

日本DECのAI製品における日本語化について

川倉 道・前端 克典 (日本DEC・ソフトウェア部)

概要

弊社のAI関連製品 (VAX LISP, OPS5, C-Prolog, Quintus Prolog) は、弊社のOSであるVMS (注1) の日本語処理機能の使用により日本語の取り扱いが可能である。本論文では、VMSがサポートする日本語処理ルーチンの説明、各AI製品ごとの日本語処理の特徴、及び将来の展望について述べる。

1. 日本語ライブラリ: JLBについて

JLBは、VMS標準装備のオブジェクト・ライブラリ (注2) である。このJLBは、VMS上で可動するほとんどの言語より呼出しが可能で、文字列の入力時に、日本語変換をユーザが端末を使って行う。例えば、FORTRANからの呼出しは、次のようになる。

```
program kanji_input
c
c   kanji input using JLB#GET_INPUT
c
c   character line*80
c   integer jlb#get_input, length
c
c   call jlb#get_input( line, 'Input> ', length)
c   write( 6, 100) line(1:length), length
c
c   stop ' '
100  format( '1The string is: ',a,'.'/
1      ' The length is: ',i,'.')
end
```

JLB#GET_INPUTというサブルーチンが、日本語変換入力を行う。入力時は変換キーを押しながらの入力となる。ローマ字→日本語変換、かな→日本語変換が可能である。また、汎用入力ルーチンとして設計されており、ファイルからの入力も可能である。

2. 漢字コードについて

弊社の漢字 (日本語) コードは表1のとおりである。この表でもわかるようにMSB (注3) を1にすることにより、ASC E I 9 6 5 4 と区別する。よってASC E I と日本語の混在が可能である。また、弊社の日本語端末 (注4) も、

第 2 バ イ ト

	00	20	40	60	80	AO	CO	EO	FF (16進表示)
第 1 バ イ ト									
	20								
	40								
	60								
	80								
	AO								
	CO	A121	拡張漢字セット			A1A1	DEC 漢字セット		
	EO								
	FF (16進表示)					FE7E	FEFE		

斜線部分は制御コード領域

表 1

このコード体系を採用している。なお、第2水準の漢字、外字の登録機能もJL Bと同じく、VMS標準の装備である。

3. 各AI関連製品の背景

I. VAX LISP

VAX LISP は Common Lisp に準拠して作成されたLISP処理系である。Common Lisp のほとんどの機能(注5)が使用でき、コンパイラ、エディタ、デバッガ、トレーサ、ステップを含んでいる。

日本語の取り扱いは、シンボルと文字列に可能であるが、Common Lispの文字型としては扱えない(注6)。

VAX LISP の核はBLISS(注7)と呼ばれる言語で書かれている。この言語は、VMSの開発言語でもある。C言語とよく似た言語だが、より細かな指定が可能である。

II. C-Prolog

C-Prolog はイシノバラ大学で作られた Prolog である。C 言語で書かれており、DEC-10 Prolog とほぼ完全な互換性を備えている。処理系としては、デバugg、トレーサを備えている。

日本語の取り扱いは、述語名、アトム、文字列に可能であるが、組込述語 name を使用し文字列を分解すると、日本語 1 文字の文字列でも、リストの長さは 2 となる。例えば次のとおり。

```
yes
! ?- name('ABC',X).
X = [65,66,67]

yes
! ?- name('漢字',X).
X = [180,193,187,250]

yes
! ?-
```

III. Quintus Prolog

Quintus Prolog は、米国 Quintus 社が開発した Prolog である。日本では弊社が直接 Quintus 社と提携し、販売している。開発言語は C 言語である。

Quintus Prolog は、開発環境として、インタプリタ、コンパイラ、デバugg、ステップ、エディタ（注 8）を装備している。C-Prolog と同じく、DEC-10 Prolog とほぼ完全な互換性を持つ。また、他言語（注 9）で書かれたルーチンを述語として登録する機能があり、この Prolog の大きな特徴となっている。

日本語の取り扱いについては、C-Prolog と同じである。

IV. OPS5

OPS5（注 10）は CMU（注 11）と米国 DEC で共同開発された、ルールベースの言語（注 10）である。当初は Franz Lisp で記述されていたが、後に BLISS（注 7）で書き直され、処理速度の点でかなり向上された。弊社の VAX-11 システム・コンフィギュレーション用エキスパート・システム XCON は、OPS5 を使用している。

日本語の使用は、ルールの名前、変数名、アトムに可能となっている。これは OPS5 のほとんどすべての部分に使用できるといってもよい。

4. 日本語使用のための変更点について

I. VAX LISP

日本語の処理を可能にするために、入力ルーチン、シンボル生成ルーチン、割り込み処理ルーチン(注12)、及び文字テーブルに変更を行なった。

入力ルーチンは、LISPの関数、ReadとRead-Lineの文字列入力の部分をJLBに変更し日本語の入力を可能とした(注13)。但し入力装置が端末でない場合や、端末の設定が特殊な場合(注14)はJLBは実行されず、通常の入力ルーチンが実行される。

シンボル生成ルーチンは、入力ルーチンより文字列を受けとり、1文字ずつ解釈し、シンボルを作成する。この際、もしMSBが1であれば、無条件で次の文字をシンボルの文字列として追加するよう変更が加えられた(注15)。

VAX LISPはコントロール・キーによる割り込みをLISPの関数として定義できる。割り込み処理ルーチンは、この割り込み関数を処理する。このルーチンでは、定義された割り込み関数に処理を移す前に、JLBによる入力要求を解除するよう変更された。

文字テーブルは、文字コードで128~254(10進数)をDEC特殊文字(注16)として解釈していたが、日本語を使用するために、特別な解釈を与えないよう、変更された。

今後の課題

i) VAX LISP エディタの日本語化について

現在、VAX LISP エディタは日本語をサポートしない。エディタがどの入力にRead、Read-Lineを使わないこと、エディタがシステムとは別に文字テーブルを持っていること、エディタの使用しているVMS標準のスクリーン管理ライブラリ(注17)が、現時点では日本語をサポートしないことが理由である。

それぞれの問題について検討中である。また、一部変更の作業を開始した部分もある。

ii) 日本語の文字型としての取り扱いについて

現在、VAX LISP の文字コードの上限は 256 である。よって、このままでは日本語を文字型としては取り扱えない。しかし、より自由度の高い日本語処理のために、将来は必ず加えるべき機能である。

II. C-Prolog

日本語の取り扱いを可能にするために、入力ルーチン、アトムを生成するルーチン、及び文字の属性テーブルに変更が加えられた。

入力ルーチンは、LISP と同様に、文字列の入力であった部分を JLB に変更し、日本語の入力を可能とした。

アトムを生成するルーチンも、LISP のシンボルを生成するルーチンと同様に、MSB が 1 の文字が現われるば、その次の文字は解釈を行なわずに、無条件でアトムとなる文字列に追加するよう変更されている。

文字の属性テーブルは、0 ~ 255 のコードに対応する文字のみに属性が与えられていたが、これを拡張し、128 ~ 255 のコードを持つ文字は、英子文字と同じ属性を持つよう変更された。

III. Quintus Prolog

C-Prolog と同様に、入力ルーチン、アトムを生成するルーチン、及び文字の属性テーブルに同等の変更が加えられた（説明は省略する）。

今後の課題

- エディタの日本語化について

VAX LISP と同様に、エディタの日本語化が今後の課題である。

IV. OPS5

他の AI 製品と同様に、日本語処理を行うために、入力ルーチン、及びアトムを生成するルーチンに変更が加えられた。

入力ルーチンは、端末からの入力時には、日本語変換を行うために、JLBを実行するよう変更された。それ以外の場合、例えばファイルからの入力時にはJLBは実行されない。これは、VAX LIS Pにおける入力ルーチンの処理と酷似している。

アトムを生成するルーチンも、他のAI製品のものと同様に、MSBが1の文字が現われた場合は、無条件で次の1文字をアトムを構成する文字列に加える、という変更が加えられた。

5. 共通の概念

各AI製品において、日本語処理を可能にするために加えられた変更には、次の共通点がある。

- ① 入力にはJLBを使用する。
- ② MSBが1の文字が現われた場合、次の1文字は解釈せず、無条件で追加する。
- (③ 1文字は必ず8ビットより成る。)

注1 弊社のハードウェアVAX-11用に開発されたOS。日本語処理機能は、日本で作られた。

注2 オブジェクト・ファイルの集り。リンクにより、必要なオブジェクトの部分のみ実行可能ファイル中に結合される。

注3 Most Significant Bit 略。このビットを1とすると、そのバイトの値は128以上になる。

注4 VT 80, VT 282 が挙げられる。

注5 現在は、一部のヘルプ機能、及び複素数型がサポートされていない。

注6 現在は、'#\' (文字型を表すシャープ・マクロ) で日本語は使えない。また、次の形式を評価すると答えは2となる。

(length "語")

- 注7 弊社のハードウェア用に作成されたコンパイラ。VAX用の他に、DEC SYSTEM-10, 20, PDP-11用のBLISSコンパイラがあり機種に依存しないようプログラムすることも可能。VAXではシステム開発言語として位置付けられており、VMS自体ほとんどがこの言語で記述されている。
- 注8 EMACSエディタが使用されている。但し、Quintus社がPrologとのインターフェースを考慮し変更を加えている。
- 注9 C言語, FORTRAN, COBOL, Pascal, Basic, 及びPL/1。
- 注10 VMS上で動作するコンパイラ言語とリソースをとることも可能。
- 注11 カネガタ・マロン大学
- 注12 現在は、コントロール・キーによる割り込みのみをサポートする。例えばコントロール・Cの入力により、トップレベル・ループに戻る、など。
- 注13 トップレベル・ループでは関数Readが実行されるので、日本語入力が可能。
- 注14 オペレータのキー入力をユーザのプログラムで処理するためのモードがある。
- 注15 表1で、拡張漢字セットの部分は、2文字目のMSBが0となる。よって1文字毎のMSBのチェックだけでは、もしMSBが0のときは、その文字が(例えば'('などの場合)1ストの始まりを表すのか、それとも、拡張漢字セットに含まれる日本語の2文字目なのか、を判別できない。
- 注16 ウムラウトやオンカストロームなどを含む。
- 注17 一般のキャラクタ・ディスプレイ(例えば、弊社の端末であるVT100など)を用いてウィンドウを処理するサブルーチン群。