレイ上に出力するプログラムについて述べた．拡張文節を組み込んだ言語モデルを使用することで構文解析以降の処理の質の向上が期待できる．また，

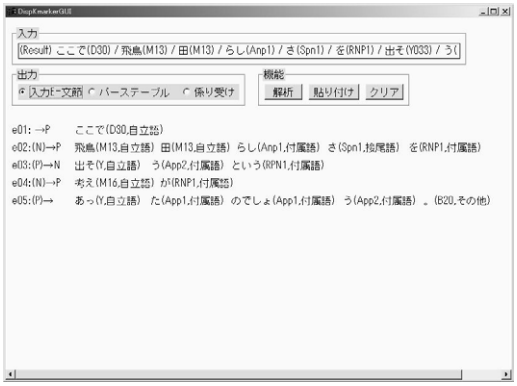


**図4**　入力Ｅ－文節表示


**図5**　パーステーブル表示

拡張文節での分かち書き結果は，一般的な文節での分かち書き結果と比べて文節の数が少なくなることから，同等の構文解析アルゴリズムを用いる場合でも計算時間の短縮が期待できる．

本論文で述べたプログラムにおける入力は，予め拡張文節で分かち書きされたもののみであり，分かち書きされていない文に対しても解析できるよう拡張文節分かち書きシステムと統合するなど改良する必要がある．

係り受けの判定には，係りの文節が「犬が」で，受けの文節が「運転する」，また係りの文節が「速い」，受けの文節が「本」などのような意味的に不自然な場合でも本論文では係り受けの条件を満たすものとしているが，係り受けの条件に，自立語の意味を組み込むことで，意味レベルでの曖昧さの解消が期待できる[9]．また，これまでのモデルでは，曖昧さが残っていても，構文解析結果相互間に優先順位を与えていなかった．そのため，構文構造に何らかの優先順位を設ける必要がある．優先順位を設ける方法としては，例えば，統計モデルを使う方法などがある[9]．

またアプリケーションの追加機能として係り受け構造の画像ファイルとしての保存機能の追加や，プリントアウト機能などがあげられる．これらの追加により，係り受け構造のデータの柔軟な活用が可能になる．

## 参　考　文　献

（1）添島創，田辺利文，吉村賢治，首藤公昭，「日本語文分かち書きのための新しい枠組み」福岡大学工学集報第70号，2003. 3

（2）首藤公昭，楢原斗志子，吉田　将，「日本語の機械処理のための文節構造モデル」，電子通信学会論文誌（D），J62-D，12，1979

（3）首藤公昭，「文節の構造と文解析」，電気四学会連合大会講演論文集，5，1979

（4）Shudo, K., Narahara, T., Yoshida, S.: “Morphological Aspect of Japanese Language Processing”, Proceedings of the 8th COLING, 1980

（5）首藤公昭，楢原斗志子，津田健蔵，「意味理解を目的とした日本語の構文解析アルゴリズム」，福岡大学工学集報第28号，1982. 3

（6）Herbert Schildt,（柏原正三訳），「c/c＋＋プログラマのための Windows95プログラミング」，翔栄社

（7）粂井康孝，「猫でもわかるプログラミング」http://www.kumei.ne.jp/c_lang/（Windows SDK 編第一部～第三部）

（8）Microsoft Visual C＋＋ マニュアル

（9）田辺利文，冨浦洋一，日高達，「係り受け文脈自由文法とその日本語への適用」，情報処理学会論文誌第41巻第１号