

パーソナル・コンピュータによる古語の自動活用

西端 幸雄

1. はじめに

最近、国語国文学の分野でも、パーソナル・コンピュータ（以下、パソコンと略す）を用いた作業報告が、以前に増して盛んに行われたり、古典のデータベースが発売されたりして、いよいよOA化の波が押し寄せてきたと言っても過言ではない。ただ、筆者の側聞するところでは、現段階では、国語国文学の分野でのパソコンの使用状況は、多くの場合、ワープロ・ソフトかデータベース・ソフトを用いるということとしか使われていないのが現状のようにである。また、国語国文学の分野での資料となるべき、市販の古典データベースの中には、その商品的価値と資料的価値を疑いたくなるようなものも出回り始めている。こうした現状から少しでも脱却するためには、使用者側がパソコンの持つ機能をより多角的に引き出す努力を行い、その一方で、データベースの提供者がより良質のデータベースを作成し、提供する努力を惜しまないことが必要であろう。そうした努力の積み重ねが、今後の、国語国文学の分野でのパソコンの利用、延いては、国語国文学の研究に資することになると思う。

ところで、筆者もパソコン用の古語辞典の作成、言い換えるなら、古語のデータベース化という作業を以前から行ってきた。しかし、この古語のデータベース化を行う際に、必要なデータを一つ一つキーボードから入力することは、多大な時間と労力を要するし、誤入力の危険性も高くなる。さらに、単にデータ入力を繰り返すだけでは、パソコンの持つ機能を半減させていることになり、筆者が、これまでパソコンと関わってきた姿勢にも反することになる。そのパソコンに関わる姿勢とは、パソコン上の作業を出来る限り自動化するとともに、入力したデータをそれだけにとどめず、そのデータから発展的、派生的データをも生み出す生産的作業を行わせるようにするというものである。

さて、小稿において、紹介するのは、用言の各活用形をプログラム上で自動的に生成させようとするものである。キーボードから語句（終止形）、品詞、活用形の種類を入力すれば、全ての活用形が求められる。それこそ、こうしたプログラムにどういった有用性があるのかという問題もあろうが、筆者が狙いとする有用性のいくつかを上げておく。古語のデータベース化を行う際、全ての活用形のデータを入力することは、煩雑であり、ディスクの容量を大量に費やしてしまう。そうした問題を解決する一手段として、データベース上には、終止形データだけを入力しておき、データを引き出す際、このプログラムによって、他の活用形を求めることも出来る。また、逆に、全ての活用形データを入力する必要があるという場合にも当然使える。さらに、人間が用言を活用させる思考過程を、このプログラムによって、シミュレートする事も出来る。そのほか、様々な使い方があろうが、その点、使用例を御教示いただければ、幸いである。

2. 用言の活用

① 活用

活用というのは、同一の単語が、文脈内部における用法の違いに応じて、語形を変化させる現象をいう。その現象は、用言（動詞・形容詞・形容動詞）と助動

詞に限り現れる。特に、用言の場合は、基本的には、その現象が現れるのは、語末であり、それを「活用語尾」と呼び、不変な部分を「語幹」と呼ぶ。

いま、その活用の具体例を、用言のひとつ、4段動詞の「読む」について見てみると、

「文を読まず」
 「文を読みけり」
 「文を読む」
 「文を読む時」
 「文を読めども」
 「文を読め」

このように、形態的には、4種類、構文的機能という観点からすれば、6種類の異なった語形を示す。一般に活用という場合、この内の構文的機能による語形の変化をいう。そして、その構文的機能の違いを識別するため、上記の例でいえば、上から順に、「未然形・連用形・終止形・連体形・已然形・命令形」と、その機能の一部を代表させた命名が便宜的になされており、これらを「活用形」と呼ぶ。

さらに、用言の活用の仕方には、一定の法則性がある。その法則性というのは、まず、用言全体を見た場合、以下のようなものである。

1. 動詞は、五十音図の一行だけに限って活用する。
2. 形容詞は、カ行とサ行とラ行に限って活用する。
3. 形容動詞は、ラ行に限って活用する。

この法則性を、品詞別にみても、さらに細かい法則のあることが分かる。まず、動詞について見てみると、以下の通りである。

1. 五十音図のア・イ・ウ・エの4段にわたって活用するもの（4段活用）
 さらに、この場合、五十音図のカ・ガ・サ・タ・ハ・バ・マ・ラの各行内でしか活用しない。
2. 五十音図のラ行内だけで、ア・イ・ウ・エの4段にわたって活用するもの（ラ行変格活用）
3. 五十音図のナ行内だけで、ア・イ・ウ・エの4段にわたって活用するもの（ナ行変格活用）
 さらに、連体形・已然形の各末尾は、それぞれ「る」「れ」となる。
4. 五十音図のイ段だけで活用するもの（上一段活用）
 さらに、この場合、五十音図のカ・ナ・ハ・マ・ヤ・ワの各行内でしか活用しない。
5. 五十音図のイ・ウの2段にわたって活用するもの（上2段活用）
 さらに、この場合、五十音図のカ・ガ・タ・ダ・ハ・バ・マ・ヤ・ラの各行内でしか活用しない。
6. 五十音図のエ段だけで活用するもの（下1段活用）
 ただし、「蹴る」の一語だけである。
7. 五十音図のウ・エの2段にわたって活用するもの（下2段活用）
 さらに、この場合、五十音図のア・カ・ガ・サ・ザ・タ・ダ・ナ・ハ・バ・マ・ヤ・ラ・ワの各行内でしか活用しない。
8. 五十音図のカ行内だけで、イ・ウ・オの3段にわたって活用するもの（カ行変格活用）
 ただし、単独語としては、「来(く)」の一語だけである。
9. 五十音図のサ(ザ)行内だけで、イ・ウ・エの3段にわたって活用するもの（サ行変格活用）

ただし、単独語としては、「す」の一語だけである。

上記の4.（上1段活用）から9.（サ行変格活用）までの連体形・已然形・命令形の各末尾は、それぞれ「る」「れ」「よ」となる。

次に、形容詞については、

1. 五十音図のカ・サの各行内で、イ・ウ・エの3段にわたってしか活用せず、体言が下の付くと、語尾が「き」となるもの。（ク活用）
2. 活用語尾の頭が「し」となり、五十音図のカ・サ各行内で、イ・ウ・エの3段にわたってしか活用せず、体言が下に付くと、語尾が「しき」となるもの。（シク活用）
3. 活用語尾の後から2文字目が「か」となり、語尾の活用の法則がラ行変格活用に準じるもの（カリ活用）

という法則性がある。

さらに、形容動詞については、

1. 五十音図のラ行内で、ア・イ・ウ・エの4段にわたってしか活用せず、語尾の活用の法則がラ行変格活用に準じるもの。ただし、連用形が「～に」または「～と」となることもある。

という法則性がある。

こうした活用の法則性というものは、現実の言語生活においては、あまり意識されることもないだろうが、いわゆる古典文法を学習したり、それをもとにして古文の品詞分解をするといった時には、これらの法則を記憶し、応用することがあろう。そして、この法則性がかなり整然としたものであるもので、記憶するにも、応用するにも、他の文法事項に対するほどの労力を必要としないと思われる。その法則性が整然としているところから、この法則性をプログラム化することによって、各活用形を求めることが容易にできる。

② プログラム化

前項で述べてきた、用言の活用にみられる法則性の中核的な部分をまとめると、次のようになろう。

- A 活用語尾の母音が変わるもの
4段活用・ラ行変格活用・形容詞・形容動詞
- B 活用語尾の母音が変わり、それに「る」「れ」「よ」が付加されるもの
上2段活用・下2段活用・ナ行変格活用
- C 語幹と活用語尾との区別がなく、同一行内で活用し、それに「る」「れ」「よ」が付加されるもの
上1段活用・下1段活用・カ行変格活用・サ行変格活用

上記の3項をプログラム化しようとする場合、A・BとCの間では、活用語尾の母音変化の有無という点で、また、AとB・Cの間では、一部の活用形に「る・れ・よ」が付加されるかどうかという点で、それぞれの項目について、異なった方法を用いなければならないことがわかる。そこで、以下に、各項目別にプログラム化する方法を説明することにする。

③ 4段動詞等

まず、<A 活用語尾の母音が変わるもの>に該当するのは、4段活用の動詞がその代表例である。この場合、活用語尾がア・イ・ウ・エの4段にわたって活用するという規則性に着目し、活用語尾の文字に充てられているJ I Sコードを利用して、他の活用形を求める方法が取り入れられる。

いま、先に掲げた「読む」を例とすると、その活用の形態的な変化は、

読ま ま→245E
 読み み→245F
 読む む→2460
 読め め→2461

となる。そして、それぞれの活用語尾の J I S コードは、右項に掲げた通りであり、その J I S コードを求めるには、以下に記した J I S \$ 関数を用いる。

10 J=VAL(JIS\$(K\$))

そして、上の計算式を基に求めた終止形の活用語尾「む」の J I S コードを基準として、他の文字の J I S コードを求めるには、

ま 2460-2 (16進数)
 み 2460-1
 め 2460+1

という計算式が成り立つ。

ところが、各平仮名に充てられた J I S コードは、上記のマ行のように1ずつ増加するとは限っていない。いま、4段活用の動詞が、カ・ガ・サ・タ・ハ・バ・マ・ラの各行の ア・イ・ウ・エ の4段にわたって活用するということであるから、それぞれの J I S コードを示すと、以下の通りである。

か	242B	さ	2435	は	244F	ま	245E
き	242D	し	2437	ひ	2452	み	245F
く	242F	す	2439	ふ	2455	む	2460
け	2431	せ	243B	へ	2458	め	2461
が	242C	た	243F	ば	2450	ら	2469
ぎ	242E	ち	2441	び	2453	り	246A
ぐ	2430	つ	2444	ぶ	2456	る	246B
げ	2432	て	2446	べ	2459	れ	246C

この J I S コードを見ると、マ・ラの各行のように1ずつ増加している場合と、カ・ガ・サの各行のように2ずつ増加している場合と、ハ・バの各行のように3ずつ増加する場合と、さらに、タ行のように、「ち」「つ」の前後では、2ずつの増加であるが、その「ち」「つ」の間だけ3増加しているというものもある。そのため、終止形を基にして、各活用形を求めようとした場合、終止形の活用語尾が、五十音図の何行の文字であるかを J I S コードを手がかりに判別して、それぞれの J I S コード別に計算式を立てなければならなくなる。

では、4段動詞の各活用形の求め方を説明することにする。

```
1000 REM 4段動詞
1010 WL$="":WR$="":WL$=LEFT$(GDAT$(KLEN(GDAT$)-1)*2):WR$=RIGHT$(GDAT$,2):KRJIS=VAL("&H"+JIS$(WR$))
1020 IF KRJIS=&H242B AND KRJIS<&H243D THEN COZ=&H4:COR=&H2:COT=&H2 ELSE IF KRJIS=&H243F AND KRJIS<&H2449 THEN COZ=&H5:COR=&H3:COT=&H2
1030 IF KRJIS=&H244F AND KRJIS<&H245C THEN COZ=&H6:COR=&H3:COT=&H3 ELSE IF KRJIS=&H245E AND KRJIS<&H246D THEN COZ=&H2:COR=&H1:COT=&H1
1040 REM 4段の活用
1050 GD$(1)=WL$+KNJ$(HEX$(KRJIS-COZ)):GD$(2)=WL$+KNJ$(HEX$(KRJIS-COR)):GD$(3)=GDAT$:GD$(4)=GDAT$:GD$(5)=WL$+KNJ$(HEX$(KRJIS+COT)):GD$(6)=WL$+KNJ$(HEX$(KRJIS+COT))
1060 REM 表示
      (以下、省略)
```

4 段動詞の場合、活用する部分（活用語尾）は、語句の右1文字であり、活用しない部分（語幹）は、語句の長さから1文字分引いた長さのものであるから、その語幹と活用語尾を、さらに、活用語尾の文字のJ I Sコードを<1010>行で求めている。以下に示したのが、用いている変数の内容である。

WL\$ （語幹）

WR\$ （終止形の活用語尾）

KRJIS（終止形の活用語尾の文字のJ I Sコード）

次の<1020~1030>行では、そのJ I Sコードに応じて、終止形以外の活用形を求めるために、前述した各行によって異なるJ I Sコードの増値分をパラメータとして、3種類の数値変数に代入する。この3種類の数値変数の内容は、以下の通りである。

COZ（ア段文字からウ段文字までのJ I Sコードの増値分）

COR（イ段文字からウ段文字までのJ I Sコードの増値分）

COT（ウ段文字からエ段文字までのJ I Sコードの増値分）

さらに、次の<1050>行で、上記の数値変数を用いて、各活用形の活用語尾を求めているのである。その方法として、終止形の活用語尾の文字（ウ段文字）のJ I Sコードを表す数値変数KRJISに対して、上記の増値分を差し引いたり、加えたりすることによって求めている。もう少し具体的に説明すると、ア段文字（未然形）やイ段文字（連用形）は、ウ段文字よりJ I Sコードが小さいから、KRJISからCOZやCORを差し引けば、それぞれの文字のJ I Sコードが求められるし、また、エ段文字（已然形・命令形）は、ウ段文字よりJ I Sコードが大きいため、KRJISにCOTを加えれば、その文字のJ I Sコードが求められる。そして、その求められた各文字のJ I Sコードに応じて、KNJ\$間数を用いて、全角の2バイト文字に変換し、それを語幹（WL\$）の後に付けることによって、各活用形毎の語形を求め、配列変数（GD\$）に収めているのである。

上記の< A 活用語尾の母音が変わるもの >の類例として、動詞のラ行変格活用（以下、ラ変と略す）や形容詞、形容動詞がある。それぞれの活用形は、以下に示すとおり、原則的には活用語尾の母音が規則的に変化はしている。

・ラ変

有ら
有り
有り
有る
有れ
有れ

・形容詞

ク活用

多く
多し
多き
多けれ

カリ活用

多から
多かり
多かる
多かれ

シク活用

美しく
美し
美しき
美しけれ

カリ活用

美しから
美しかり
美しかる
美しかれ

・形容動詞

ナリ活用

静かなら
静かなり (に)
静かなり
静かなる
静かなれ
静かなれ

タリ活用

堂々たら
堂々たり (と)
堂々たり
堂々たる
堂々たれ
堂々たれ

しかし、先に述べた「活用語尾の母音に変化する」とは言っても、形容詞については、カリ活用においてだけで、ク・シク活用では、一部に規則的な変化はみられるものの、全体としては、その規則性は崩れている。また、形容動詞については、ナリ活用・タリ活用ともに、ラ行内で規則的に変化しているが、連用形に「～に」「～と」の活用があり、その部分では、規則性が崩れているといえる。さらに、ラ変や形容詞のナリ活用、形容動詞の主たる活用は、ラ行内に限って行われる。そのため、こうしたものについては、いちいち活用語尾の文字のJ I Sコードによって各活用形を求めることをせず、語幹に、それぞれの活用形の活用語尾を直接付加するという方法をとった。以下に示すプログラムが、それぞれの活用形を求めるものである。

まず、ラ変の場合、<1010>行で元の語句の語幹(WL\$)を求め、その語幹に<1020>行で、活用語尾「ら・る・れ」(連用形と終止形は、元の語句のまま)を付加するという方法をとっている。(<1020>行で、文字変数BUNP\$に代入される「ら3」は、私に決めたラ変の終止形を表すコード番号である。以下、同じ)

```
1000 REM ラ変
1010 WL$="":WL$=LEFT$(GDAT$(KLEN(GDAT$)-1)*2)
1020 IF BUNP$="ら3" THEN GD$(1)=WL$+"ら":GD$(2)=GDAT$:GD$(3)=GDAT$:GD$(4)
=WL$+"る":GD$(5)=WL$+"れ":GD$(6)=WL$+"れ":GOTO 1030
1030 REM 表示
(以下、省略)
```

形容詞や形容動詞の場合も、それぞれラ変と同じ方法ではあるが、活用の種類によって、語幹に付加すべき活用語尾の語形が違うので、形容詞については、<1030・1050・1070>の各行の最初で、また、形容動詞については、<1030・1050>の各行の最初で、活用の種類を判定し、当該の活用の種類に応じた活用語尾を付加させている。

なお、形容詞のシク活用には、「同じ」のように、活用語尾が「～じ」となるものがあるので、その区別を終止形の活用語尾(WR\$)によって行っている。

```

1000 REM 形容詞
1010 WL$="" : WR$="" : WL$=LEFT$(GDAT$, (KLEN(GDAT$)-1)*2) : WR$=RIGHT$(GDAT$, 2)
)
1020 REM ク活用
1030 IF BUNP$="73" THEN GD$(1)=WL$+"<" : GD$(3)=WL$+"<" : GD$(5)=GDAT$:GD$(6)=WL$+"き" : GD$(8)=WL$+"けれ" : GD$(2)=WL$+"から" : GD$(4)=WL$+"かり" : GD$(7)=WL$+"かる" : GD$(9)=WL$+"かれ" : GOTO 1080
1040 REM シク活用 (語尾がシのもの)
1050 IF BUNP$="73" AND WR$="し" THEN GD$(1)=WL$+"しく" : GD$(3)=WL$+"しく" : GD$(5)=GDAT$:GD$(6)=WL$+"しき" : GD$(8)=WL$+"しけれ" : GD$(2)=WL$+"しから" : GD$(4)=WL$+"しかり" : GD$(7)=WL$+"しかる" : GD$(9)=WL$+"しかれ" : GOTO 1080
1060 REM シク活用 (語尾がジのもの)
1070 IF BUNP$="73" AND WR$="じ" THEN GD$(1)=WL$+"じく" : GD$(3)=WL$+"じく" : GD$(5)=GDAT$:GD$(6)=WL$+"じき" : GD$(8)=WL$+"じけれ" : GD$(2)=WL$+"じから" : GD$(4)=WL$+"じかり" : GD$(7)=WL$+"じかる" : GD$(9)=WL$+"じかれ"
1080 REM 表示
(以下、省略)

```

```

1000 REM 形容動詞
1010 WL$="" : WL$=LEFT$(GDAT$, (KLEN(GDAT$)-2)*2)
1020 REM ナリ活用
1030 IF BUNP$="N3" THEN GD$(1)=WL$+"なら" : GD$(2)=GDAT$:GD$(3)=WL$+"に" : GD$(4)=GDAT$:GD$(5)=WL$+"なる" : GD$(6)=WL$+"なれ" : GD$(7)=WL$+"なれ" : GOTO 1060
1040 REM タリ活用
1050 IF BUNP$="T3" THEN GD$(1)=WL$+"たら" : GD$(2)=GDAT$:GD$(3)=WL$+"と" : GD$(4)=GDAT$:GD$(5)=WL$+"たる" : GD$(6)=WL$+"たれ" : GD$(7)=WL$+"たれ"
1060 REM 表示
(以下、省略)

```

④ 下2段動詞等

次に、<B 活用語尾の母音が変わり、それに「る」「れ」「よ」が添加されるもの>として、下2段動詞を例にして、その各活用形の求め方を説明する。

例えば、下2段動詞「受く」の活用の仕方を見ると、以下のようになる。

受け	未然形
受け	連用形
受く	終止形
受くる	連体形
受くれ	已然形
受けよ	命令形

このように、下2段動詞は、まず、ウ・エの2段にわたって活用語尾が変わり、また、その活用語尾に「る」「れ」「よ」が付加されることによって、連体形・已然形・命令形をそれぞれ形成していることが分かる。だから、4段動詞のような方法だけでは、各活用形が求められないということになる。そこで、この下2段動詞の各活用形を求める場合、4段動詞の場合に用いた活用語尾の文字のJISコードによって未然形を求め、その未然形と終止形に、それぞれ「る」「れ」

「よ」を付加し、連体形・已然形・命令形を求めるといった方法をとっている。それをプログラム化したのが下記のものである。

```

1000 REM 下2段動詞
1010 WLS="":WR$=""WLS=LEFT$(GDAT$(KLEN(GDAT$)-1)*2):WR$=RIGHT$(GDAT$,2)
:KRJIS=VAL("&H"+JIS$(WR$))
1020 IF WR$="ゆ" THEN GD$(1)=WLS+"え":GD$(2)=WLS+"え":GD$(3)=GDAT$:GD$(4)
)=GDAT$+"る":GD$(5)=GDAT$+"れ":GD$(6)=WLS+"えよ":GOTO 1060
1030 IF WR$="う" THEN GD$(1)=WLS+"ゑ":GD$(2)=WLS+"ゑ":GD$(3)=GDAT$:GD$(4)
)=GDAT$+"る":GD$(5)=GDAT$+"れ":GD$(6)=WLS+"ゑよ":GOTO 1060
1040 IF KRJIS=>&H2422 AND KRJIS<&H2449 THEN COZ=&H2 ELSE IF KRJIS=>&H24
4F AND KRJIS<&H245C THEN COZ=&H3 ELSE IF (KRJIS=>&H244A AND KRJIS<&H24
4E) OR (KRJIS=>&H245E AND KRJIS<&H246D) THEN COZ=&H1
1050 GD$(1)=WLS+KNJ$(HEX$(KRJIS+COZ)):GD$(2)=GD$(1):GD$(3)=GDAT$:GD$(4)=
GDAT$+"る":GD$(5)=GDAT$+"れ":GD$(6)=GD$(1)+"よ"
1060 REM 表示

```

(以下、省略)

用いている変数は、先の4段動詞の場合と同じであるので説明は省略する。(以下、同じ)まず、<1020~1030>行で、ヤ行とワ行内で活用するものについて、その各活用形を求めている。これは、現代仮名遣いには存在しない仮名遣いに応じた活用であるため、当然JISコードの規則性を利用しての計算式では、その各活用形を求めることができないものである。そのため、語幹と終止形の語形を基にして、それに活用語尾や「る」「れ」「よ」の各文字を付けるという方法をとっているのである。また、活用語尾が上記のヤ行やワ行内以外で活用するものについては、終止形の活用語尾(ウ段文字)から未然形・連用形の活用語尾(エ段文字)を求める際の、JISコードの増値分を幾らにするかを活用語尾の文字によって決めなければならない。(前掲のJISコード表参照)その判定を行っているのが<1040>行である。次の<1050>行において、その増値分を終止形の活用語尾のJISコードに加えて、まず、未然形を求め、その未然形を基に連用形と命令形(「よ」を付加)、また、終止形を基に連体形(「る」を付加)と已然形(「れ」を付加)を求めているのである。

上記の<B 活用語尾の母音>が変化し、それに「る」「れ」「よ」が添加されるもの>の類例としては、上2段活用やナ行変格活用(以下、ナ変と略す)の動詞がある。それぞれ活用形は、以下に示すとおりである。

・上2段活用	・ナ変
起き	死な
起き	死に
起く	死ぬ
起くる	死ぬる
起くれ	死ぬれ
起き(よ)	死ぬ

このうち、ナ変は、「死ぬ・往ぬ」の2語だけであり、4段動詞に近い活用の仕方をとってはいるが、連体形と已然形で規則性が崩れていることから、前述のラ変と同じように、語幹に活用語尾「な・に・ぬる・ぬれ・ぬ」(終止形は、元の語句のまま)を付加するという方法をとっている。

```

1000 REM ナ変
1010 WL$="":WR$="":WL$=LEFT$(GDAT$, (KLEN(GDAT$)-1)*2)
1020 IF BUNP$="+3" THEN GD$(1)=WL$+"な":GD$(2)=WL$+"に":GD$(3)=GDAT$:GD$(4)=WL$+"ぬる":GD$(5)=WL$+"ぬれ":GD$(6)=WL$+"ね"
1030 REM 表示
      (以下、省略)

```

また、上2段動詞の場合は、未然形・連用形・命令形の活用語尾がイ段文字であるから、前述の下2段動詞の場合とは逆に、終止形の活用語尾(ウ段文字)から未然形の活用語尾を求める際に、イ段文字とウ段文字との間のJ I Sコードの増値分をウ段文字のJ I Sコードから差し引けばよいということになる。(前掲のJ I Sコード表参照)

```

1000 REM 上2段動詞
1010 WL$="":WR$="":WL$=LEFT$(GDAT$, (KLEN(GDAT$)-1)*2):WR$=RIGHT$(GDAT$, 2):KRJIS=VAL("&H"+JIS$(WR$))
1020 IF WR$="ゆ" THEN GD$(1)=WL$+"い":GD$(2)=WL$+"い":GD$(3)=GDAT$:GD$(4)=GDAT$+"る":GD$(5)=GDAT$+"れ":GD$(6)=WL$+"いよ":GOTO 1050
1030 IF KRJIS>&H242B AND KRJIS<&H2434 THEN COZ=&H2 ELSE IF KRJIS>&H243F AND KRJIS<&H245C THEN COZ=&H3 ELSE IF KRJIS>&H245E AND KRJIS<&H246D THEN COZ=&H1
1040 GD$(1)=WL$+KNJ$(HEX$(KRJIS-COZ)):GD$(2)=WL$+KNJ$(HEX$(KRJIS-COZ)):GD$(3)=GDAT$:GD$(4)=GDAT$+"る":GD$(5)=GDAT$+"れ":GD$(6)=WL$+KNJ$(HEX$(KRJIS-COZ))+ "よ"
1050 REM 表示
      (以下、省略)

```

<1020>行で、ヤ行内で活用するものについて、その各活用形を求めている。これは、現代仮名遣いには存在しない仮名遣いによる活用であるため、当然J I Sコードの規則性を利用しての計算式では、その各活用形を求めることができないものである。そのため、語幹と終止形の語形を基にして、それに活用語尾や「る」「れ」「よ」の各文字を付けるという方法をとっているのである。また、活用語尾が上記のヤ行内以外で活用するものについては、終止形の活用語尾(ウ段文字)から未然形の活用語尾(イ段文字)を求める際の、J I Sコードの減値分を幾らにするかを活用語尾の文字によって決めなければならない。(前掲のJ I Sコード表参照)その判定を行っているのが<1030>行である。次の<1040>行において、その減値分を終止形の活用語尾のJ I Sコードから差し引いて、まず未然形を求め、その未然形を基に連用形と命令形(「よ」を付加)、また、終止形を基に連体形(「る」を付加)と已然形(「れ」を付加)を求めているのである。

⑤ 上1段動詞等

最後に、<C 語幹と活用語尾との区別がなく、同一行内で活用し、それに「る」「れ」「よ」が添加されるもの>について、説明する。

いま、上1段活用の「見る」とカ行変格活用(以下、カ変と略す)の「(尋ね)来」を例として考えてみると、それぞれの活用形は、以下に示すとおりである。

・上1段活用

・カ変

み	たづねこ
み	たづねき
みる	たづねく
みる	たづねくる
みれ	たづねくれ
みよ	たづねこ(よ)

上1段動詞では、「みる」の中の「み」が全ての活用形において、変化していないことが分かる。言い換えれば、終止形の「みる」の語尾1文字を取り除いた「み」を基準にして、それをそのまま使えば、未然形・連用形、それに「る」を付加すれば、終止形・連体形、それに「れ」を付加すれば、已然形、それに「よ」を付加すれば、命令形が求められることになる。その点は、下1段動詞(実際には、「蹴る」だけ)も同様である。

また、力変の場合は、単独語として用いられることもあるが、上記の例のように複合語として用いられることも多いので、双方ともに処理できることを考慮した。力変は、前項で述べた下2段動詞と同様の、終止形の活用語尾のJISコードを基に未然形・連用形を求め、さらに、終止形や未然形に「る・れ・よ」を付加することによって、連体形・已然形・命令形を求めるという方法を用いることができるが、その語種が「く」1語であるということで、終止形の語尾「く」を取り除いたもの(単独語の場合は、空白となる)に、それぞれ「こ・き・く・くる・くれ・こよ」を付加するという単純な方法をとっている。その点は、サ行変格活用(以下、サ変と略す)も同様である。

以上に述べた点をプログラム化したのが下記のものである。

```

1000 REM
1010 WL$="" : WR$="" : WL$=LEFT$(GDAT$, (KLEN(GDAT$)-1)*2) : WR$=RIGHT$(GDAT$, 2) : KRJIS=VAL("&H"+JIS$(WR$))
1020 REM 上1段・下1段
1030 IF BUNP$="23" THEN GD$(1)=WL$:GD$(2)=WL$:GD$(3)=GDAT$:GD$(4)=GDAT$:GD$(5)=WL$+"れ":GD$(6)=WL$+"よ":GOTO 1090 ELSE IF BUNP$="43" THEN GD$(1)=WL$:GD$(2)=WL$:GD$(3)=GDAT$:GD$(4)=GDAT$:GD$(5)=WL$+"れ":GD$(6)=WL$+"よ":GOTO 1090
1040 REM 力変
1050 IF BUNP$="43" THEN GD$(1)=WL$+"こ":GD$(2)=WL$+"き":GD$(3)=GDAT$:GD$(4)=WL$+"くる":GD$(5)=WL$+"くれ":GD$(6)=WL$+"こ":GOTO 1090
1060 REM サ変
1070 IF BUNP$="43" AND WR$="す" THEN GD$(1)=WL$+"せ":GD$(2)=WL$+"し":GD$(3)=GDAT$:GD$(4)=WL$+"する":GD$(5)=WL$+"すれ":GD$(6)=WL$+"せよ":GOTO 1090
1080 IF BUNP$="43" AND WR$="ず" THEN GD$(1)=WL$+"ぜ":GD$(2)=WL$+"じ":GD$(3)=GDAT$:GD$(4)=WL$+"ずる":GD$(5)=WL$+"ずれ":GD$(6)=WL$+"ぜよ":GOTO 1090
1090 REM 表示
      (以下、省略)

```

まず、上1段・下1段について見てみると、<1010>行において、終止形を基に、その語尾1文字を取り除いた語形を文字変数WL\$に代入し、<1030>行で、それ