計算言語学 25－4
（1981. 2. 6）

仮名漢字変換のための文法解析

大河内正明・藤崎哲之助・諸橋正幸
（日本アイ・ビー・エム株式会社 東京サイエンティフィック・センター）

概要

日本語文書処理システムの研究の一環として、仮名漢字変換方式の日本語エディタを開発したので、これを文法情報の取扱いを中心にして紹介する。この仮名漢字変換の入力文はほぼ文節の単位の柔軟な分かち書きであり、字種指定も併用できる。同音異義語からの選択は、文法規則との整合性と単語の頻度情報に基いて決められる。形態素解析は、複数の接続条件を「接続ベクトル」と呼ぶ固定長のビット列で統一的に扱い、バック・トラッキングを必要としない簡潔な処理になっている。また、付属語の解析を文節の後端から逆方向にまとめて行ったあと自立語を前端から引き当てる方式のため、処理が効率的になるだけでなく、辞書にない外来語の推定や未登録語の品詞活用推定も容易になるという特徴がある。接続ベクトルによる解析法は、一般の構文解析の前処理にも適している。

1. はじめに

日本語文章の入力法として、仮名で入力したものを計算機で漢字仮名混り表記に変換するいわゆる「仮名漢字変換」は、一般の人にとって使い易いため、近年着目されてきている。

仮名漢字変換のための入力方式は、べた書き[1]、字種指定[2]、分かち書きの3つが代表的である。べた書きは、打鍵者の負担が最も少いが、辞書引き単位の抽出が難しくあいまいさも増大し易い（例えば、「キノウハイシャニイッタ」は、「きのうは 医者に行った」とも「きのう歯医者に行った」とも解釈しうる）。字種指定は、辞書引き単位が抽出し易く、漢字表記するか否か一定していない表現に対しても変換の意図を明示できるが、文章を考えながら入力する場合などに漢字の送り仮名の仕方で迷い易い。分かち書きは、打鍵し易い空白キーで文章を区切るので特殊な仮名鍵盤を必要としない。分かち書き単位の取り方によって、文節分かち書き、自立語付属語分かち書き、単語分かち書きなどに細分されるが、あまり細かい分かち書きは煩雑であり、柔軟性のないものは打鍵者への負担が大きい。

筆者らは、日本語文書の作成・編集・検索等を一貫して処理することを目的として、日本語文書処理システム「ことだま」[8]の研究開発を進めており、その一環として、仮名漢字変換方式のエディタ（実験システム）を開発した。この仮名漢字変換は、文節分かち書きを基礎にしているが、文節の概念を拡張して分かち書きを柔軟にする一方、字種指定も併用できるようにしてある。

一般に仮名漢字変換においては、同音異義語の取扱いが大きな課題であり、
①使用頻度などによる優先順序づけ
②文法との整合性（特に文節内）
③係り受け関係や意味処理
などによって妥当なものを選択することが試みられてきているが、筆者らは、③は不特定の対象に適用するにはまだ解決すべき問題が多いと考え、①と②の面で、従来のアプローチを改善する方針をとった。特に②のための文法解析は、複数の接続可能性がバック・トラッキングなしに簡潔に扱えるとか、未登録語の品詞活用の推定ができるなど、従来のシステムにない特徴がある。

本稿では、このエディタの仮名漢字変換の特徴を概観したのち、その文法情報の取扱いに関して報告する。

2. 日本語エディタの構成と特徴

「ことだま」の日本語エディタは、図1のように構成されており、片仮名入力文を仮名漢字変換によって漢字仮名混り文に変換（ローマ字入力も可能）したり、既存の文章ファイルを修正するのに使われる。このエディタの仮名漢字変換機能には以下の特徴がある。

① 柔軟な分かち書き

入力文の分かち書きの仕方は、文節単位を原則とするが、柔軟性がある。

文節は、文法的には、

<自立語> [<付属語>]*

（[a]* は a が0個以上続くこと）

である（<付属語>は助詞と助動詞、<自立語>は他の単語）が、本エディタで許される分かち書き単位は、次のような「拡張された文節」（本稿では単に「文節」と呼ぶ）である。

[<連体詞>]<自立語>[<拡張された付属語>]*

ただし、<連体詞>は、

アル, コノ, ソノ, アノ, ドノ, コンナ, …

などのコソアド系を主とする連体詞のみである。<自立語>と<拡張された付属語>の内容については後述する。

例えば、

「そんな危険なことをしないで下さい」

という文は、本来の文節分かち書き

ソンナ キケンナ コトヲ シナイデ クダサイ

だけでなく、

ソンナ キケンナコトヲ シナイデ クダサイ
ソンナキケンナコトヲ シナイデクダサイ
ソンナキケンナコトヲ シナイデクダサイ

などの分かち書きでも扱いうる。

② 文法情報と頻度情報による同音異義語処理

日本語は同音異義語が多いが、入力文は文節単位で文法規則との整合をとりながら分析されるので、同音異義語でも文法に合わないものは自動的に除かれる。例えば、「危険」と「棄権」は同音異義語であるが、品詞・活用の

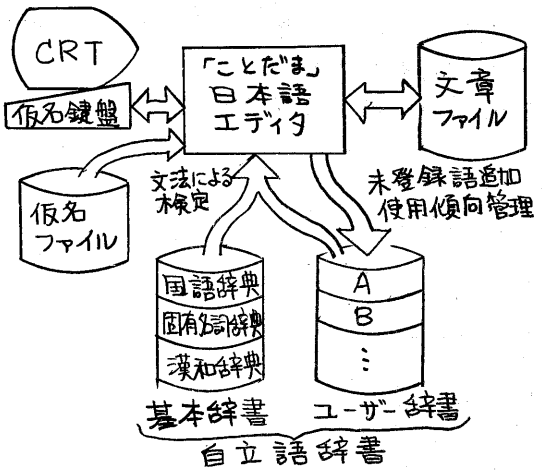


図1 「ことだま」日本語エディタの構成

違いにより、変換候補が次のように限定される。

キケンナ ⟶ 危険な
キケンスル ⟶ 棄権する
キケンヲ ⟶ 危険を、棄権を

一方、「科学」と「化学」のように、文法的に区別できない同音異義語に対しては、ユーザー毎の単語使用頻度等によって、第一変換候補が決められる。また、同音異義語から一度選択された単語は、そのセション中は第一変換候補とされるので、同じ同音異義語選択を繰り返すことはない。

③ 未登録語の自動追加登録

自立語の辞書としては、全ユーザーに共用される「基本辞書」と、ユーザー毎の使用傾向を反映する「ユーザー辞書」がある。基本辞書に無い単語でも、当て字などの校正処理で正しく変換されると、自動的にユーザー辞書に保存されて以後の処理に反映される。しかも、その単語が使われた文脈から品詞・活用が推定され、単語とともに保存されるので、以後の同音異義語の区別にも反映される。

複合語や数詞も分かち書きせずに処理されるが、複雑な場合には再分割などの校正処理が必要となることがある。

その場合も、正しく変換された結果がユーザー辞書に登録されるので、以後、この複合語は分かち書きせずに扱えるようになる。

④ 未登録外来語・擬音語の推定

片仮名表記の外来語、擬音語は、辞書に無くてしかも字種指定で明示されてない場合でも、文脈から文法的に推定している。一般に外来語・擬音語は、

- 名詞（例：データ、ロンドン）
- サ変動詞語幹（例：コーチする、ガタガタする）
- 形容動詞語幹（例：フレッシュな）

のいずれかとして使われるので、複合語処理などによっても適当な変換候補が見つからないときは、文法解析によって上記3種のいずれかと見なせる文字列（複数あれば最短のもの）を求めて、それを未登録外来語・擬音語の候補として片仮名表記している。この片仮名表記は、適当な変換候補が無かったことの警告をも兼ねている。

3. 自立語と付属語に関する考慮点

自立語は、接続関係の観点から表1のように分類して扱っている。また、付属語は助詞、助動詞だけでなく、分かち書きが柔軟になるように拡張されている。以下に、これらの内容の特徴を述べる。

① 用言に関しては、語幹のみを自立語とし、活用語尾は付属語に含めてある。ただし、一段活用動詞の活用語尾の1字目は変化しないので、自立語に含む（例：「見る」は「見」が自立語）。

② 動詞「行く」は、他のカ行五段活用動詞と音便形が異なるので別扱いにしてある。命令形が特殊な「く（呉）れる」や、「下さい」などの「ーい」型命令形の動詞は、付属語（補助用言）に含めて扱っている。

③ 動詞（カ変、サ変を除く）の連用形は体言化することが多い（例：動き、延び）ので、辞書に名詞としてなくても、規則化して扱えるようにしてある。ただし一段活用動詞は、体言化しないもの（特に1音節のもの）が多く、

ケガ → 蹴む が

などの誤変換を避けるため、名詞と見なせるか否かによって分類してある。

④ 五段活用動詞からは、

動く（カ行五段）→ 動ける（下一段）
　　→ 動かす（サ行五段）

のように、可能動詞や他動詞、使役動詞が派生できることが多いが、これらの派生動詞は、自立語辞書に無くても扱えるように、規則化してある。

⑤ 助動詞の多くは、語幹と活用語尾を分離して、それぞれを付属語としてある（例：来るそうだ）

⑥ 汎用性の高い接尾辞は付属語として処理している（例：若さ、書き方、言い過ぎる）。

表1：自立語の分類

分類番号	品詞、活用	例
1	カ行五段活用動詞語幹	書（く）
2	同上（特殊）	行（く）
3	ガ行五段活用動詞語幹	泳（ぐ）
4	サ　〃	押（す）
5	タ　〃	立（つ）
6	ナ　〃	死（ぬ）
7	バ　〃	飛（ぶ）
8	マ　〃	進（む）
9	ラ　〃	走（る）
10	ワア　〃	思（う）
11	一段活用動詞不変化部（体言）	延（び）（る）
12	同上（非体言）	見（る）
13	サ変動詞（名詞型）語幹	実験（す）
14	〃（する型）〃	察（する）
15	〃（ずる型）〃	案（ずる）
16	カ変動詞 漢字部	来（る）
17	形容詞 語幹	高（い）
18	形容動詞 語幹	静か（だ）
19	名詞	学校
20	連体詞	あらゆる
21	副詞	いきなり
22	接続詞、感動詞	しかし

⑦ 形式名詞、補助用言、主要自立語（特に仮名書きされるもの）は付属語に含めている（例：見ることもよいが）。
⑧ 慣用句や例外的な接続関係は単一の付属語としてある（例：よさそうだ、本当かどうかが）

4. 接続条件の管理

単語間の接続条件は、「リンク」と「接続ベクトル」という概念で扱っており、具体的には、主記憶域内の付属語表と外部記憶装置内の自立語辞書によって管理されている。

リンク

各単語はその前後の接続条件に対応して「前端リンク」と「後端リンク」のリンク対を持ち、2つの単語は単語間のリンクが一致するとき接続可能になる（自立語は前端リンクを省略することもある）。品詞や用法が違っても文字列として一致する単語群は、まとめて1つの単語として扱っており、単語や単語連鎖の両端に許されるリンクは1対とは限らない。ただし、名詞に後続するとか、文節の最後になるなどの境界条件を与えて、一端でとりうるリンク群を制限すると、他端に許されるリンク群も変わる。例えば、図2の単語Wは6組のリンク対を持ち、後端リンクとして n_2 と n_3 のみが可能ならば、前端リンクとしては P_1, P_3, P_4 だけが許される。また、リンク群として P_1, P_2, P_3, P_4 のいずれも許されなければ、単語Wはその後に接続しない。

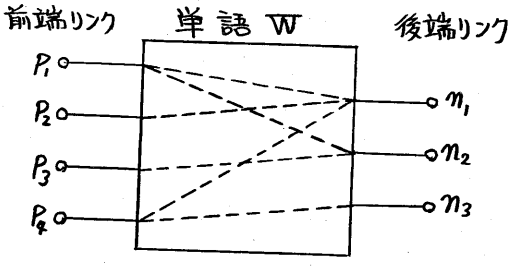


図2 単語の接続条件の概念

接続ベクトル

接続ベクトルは、形態素解析に必要な文法情報をまとめて統一的に扱うために導入した固定長のビット列であり、各ビットの1/0が、各文法情報（リンクなど）の有/無に対応している。

本エディタでは図3のようなビット構成になっており、表1の各自立語に1ビットを対応させた「品詞活用ベクトル」と、次の4種の位置情報フラッグを含んでいる（いずれもリンクの一種として扱われる）。

Eフラッグ：文節終端
Fフラッグ：付属語連鎖の前端
Jフラッグ：自立語に接続しうる付属語連鎖前端
Sフラッグ：文節前端となりうる位置

品詞活用ベクトル (22ビット)	位置情報フラッグ (4ビット)	その他のリンク (70ビット)

図3 接続ベクトルの構成

付属語連鎖の解析には接続ベクトル全体が使われ、自立語の引き当てでは、位置情報フラッグを参照しながら品詞・活用ベクトルのみが使われる。

付属語表

付属語とその接続関係は、図4の付属語表で管理している。ただし、主記憶域上では、(FZK, KJX)の共通なエントリーをまとめて探し易いように、木構造で構成している。

逆引き仮名見出し (FZK)	漢字部インデクス (KJX)	後端リンク (NXT)	前端接続ベクトル (PVTR)

図4 付属語表の構成

本エディタでは、付属語の解析を文節後端から逆方向に行うため、付属語の見出しも逆順になっており、前端リンク群は後端リンク毎に接続ベクトル

表現にまとめられている。漢字を含む付属語は、図5のように管理される。

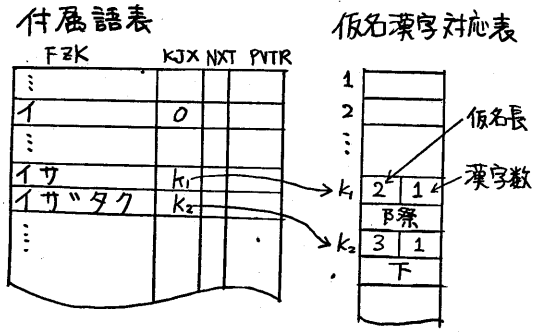


図5 漢字を含む付属語の管理

自立語辞書

自立語辞書のエントリーは、基本辞書もユーザー辞書も共通に、図6のように構成されている。品詞活用ベクトルは、例えば「満足」の場合、名詞、形容動詞語幹、サ変動詞(名詞型)語幹に対応する3ビットが1になっている。複合語も数詞も共通に管理される。その際、数詞表現は、

ダイ &カイ ⟷ 第 &回
(仮名見出し) (正書表記)

のように対応づけられて管理される。

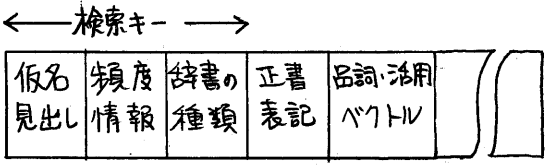


図6 自立語辞書の構成

5. 仮名漢字変換のための処理

5.1 処理の概要

仮名漢字変換は図7のように処理される。入力文に対する文法解析は、空白や字種変化によって抽出された文節の単位で行われる。

入力文節が「キケンシナイデクダサイ」の場合は図8のようになる。まず、文節後端から逆方向に、文節の最後になりうる付属語をすべて求める。この

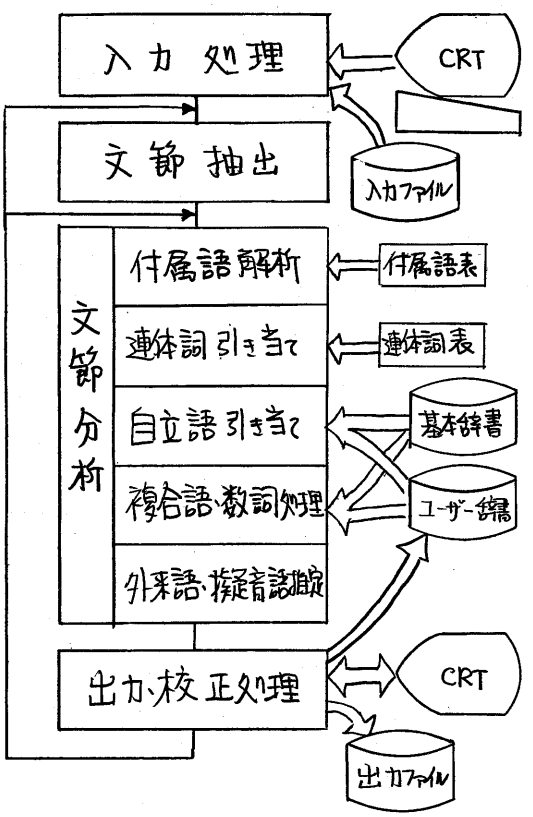


図7 仮名漢字変換処理の流れ

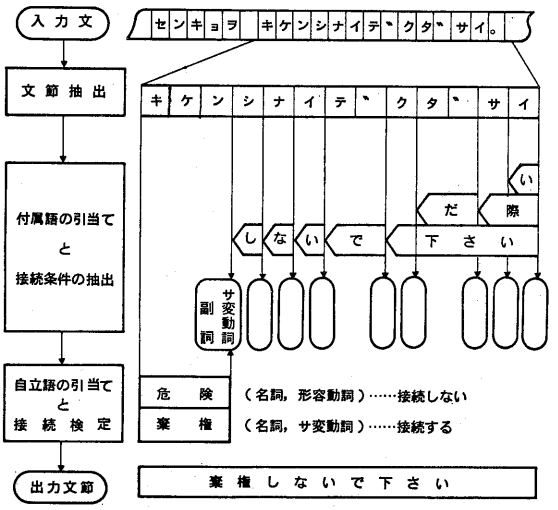


図8 文節分析の処理概念

例では、「い」、「際」、「下さい」の3つが求まり、その前端の接続条件が算出される。以後、解析済みの付属語連鎖の短いものから順に、その前に接続しうるすべての付属語を求めていく。その結果、付属語連鎖としては、最短の「い」から最長の「しないで下さい」までの8つが求まり、これらの前端と文節終端に接続条件が残る。これらの接続条件のうち、自立語と接続可能なものが、自立語の後端位置の候補になる。次に、文節前端から連体詞の引き当てが試みられるが、この例では見つかっておらず、自立語の前端位置候補は文節前端のみである。自立語境界候補を満たす自立語としては、「危険」と「棄権」が見つかっているが、後者のみがサ変動詞語幹として接続条件を満たす。以上の結果、出力文節「棄権しないで下さい」が得られる。

入力文節中に数字が含まれている場合は、図9のように処理される。

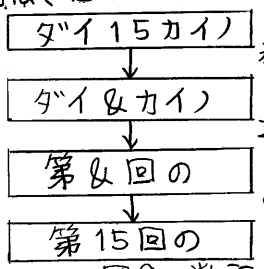


図9 数詞の処理例

付属語連鎖と整合する自立語が見つからない場合は複合語の可能性が調べられ、それでも整合するものが見つからない場合は、辞書にない外来語、擬音語が使われていると仮定して、図10のように処理される。

5.2 文節分析作業域

文節分析は、図11の作業域で行われる。文節長がLのとき、接続ベクトルA_iは付属語連鎖$I_i I_{i+1} \cdots I_L$の前端リンク群に対応する。また、付属語連鎖構成ベクトルF_iは、この付属語連鎖を構成している付属語を示すビット列であり、この第mビットが1ならば、図12の構成付属語表の第mエントリーに対応する付属語が、この付属語連鎖の要素であることを示す。漢字フラッグK_iは、F_iで示される付属語群に漢字を含むものがあることを示す1ビットのフラッグである（処理効率を上げるために導入したものであり不可欠ではない）。

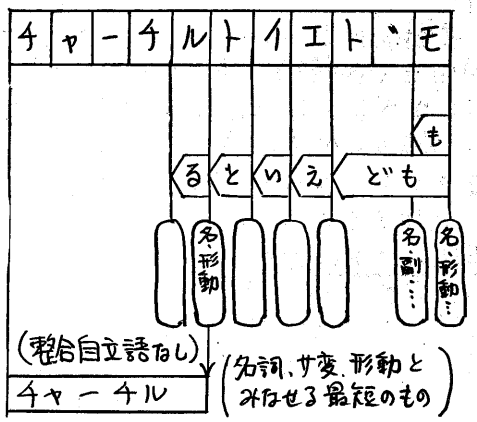
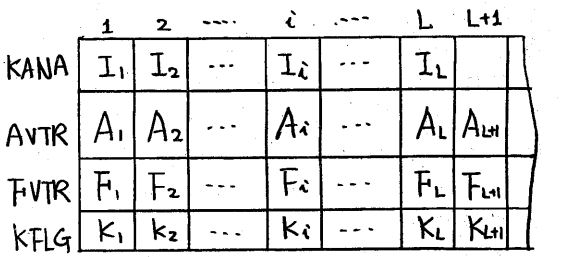


図10 未登録外来語の推定



I_i：入力仮名文節のi文字目
A_i：付属語連鎖$I_i I_{i+1} \cdots I_L$の前端接続ベクトル
F_i：〃　　の構成ベクトル
K_i：〃　　の漢字フラグ

図11 文節分析作業域

付属語の前端位置	付属語の後続位置	付属語表中の通し番号

図12 構成付属語表のエントリー

文節長が L のときの主要な処理は、A_{L+1} を文節終端接続ベクトル（文節末になりうる単語群に対応する*）に設定し、他のすべての A_i, F_i, K_i のビットを0としておいて、作業域上を左方向に分析することである。**)

接続ベクトルの処理

文字列 $I_{i-l} I_{i-l+1} \cdots I_{i-1}$ が付属語候補として見つかったとき、その（後端リンク、前端接続ベクトル）の対として（m, P）があり、接続ベクトル A_i の第 m ビットが1ならば、この付属語候補は採用され、A_{i-l} は、

$$A_{i-l} = A_{i-l} \vee P$$

とビット列の論理和演算で更新される。なお、接続ベクトル内の位置情報フラッグは、Eフラッグが文節終端接続ベクトル内で、その他のフラッグが付属語表の前端接続ベクトル内で設定されているので、上記の論理和演算で同時に処理される。

付属語連鎖構成ベクトルの処理**)

文字列 $I_{i-l} I_{i-l+1} \cdots I_{i-1}$ が第 m 番目の付属語として採用されたとき、構成付属語表の第 m エントリーが追加されるとともに、付属語連鎖構成ベクトル F_{i-l} は、F_i の内容に第 m ビット＝1 を追加したものに設定される。図8に対応する処理内容を図13に示す。

自立語の引き当て

接続ベクトル A_j 内のJフラッグが1であり、文字列 $I_1 I_2 \cdots I_{j-1}$ が自立語候補として見つかると、以下の接続検定が行われる。自立語の品詞・活用ベクトル H と、接続ベクトル A_j 内の品詞・活用ベクトル $A_j^{(h)}$ の論理積をとって、

$$H \wedge A_j^{(h)} \neq 0$$

ならば、この自立語候補は、付属語連鎖 $I_j I_{j+1} \cdots I_L$ の前端に接続する。

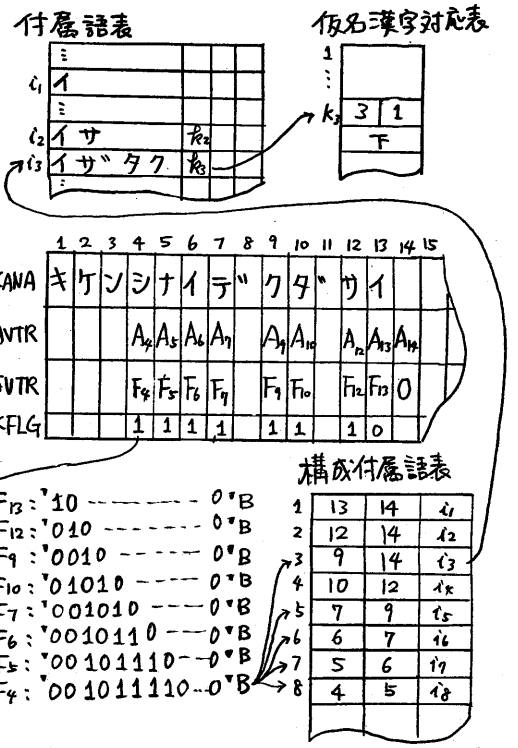


図13 付属語連鎖の構成情報の処理

構成付属語の確認**)

自立語と整合した付属語連鎖が漢字を含んでいる（漢字フラグでわかる）場合は、付属語連鎖構成ベクトル F_i によって漢字を含む付属語を探し出せる。ただし、異なる付属語群によって同一の付属語連鎖が作られる場合は、F_i によって示される付属語群の位置に重複が生じる。この場合も、付属語連鎖の左端に境界条件（引き当てた自立語の品詞・活用ベクトル）を与えて、順方向に接続関係を確認すれば、実際の構成付属語がわかる。本エディタの場合、漢字を含む付属語が識別できればよいので、確認が必要になるのは、漢字を含む付属語の位置が他の採用付属語と重なっている場合だけであり（部分的な重なりなら不要）、ほとんど生じない。

---

*) 後続文節が付属語連鎖だけからなる場合は、その前端接続ベクトルを文節終端ベクトルに使う。従って、「キケンスル」を切って「キケン　スル」と入力しても同じ変換結果を得る。

**) 漢字を含む付属語を扱わない場合は、文節分析作業域の付属語連鎖構成ベクトルと漢字フラッグは不要であり、主要な解析は接続ベクトルの処理だけになる。

6. 文法解析に関する考察

本システムの文法解析は、従来の仮名漢字変換システム[1][2][3]などと比べて、次の2点が大きな特徴である。

① 文節終端からの付属語連鎖解析

従来の文法解析は、自立語の候補を引き当ててから、それに続く付属語連鎖を順方向に解析して、候補の妥当性を検査しているものが多いが、本システムでは、主記憶域内で高速処理できる付属語連鎖解析を先に行って、自立語候補の境界と接続条件をしぼってから、外部記憶装置のアクセスを必要とする自立語を引き当てている。しかも付属語の引き当ては、境界条件の明確な文節終端から、それと整合するものだけをまとめて求めるので、余分な引き当ては少くてすむ。また、付属語連鎖前端の接続条件が先に求まるので、辞書にない外来語の推定や未登録語の品詞活用の推定も容易になる。

② 接続ベクトルによる簡潔な処理

従来は単語間の接続検定を数値コードの一対一比較で行うリスト処理を基礎にしていたため、接続関係に複数の可能性がある場合は、バック・トラッキングなどの複雑なスタック操作が必要であった。本システムでは、単語間の接続条件を「接続ベクトル」という固定長のビット列で統一的に扱っているため、主要な処理がビット列の論理演算として簡潔に表現でき、一次元の配列として構成された作業域上を一方向に解析するだけで、すべての接続可能性が分析できる。つまり複数の接続可能性(有限オートマトンの複数の状態に対応)を、同じ深さ(単語連鎖の文字列が同じ長さ)に関して、1つの接続ベクトルで同時に基めている。初期値も複数条件に対応するものでよい。処理量は解析対象の文字列の長さLに比例する程度ですむ。

本システムの文法規則は有限決定オートマトンとしても実現できるが、状態の数が増え状態の意味がわかりにくくなるだけでなく、必要記憶域も増大する。接続ベクトルの導入による分析は、一般の順方向の形態素解析にも適用できる。また、構文解析の前処理に用いて処理効率の向上が望める(特に単語自体の抽出が難しく同音異義語の多い日本語文の解析に適している)。

7. おわりに

日本語文書処理システム「ことだま」の研究の一環として開発した日本語エディタの仮名漢字変換について、文法情報の取扱いを中心にして報告した。本稿の内容は、報告書[6]を基礎にしているが、その後の改善(一部開発中)を反映している。本稿で述べられなかった「ことだま」の他の側面や文法規則については、別の機会に報告したい。

なお、「ことだま」の辞書作成においては、三省堂の新明解国語辞典を利用させて頂いた。また、本研究の実施に当っては、辞書の整備等に協力頂いた戸沢義夫、大深悦子両氏を初め、多数の関係者の助言、協力を得たことを感謝します。

参考文献

[1] 牧野・木澤「べた書き文の仮名漢字変換システムとその同音語処理」情報学会論文誌, 22-1, pp. 59-67 (1981)

[2] 河田・天野・武田・森「ミニコンピュータを用いたカナ漢字変換システム」電子通信学会技報 PRL76-45 (1976)

[3] 稲永・小西「カナ文字文の構文処理のための辞書について」電子通信学会技報 EC76-45 (1976)

[4] 佐藤・田中「日本語の構文解析」情報処理, 20-10, pp. 865-872 (1979)

[5] 長尾・辻井・他「計算機による日本語文章の解析に関する研究」文部省科研費特定研究(1), 昭和53年度報告書 (1979)

[6] 大河内・藤崎・諸橋・戸沢「仮名漢字変換のための文法情報の管理と処理」, IBM TSC レポート N:G318-1510 (1979)

[7] 諸橋・藤崎・大河内・戸沢「『ことだま』の日本語エディタ」IBM TSC レポート N:G318-1511 (1979)

[8] 藤崎・大河内・諸橋・戸沢「日本語文書処理システム『ことだま』—概念と設計思想—」IBM TSC レポート N:G318-1512 (1980)