

マルチタスク形式によるカナ漢字変換システムについて

大山 裕 宮井 均 首藤 正道
(日本電気株式会社 C & C システム研究所)

1. はじめに

カナ漢字変換による日本語入力方式は、少ないキーで多種の漢字を入力できること、タッチ法によるため比較的高速な入力が可能であること、等の理由で、現在ワードプロセッサ等の日本語情報処理機器における入力法の主流となっている。今後も、エンドユーザ指向のシステムへの需要が高まると思われるが、音声による入力が有効なカナ漢字変換は、エンドユーザ向きの日本語入力方式であると考えられる。

しかし、現時点では、カナ漢字変換方式にもいくつかの問題点が残されている。例えば、

- 入力単位に強い制約がある
(分かち書きの強要)
- カナと漢字が1対1に対応しない
(同音語の選択)

等である。また、カナ漢字変換方式自身の問題ではないものの、

- キーボードによる入力
(慣れない人に使いづらい)

等の問題もあり、現在のところ、利用者は多大な負担を強いられている。

筆者らは、これらの問題を解決し、マンマシンインタフェースの良さを重視したカナ漢字変換システムの実現を目指しているが、このための実験用ベースシステムとして、マルチタスク形式によるカナ漢字変換システムを試作した。本システムは、現在も機能の強化・拡張を続けているが、本稿では、現バージョンのシステムについて述べる。

2. 概要および特徴

本システムは、文節単位に分かち書きされたカナ文字列を入力の基本として、これを総当たり方式でカナ漢字変換し、漢字かな混じり文字列または解析木に相当する単語列を出力する。

本システムは、次に掲げる特徴を有する。

〔Ⅰ〕機能別タスクの積上げ形式

本システムは、各種機能別タスクの積上げによって構成されるため、各種実験や機能の追加・改造に柔軟に対応することができる。

〔Ⅱ〕並列処理指向

従来のカナ漢字変換は、変換に要する一連の手順をプログラムによって制御しているため、同時に2つ以上の辞書を検索したり、辞書検索を行なう一方で接続検定を行なう、といった並列性を求めることはできなかった。

本システムにおいては、各機能がタスク形式に構成されるため、いくつかのタスクが同時に処理を行なうことが可能であり、各機能を効率よく働かせることができる。

〔Ⅲ〕非バックトラック形

本来、カナ漢字変換システムとは、入力カナ文字列を文法的に解析して、解析木(または解析木の一部)を生成し、そのルートからリーフまでの各ノードに存在する単語の表記部を結合して、漢字かな混じり文字列を作成し、これを出力するシステムである。

従来のシステムにおいては、図1-aに示すように、解析をdepth-first方式で求めているため、途中結果をスタックに貯め必要に応じて取り出すバックトラック処理が必要であった。

本システムでは、図1-bに示すように、あるノードにおいて生成可能な枝をすべて生成しながら解析木を作っていく方式をとるため、複雑なバックトラック処理を必要としない。

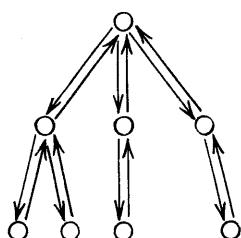


図1-a

従来の解析手順

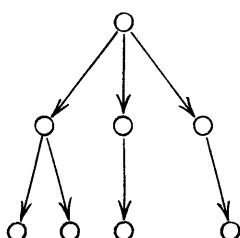


図1-b

本システムの解析手順

〔Ⅳ〕文法記述

従来の方式では、日本語の文法を、解析途中における次に検索すべき辞書を表わす情報と、単語間の接続可否を示す情報に分離し、前者をプログラムに、後者をテーブルとして実現していた。本システムでは、前者を状態遷移ネットワーク形式のテーブルで記述しているため、文法の容易な変更が可能である。

3. 構成

本システムは、図2に示すように、カナ文字列（またはローマ字列、単音節列）を入力するための入力系タスク群と、入力されたカナ文字列から解析木を生成し、漢字かな混じり文字列を作り出す処理系タスク群と、漢字かな混じり文字列を表示・出力する出力系タスク群、および各種ユーティリティにより構成される。これらについて、処理系、ユーティリティ、入力系、出力系の順に述べる。

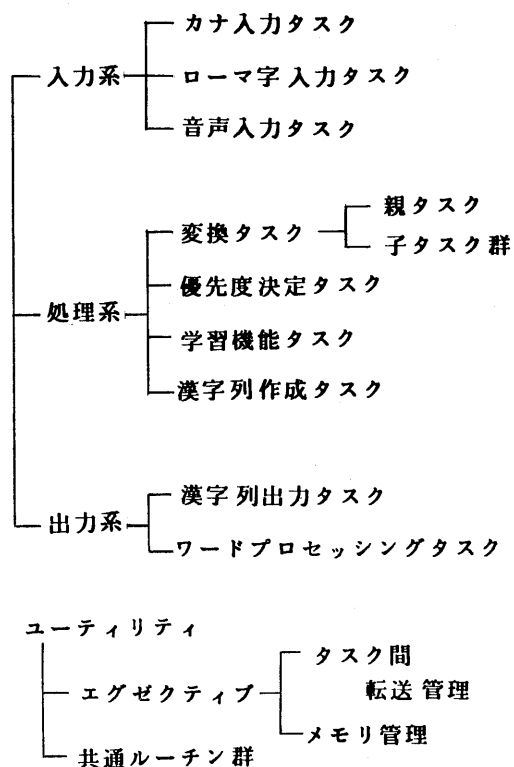


図2 システム構成

4. 処理系タスク群

4.1 変換タスク

変換タスクは、先導タスクよりカナ文字列を入力し、解析結果を先導タスクに戻す役割を果たす。

変換タスクは、先導タスクによるリクエストによって起動され、親タスクと子タスク群の間で、辞書やテーブル類を参照しながら変換途中結果をやり取りすることによって解析をすすめていく。変換タスクの処理におけるデータの流れを図3に示す。

親タスクは、先導タスクとの窓口となるとともに、子タスクを繰り返しリクエストすることにより解析を進めていく役割を果たす。

子タスク群は、すべて同一のタスクコード（プログラム）を持つタスクであり、親タスクからのリクエストを受けて、辞書検索、接続検定等の処理を行なう。

解析は、先導タスクが親タスクをリクエストすることにより開始する。この時、パラメータは、T R B（タスクリクエストブロック）のパラメータ部を介して親タスクに渡される。

親タスクへ渡されるパラメータのうち主なものは次の通りである。

- カナ文字列先頭アドレス
- カナ文字列長
- 出力領域先頭アドレス
- 出力領域サイズ
- 出力データ長
- 解析結果数（同音語数）

変換途中において、親タスクと子タスクのデータの受け渡しは、T B（トランスファブロック）を介することによって行なわれる。

T Bが有する主なデータは、次の通りである。

- 未変換文字列についての情報
- 変換済の解析木のリーフ部の単語ノードへのポインタおよびこの単語の文法情報
- 文節終了判定フラグ
- 次に検索すべき辞書を示す情報
- リクエスト用 T R B 情報
- キューイング用優先順位および T B 用ポインタ

変換タスクにおけるカナ漢字変換の手順を図4に示す。

変換タスクは、次に示す手順で解析をすすめる。

ー親タスクー

① （ T B 作成 ）

T R B パラメータを参照し、ブート用初期 T B を作成し、キューにチェーンする。（この時点では、T B は1つだけ存在する。）

② （ 終了判定 ）

キューより T B を1つ取り出し、

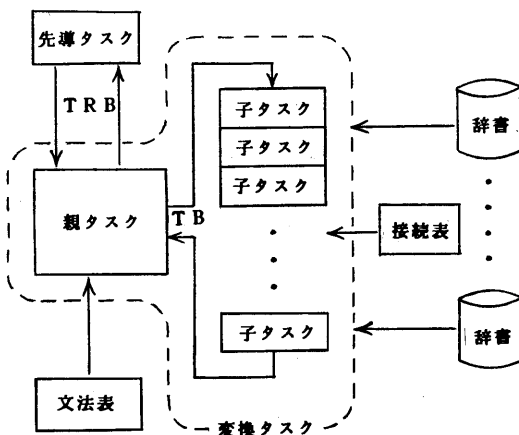


図3 変換タスクにおけるデータの流れ

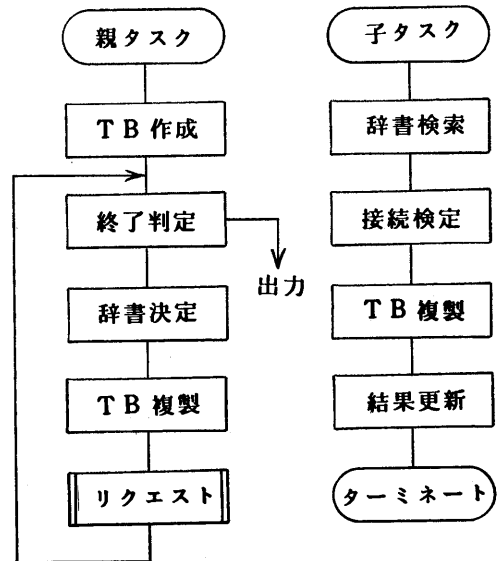


図4 親タスクと子タスクの動作

未解析カナ文字列部を調べ、存在しない場合には文節終了判定フラグを参照する。終了可である時には、このTBに関する解析が成功したものととして、TRBの出力領域に結果を出力した上でTBをリリースする。終了不可の場合には、解析が失敗したものととして、そのままTBをリリースする。(文節終了判定フラグは、子タスクの結果更新時にセットされる。)

③ (検索対象辞書決定)

未解析カナ文字列が残っている場合には、解析進行状態を調べ、文法表を参照することにより、次に検索すべき辞書を決定し、これをTBにセットする。

④ (TBの複製)

文法表を参照した結果、次に検索すべき辞書が複数個存在する場合には、TBを複製してそれぞれに辞書についての情報をセットする。

⑤ (子タスクのリクエスト)

④までで得られたTBをパラメータとして、子タスクをリクエストする。そして②に戻る。

なお、親タスクは上記の処理の他に下記の処理も行なう。

○タスク、辞書等の初期化

先導タスクからの第一リクエストを受け、メモリ確保、子タスク群の生成、辞書・テーブル類の初期化等を行なう。

○カタカナ、数字等のエスケープ

入力文字列内に存在する、カタカナ・数字・英記号など、辞書とマッチングできないものについて、あらかじめエスケープコードに置き換えておき、変換終了後に復元する。

○変換結果評価

変換結果の優先度を算出する優先度決定タスクをリクエストする。

○出力文字列作成

解析結果を先導タスクに出力するために、作成された解析木をたどることによって出力すべき文字列を作成する。(これを漢字かな混じり文字列に変換する処理は、漢字列作成タスクが行なう。)

○学習処理

先導タスクからの学習処理要求を受け、学習機能タスクをリクエストする。

○後処理

先導タスクからの終了要求を受けて、辞書・テーブル類の後処理、子タスク群の消去、メモリ解放等を行なう。

—子タスク—

① (辞書検索)

親タスクより渡されたTBを参照し、指定された辞書を、未変換カナ文字列で検索し、カナ文字列の先頭から始まる部分文字列と一致する見出しをもつ単語を、辞書の中からすべて抽出する。単語が検索できなかった場合にはTBをリリースして、タスクはターミネートする。

② (接続検定)

①で得られた単語が、TBが示す既解析木に文法的に接続できるか否かを、接続表を参照することによって調べる。

③ (TBの複製)

②で接続可と判定された単語が複数個存在する場合には、TBを複製する。

④ (変換結果の更新)

抽出された単語を解析木に結合するとともに、TB内の情報を更新する。そして、子タスクはターミネートする。この時、TBは(複製されたものも含め)親タスクに戻され、キューにチェーンされる。

1 文節に対する処理は、T B が 0 個になった時点で終了する。T B の個数管理や、T B の生成、抹消、親タスクと子タスクの間の転送など、T B に関する処理は、ユーティリティの 1 つであるタスク間転送管理エグゼクティブが行なう。

変換処理で作成した解析木の例を図 5 に示す。解析木は、抽出された単語のリスト構造（文節末→文節頭向きのポインタ）で表現される。

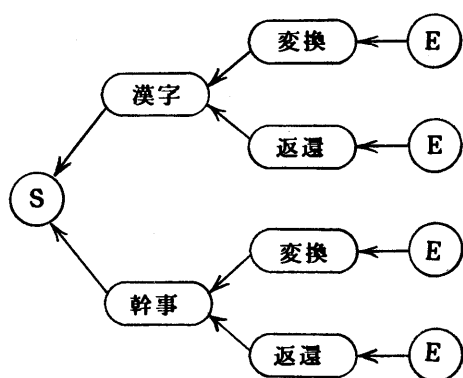


図5 解析木の例

ーエスケープ処理についてー

本システムにおいて、数値列や英記号列やカタカナ列（前後にカタカナシフト記号を付加することによってひらがな列と区別する）については、各文字列を見出しとする単語を辞書の中に持たずに、前処理時にこれら文字列をエスケープコードに置き換えて解析を行ない、解析終了後エスケープコードを復元する方式をとっている。

エスケープ処理の例として、「1 ジセッケイ〔システム〕ヲ」という文字列の処理手順を図 6 に示す。この例で、「〔」と「〕」がカタカナシフトコードである。

(1) 入力カナ文字列の中からエスケープ対象文字列を抽出し、F I F O スタックに格納しておくとともに、入力カナ文字列の所定の部分をエスケープコードに置き換える。（親タスク）

(2) エスケープ済み文字列を用いてカナ漢字変換処理を行なう。辞書の中には、エスケープコードを見出しとする項目が登録されており、カナ文字列の処理と何ら変わることなく解析を行なうことができる。（親タスクおよび子タスク群）

(3) 解析木から出力文字列を形成する処理において、解析木のノードにエスケープコードがある場合には、スタックの語と置き換える。（この時、J I S 1 6 系文字列への変換が行なわれる。）

これらの処理により、
変換システム（複合語）

第 1 回（助数詞＋名詞）

H / W 化（英記号＋接尾語）

についても、1 文節として扱うことができる。

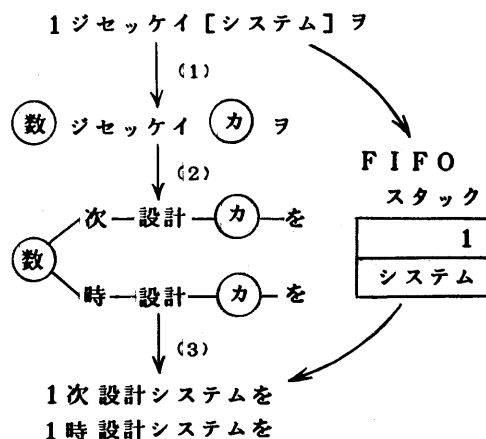


図6 エスケープ処理手順

4.2 優先度決定タスク

図5の解析木において、ダミーノードである③と④の間を結ぶパス1つ1つが、いわゆる同音語（ここでは、同音語を広義解釈して、品詞の違う語や全く別の単語区切りを持つ語も同音語と呼ぶ）である。優先度決定タスクは同音語に対して種々の評価関数を適用して評価値を求め、これらを総合して優先度を決定する役割を果たす。

優先度Pは、次式で与えられる。

$$P = \sum_{i=1}^n (f_i * W_i)$$

上式において、nは評価関数の個数である。また、 f_i は第i番目の評価関数によって得られる評価値であり、 W_i は評価関数 f_i の重み付け値である。

評価関数については、現在のところ下記のものをを用いている。

(1) 単語接続

解析木のパスを構成する単語列は文法的に正しいものであるものの、単語間の結びつきの強弱については評価されていない。そこで、品詞レベルでの隣接する単語間の関係のウェイトを設定し、これらの和によって接続関係を評価する。

(2) 見出し長

単語の見出しの長さが同音語の優先度決定にとって重要なファクタであることが知られている。本システムにおいては、単語の見出し長にその単語のウェイト値（品詞、文節内における位置、出現回数等によって決定される）を乗じた値の総和を、評価値としている。

(3) 複合語ポインタ

名詞から別の名詞、名詞から接辞の項目へのポインタ。ポインタが既に存在するか否かを評価値として出力する。

(4) 短期頻度

最近選択された語の優先度を上げるための評価関数である。選択された事実は、学習機能タスクによってシステム内に記憶される。

(5) 長期頻度

自立語辞書の項目の中の、頻度カウンタの値。語が選択されるたびに学習機能タスクにより更新される。

4.3 学習機能タスク

本システムの利用者は、カナ文字列を漢字かな混じり文字列に変換した後で、出力された同音語のうちから所望のものを選択することになる。本システムは、出力の1つが選択されると、これを構成する単語が次回から優先されるように処理を行なう。これを行なうのが学習機能タスクである。

学習機能タスクは、先導タスクからのパラメータ（選択された文字列に関する情報）を受けた親タスクからリクエストされることにより起動される。

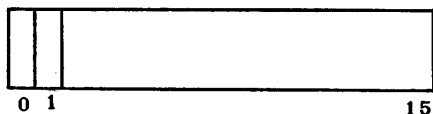
学習機能タスクは、下記の処理を行なう。

(1) 複合語ポインタ追加

選択された文節の中に、〈名詞〉〈名詞〉の組合せが存在する場合には、辞書内の先行する名詞のレコードの複合語ポインタ用フィールドに後続名詞のレコード番号を書き込むことによって、名詞間に参照用ポインタをセットする。また、〈名詞〉〈接尾語〉や〈接頭語〉〈名詞〉の組合せが存在する場合には、名詞のレコードの複合語ポインタ用フィールドに、接頭・接尾の種別とレコード番号がセットされる。図7に、複合語ポインタの構成を示す。

(2) 短期学習

選択された自立語を主メモリ上の被選択自立語バッファに登録する。



●辞書種別 0, 1ビット

0ビット=0 : 自立語辞書

0ビット=1 1ビット=0 : 接頭語辞書

0ビット=1 1ビット=1 : 接尾語辞書

●レコード番号

1~15ビット : 自立語辞書

2~15ビット : 接頭語, 接尾語辞書

図7 複合語ポインタ構成

(3) 長期頻度

選択された自立語に対応する辞書内のレコードにおける頻度カウンタを更新する。

これら学習処理の結果は、前述の優先度決定処理の際に使用される。

5. ユーティリティ群

ユーティリティ群は、主に処理系タスクの実行を補助するために使用される。

タスク間転送管理は、変換タスクにおけるタスク間の相互通信、同期等を管理するほか、タスクの生成や抹消、TBの転送など、タスク全般の管理を行なう。

メモリ管理は、変換タスク稼動時に生じるメモリ取得・解放を円滑に行なわせるため、専用メモリプールを設けて動的なメモリセルの供給に便宜を図るためのものである。

共通ルーチン群は、文字列検索、リスト処理等、下位部分を受持つための再入可能プログラムの集合である。

6. 入力系タスク群

入力系タスク群は、処理系に送るカナ文字列を作成する役割を果たす。入力形態については、

1. カナ文字
2. ローマ字
3. 単音節音声

の3種類を想定している。これらのうち、1と2については、キーボードからの入力で、入力したい文字を一意に決定しながら文字列を作成することができるが、3については、単音節音声認識装置から得られるカナ文字が必ずしも一意に決定されないため、場合によっては誤まったカナ文字を含む文字列を作成してしまうこともある。本システムでは、単音節音声認識装置から、複数の単音節音声候補およびその確からしさを表わす尤度を得て、文字候補と尤度のマトリクスを作成し、これによりカナ文字列を得る方式がとられている。

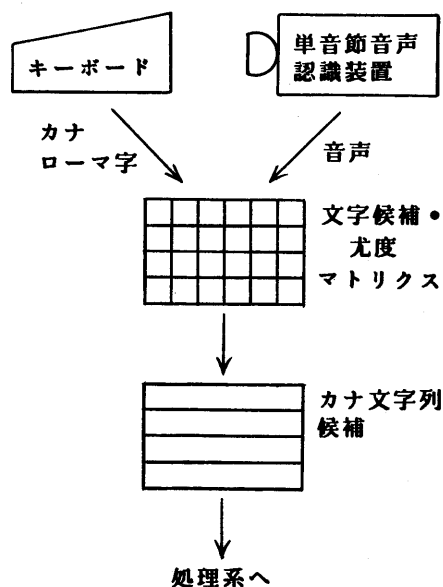


図8 入力系タスク概要

図8は、入力系タスクの概要を表わしたものであるが、カナ文字、ローマ字による入力と単音節候補による入力は、入力系内部において全く同じデータ構造を持つように構成されている。このとき、入力系において、キーボードからの入力は、最も確からしい文字候補（当然候補数は1である）であるものとみなされ、内部形式に変換される。入力系は、キーボードと音声認識装置のどちらからの入力も受けつけることができ、両者からの入力が混在することも可能である。

7. 出力系タスク群

処理系タスクにより得た漢字かな混じり文字列は、出力系タスクによって表示・出力される。出力系タスク群の概要を図9に示す。

漢字列出力タスクは、与えられた漢字かな混じり文字列を表示出力するためのタスクで、主にエコーバック表示等に用いられる。

WPタスクは、処理系で得られた漢字かな混じり文字列をもとに、テキストを作成するために用いられるタスクで、テキスト文字列および各種編集コマンドを受けて、日本語テキストを作成する。WPタスクで作成された日本語テキストは、統合文書処理システムIDEOS〔2〕において使用することができる。

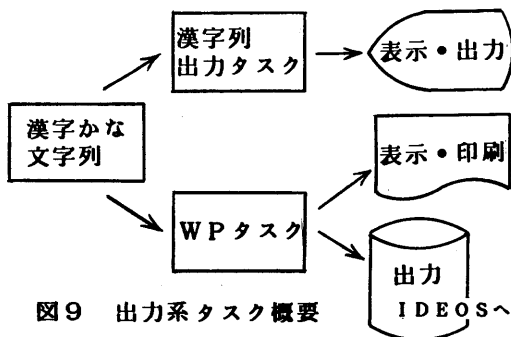


図9 出力系タスク概要

8. 辞書・テーブル

本システムで現在使用している辞書を、図10に示す。

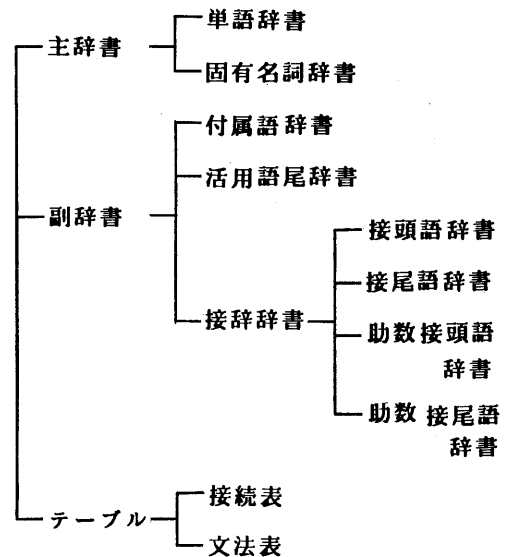


図10 辞書・テーブル

〔1〕主辞書

自立語用辞書で、磁気ディスク上に置かれている。単語の見出し部とレコード番号をベタ詰めした検食用ヘッダを持ち、これがメモリに常駐される。検索時には、ヘッダ部のバイナリサーチによってレコード番号を得た上で、これをキーにディスクからレコードを読み出す。辞書は次の2つである。

- ・単語辞書 約15000語
- ・固有名詞辞書（苗字・名前等）約5000語

〔2〕副辞書

メモリ常駐の辞書で、先頭1文字のインデックスヘッダを持つ。検索は、ヘッダで参照される範囲をシーケンシャルに調べていく方式をとる。

副辞書は、下記の辞書により構成される。

- ・付属語辞書助詞，助動詞等の他補助動詞，連語等も含まれる。

約 5 0 0 語

- ・活用語尾辞書 動詞・形容詞用

約 1 0 0 語

- ・接辞辞書

接頭語辞書，接尾語辞書，助数

接頭語辞書，助数接尾語辞書の

4 種類よりなる。約 3 5 0 語。

〔 3 〕 テーブル

本システムで使用する日本語文法を記述したもの。次の 2 つがある。

- ・接続表 2 つの単語が文法的に接続可能か否かを記述するためのテーブル。3 5 0 × 2 5 6 ビットのマトリクスで表現されている。メモリ常駐。
- ・文法表 カナ漢字変換の途中において，次に検索すべき辞書を状態遷移ネットワークの形で記述したもの。次章に実例を示す。メモリ常駐。

9 . 各種変換実験

本システムは，文節分かち書き入力を原則としているが，用いられる文節の定義は，第 8 章で述べた文法表の内容のみに依存するように設計されている。このため，文節の定義を変更したい場合には，文法表のみ変更すればよい。（文法自体は，文法表と接続表の 2 つのテーブルで管理されるが，種々の文法の定義に対して，接続表は同一のものを使用できる。）

9 . 1 基本構成文節分かち書き

本システムでは，図 1 1 - a の文法記述を，基本構成文節として定義している。

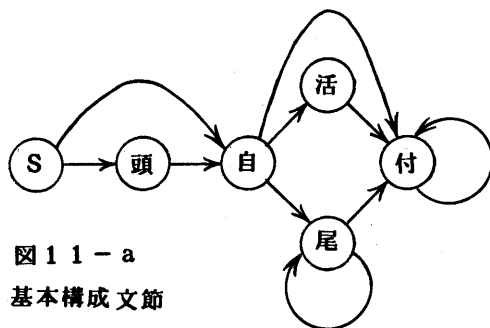


図 1 1 - a

基本構成文節

入力用文法表

図において，S はスタート用ノードであり，頭は接頭語辞書，自は自立語（単語・固有名詞）辞書，活は活用語尾辞書，尾は接尾語辞書，付は付属語辞書である。文節の終了は，接続表によって管理されるため，文法表の中には表わされていない。助数接頭語と助数接尾語は，それぞれ接頭語，接尾語の部分に位置する。なお，接尾語辞書の部分にループが存在するのは，「事業部用（事業＋部＋用）」や「8 0 点台（8 0 ＋点＋台）」等の文節に対処するためである。また，付属語の中には，形式名詞や補助動詞，連語等も含まれているため，「～するうちに」，「～となる」，「～における」等は，1 文節として取り扱うことができる。

9 . 2 拡張文節分かち書き

図 1 1 - a の文法に加え，複合語の入力を可能としたものである。文法表は，図 1 1 - b のようになる。

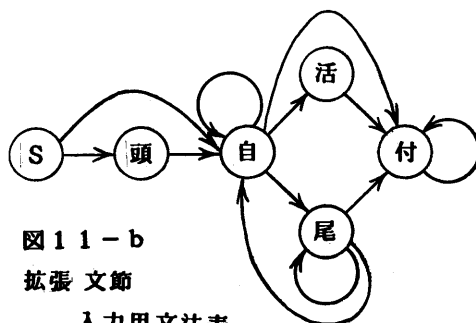


図 1 1 - b

拡張文節

入力用文法表

本システムにおいて、複合語は、名詞および名詞＋接尾語の繰り返しと定義しているため、情報処理学会（情報＋処理＋学会）や、日本語入力法（日本＋語＋入力＋法）は、1文節として認められる。

9.3 複数文節入力

9.1, 9.2 は、共に文節分かち書きの例であるが、本システムにおいては、図11-cのように文法表を定義し、付属語辞書にレコードを1つ追加するだけで、複数文節入力を実現することができる。

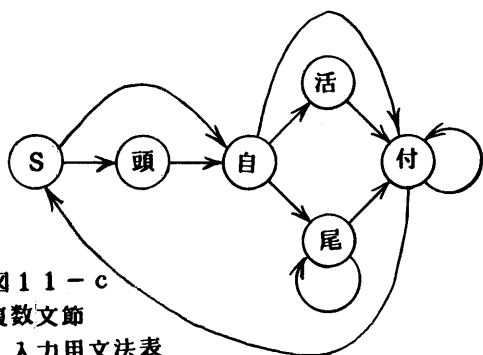


図11-c
複数文節
入力用文法表

図11-cは、基本構成文節入力用文法表（図11-a）に、付属語辞書からスタートノードへの戻りを可能とするパスを追加したものである。

付属語辞書に追加するレコードは、

- ・見出し、表記ともに nullで、見出し長、表記長がともに0。
- ・文節末になりうる（すなわち終了可）単語の後に接続する。
- ・文節の先頭になりうる単語の前に接続する。

の条件を満たすものである。

本システムでは、得られた変換結果に優先度を与える際に、変換結果内における上記レコードの数を、新たな評価関数として採用している。この関数（値が小さいほど確からしい）は、文節数最小法と同じ意味を持つ。

10. おわりに

本システムは、MS50上にインプリメントされ、現在システムの評価を行なうとともに、機能拡張作業を続けている。また、本システムのH/W化についても検討を加えている。

今後は、本システムのより一層の充実を図るとともに、本システムをベースに、マンマシンインタフェースのよい、高性能なカナ漢字変換システムを構築していく予定である。

－参考文献－

- [1] 大山, 宮井, 首藤, 小野田
「マルチタスク形式によるカナ漢字変換」 情報処理学会第25回全国大会予稿 6J-2 (1982)
- [2] 宮井, 森下, 首藤
「統合文書処理システム (IDEO S) - 構成と処理 -」 情報処理学会コンピュータビジョン研究会 17-2 (1982)