

## 7. L I S P†

黒川 利明\*\*

## 1. はじめに

文献1)で述べられているように、LISPは1960年代初頭に生まれた古参言語の1つである。しかし、LISPに対する興味自体は近年に至って盛んになっており、本稿でも最近の動向を中心に述べる。

一応常識的なことを述べると、LISPはLISt Processorの略で、字義は「リスト処理用プログラミング言語」である。Newell, Simon, ShawのIPL<sup>16)</sup>がリスト処理の先駆である。ここでリストというのは図-1のようなポインタでつながれた2進木のことを指している。

LISPはLots of Insipid Stupid Parenthesesの略だという冗談があるように、括弧を用いたポーランド記法が現在のLISPプログラム表記の主流である。たとえば、 $3+x*4$ を(PLUS 3(TIMES X4))というふうに表記する。これをS式(Symbolic expression)と呼ぶ。歴史的経緯からいうと、S式は本来はデータ表記法で、プログラムはM式という、よりALGOLに近い形式で記述されていた。このような変遷は後述するLISPの現代的風潮にもつながっている。

LISP言語の詳細については教科書<sup>12),13)</sup>や手近なマニュアルを参照していただいた方がよい。あとで述べるように細部についてはシステムによって変動があるから。

本来のLISP、一般にはリスト処理、の応用目的は人工知能と称される分野であった。

プログラミング言語としてのLISPは、この応用分野におけるハードウェアの変化や問題意識の変遷に従い、多大の影響を受けてきている。すなわち、LISPの評価は、ほとんど人工知能の評価と軌を一にしており、LISP処理系の大部分は人工知能(あるいはそこから派生した数式処理や知識工学など)の研究のため

に開発、導入されている。

今日、ハードウェアの長足の進歩により、かつてLISPの最大の難点とされていた処理速度や占有記憶量は(技術的に興味のある問題だが)実用的に大きな障害とは考えられなくなり、むしろLISPのもっている長所(会話機能、数学的性質、記号処理の容易性など)が大きくクローズ・アップされている状況は、人工知能の研究にもあてはまるように見受けられる。

以下、この計算機技術の過渡期という認識のもとづいてLISP言語の動向を述べてみたい。

## 2. 最近の動向

日本の動向については若干古い文献<sup>2)</sup>にまとめがある。国際的な動向については昨年のLISPコンファレンスの会議録<sup>3)</sup>が役に立つ。

まずLISPの基本的な特徴をまとめてみる。

言語仕様上のLISPの特徴は以下の通りである。

## ① リストというデータ型の存在

リストは抽象データ型では次の3操作、**car**, **cdr**, **cons**の満たす次の関係で定義される。

$x$ はリストである  $\iff \text{cons}(\text{car}(x), \text{cdr}(x)) \equiv x$

## ② プログラムがリストで実現されている

これはいい換えれば、リストを「プログラムとして」実行する関数evalが存在することである。

## ③ プログラムが関数型式で表現されている

(<関数名> <引数>...)という括弧を用いたポーランド記法である。

言語システムとしては次のような特徴がある。

## ④ データ領域の自動管理(ガーベジ・コレクタ)

リストはガーベジ・コレクタ(GC)によって、自動的に再生使用されるので、データ領域はユーザが管理する必要がない。

## ⑤ 文字アトム(識別子)の属性リスト

LISPの文字アトム(識別子 identifier)には属性リストが付随している。この属性リストは文字アトムに対して任意の属性と属性値をセットしたり、参照した

† LISP by Toshiaki KUROKAWA (Information Systems Laboratory, TOSHIBA R and D Center).

\*\* 東京芝浦電気(株)総合研究所

りすることができる。

以上が LISP 言語の最低限の特徴といえるだろう。もう一項目、特にユーザの立場から付け加えるならば、

⑥ 「LISP プログラム」が稼動する  
を挙げることができる。

この項には多少の注釈がある。第1に LISP には現在のところ標準規格が存在していない。LISP 1.5 Programmer's Manual<sup>4)</sup> は、LISP の1つの原型を与えてはいるが、現実の LISP はかなり食い違っている。第2に LISP インプリメンタの大部分は独自のシステムを設計、製作する傾向をもっている。この傾向の由来については後に論じたい。第3に LISP ユーザは他人の「LISP プログラム」を利用する傾向をもつ。

このような状況の中で、「LISP プログラム」が任意の LISP システムで稼動することができるのは、各 LISP システムが基本的な特徴を共有しているので、LISP プログラムの移植は比較的容易なこと、また自動化も困難ではないことによっている。

以下に LISP の最近の動向を示す話題を追う。

#### (1) LISP マシン

言語の話題の中にハードウェアを取り上げるのは奇妙かもしれない。しかし最近の LISP についての議論の多くは LISP マシンを離れては存在しない。

LISP マシンが重要なのは、従来 LISP 言語の最大の欠陥が、処理速度が遅く、主記憶を大量に消費するという実用的な性能にあるとされてきたからである。

LISP マシンのアイデアは CMU の C. ai<sup>5)</sup> に逆上るが、実質的には 1978 年の MIT の提案書以降に始まったといって良い。すでに MIT では 1 号機 CONS に続いて CADR を開発しており、米国内ではほかに、Xerox PARC, BBN, ユタ大学で開発が進んでいる。特に Xerox PARC では、Alto マシンの後継機種として Dorado, Dolphin という 2 種類の計算機を開発している。

MIT および Xerox の LISP マシンの特徴は、①パーソナル・コンピュータとして位置づけられていること、②高解像度のディスプレイを備えていることである。

日本においては、1974年に青山学院大の ALPS<sup>3)</sup> マシンが LISP マシンとしての先鞭をつけたのであるが、各所での開発は MIT の提案書以降であった。具体的には神戸大学、京都大学、理化学研究所、電電公社武蔵野電気通信研究所、慶応大学、大阪大学、電子

技術総合研究所などで LISP マシンの開発が開始された。

LISP マシンの重要性は従来の LISP 言語につきまとうっていた不満の解消とパーソナル・コンピュータという新しい計算機システム概念の実現にあることはすでに述べた通りである。言語技術的にも幾つかの興味ある問題を LISP マシンに提出している。

① 実時間(または並列)ガーベジ・コレクタの問題  
これは各 LISP マシンで種々の技法が試みられている。並列 GC についての日比野の実験<sup>6)</sup>は興味深い。

#### ② プログラムの実行手法の問題

文脈処理(context switching mechanism)がハードウェアのサポートにより純ソフトウェアによる方法より効率的に実現されるので、いくつかの方法が提案されている。

#### ③ 並列処理の問題

データフロー・アーキテクチャへの一般的関心やハードウェア価格の低下などにより、LISP の並列処理が最近注目を浴びている。阪大の安井研ではハードウェアの製作にとりかかっている<sup>4)</sup>。

#### ④ ユーザ用の構文の問題

LISP マシンでは、ユーザ入力を実時間で翻訳して処理することが可能なので、かならずしも従来の LISP 表記法に従う必要はない。これまでに議論されてきた各種の構文要素を取り入れたり、pretty reader のようなサポートを追加することができる。

#### ⑤ データ構造の問題

リスト・データの抽象データ型定義を保存して、アクセス、メモリ量の両観点から効率の良い構造化が提案されてきた。

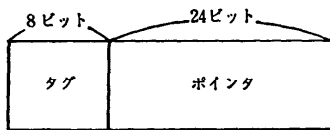
たとえば図-1のようにリストのデータにタグを付けて (A, B, C) というリストを、従来の 6 ポインタから 3 ポインタに圧縮される方法 (cdr coaling) が MIT の LISP マシンでは取られている。

従来の LISP データのほかに、tuple あるいは vector と呼ばれるような新しいデータ型を LISP に導入することも真面目に検討されるようになった。

#### (2) ユーザ層の変化

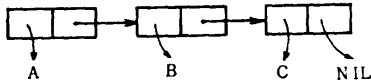
LISP 言語には従来から、ユーザと設計者との間に緊密な情報交換の習慣があった。LISP システムが現在のように多様化している一つの理由は、ユーザの応用分野や好みに合わせ、LISP 言語が改良されたためである。

このユーザ層は従来はいわゆる人工知能研究者に限



タグ情報として、以下のようなものがある。  
 cdr 部は次のアドレスを指す。  
 cdr 部は NIL。  
 cdr 部の内容は次のアドレスに書いてある。

(A, B, C) の従来の表現



(A, B, C) の cdr-coding による表現

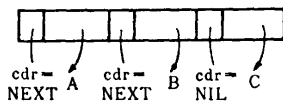


図-1 リスト・データの表現法

定されていた。しかし、最近ユーザ層が大きく変化しつつある。新しいユーザ層の動きとしては、以下のようなものが挙げられる。

#### ① マイコン・ユーザの出現

LISP 言語の単純さと対話的操作性の良さは、マイコン・ユーザから好ましい評価を受け始めている。メモリの制約のために大規模な LISP 応用プログラムはまだマイコン上では動かせないが、LISP の使用によってマイコン・ユーザに新しい可能性が開かれていることは疑いない。

#### ② 応用システム・ユーザの定着

LISP を使っていることを知らない（あるいは意識しないで済む）ユーザが徐々に定着しつつある。従来も数式処理の MACSIMA や REDUCE、質量スペクトル分析の DENDRAL などのユーザは、自分が LISP を使っていることを意識する必要はなかった。

知識工学の出現によって、LISP は知識ベース・システムの基礎言語（システム記述言語）としての性格を帯びるようになった。これらの知識ベース・システムのユーザは、自分が LISP を使っていることを知らなくとも良い。（もちろん、知っていた方がシステムの利用や改良に便利ははずだが。）

#### (3) LISP の評価原理について

LISP の単純さがラムダ計算法 ( $\lambda$ -calculus)<sup>7)</sup> に基づいた関数型の評価方式にあることは明らかである。

ところで、LISP の評価方式はラムダ計算法と同一

ではない。ラムダ計算では「置き換え規則 (rewriting rules)」によって計算を進めるのだが、LISP では値を変数に与えることによって計算してゆく。

したがって LISP の計算方式ではラムダ計算では生じないような問題が発生してくる。このため各種の評価方式が新たに提案されている。

新しい評価原理としては、Steele, Jr. らの SCHEME<sup>8)</sup> Cartwright の typed LISP<sup>9)</sup>, Kurokawa の function class<sup>10)</sup>などを挙げることができる。

これらは LISP における変数の処理やデータ宣言の処理、関数実行形式の変更などを含み、従来の LISP プログラムの大部分はそのまま動かせるようになっていく。

LISP の方言という範囲からは逸脱するが、actor 理論に基づいた Kay らの smalltalk も<sup>11)</sup> LISP の評価方式の考え方に大きな影響を及ぼしている。

### 3. 今後の動向

#### (1) 規格化

米国防省で最近 LISP の規格化を検討し始めたという噂がある。検討したがあきらめたいという説もある。

プログラミング言語の規格化は、プログラムの流通を促進し、プログラミング言語の教育を容易にするので、プログラミング言語の普及には不可欠である。

しかるに LISP インプリメンタには規格化に反対する意見が多く、ユーザのなかにも規格化に反対する人が少なくない。

LISP 言語の関係者が規格化に反対するのには、それなりの理由がある。

#### ① 規格化は言語の発達を阻害する

LISP の最近の傾向についての項目を見ても推察できるように、LISP 言語の「方言」という幅の広さは、とても PASCAL や BASIC の比ではない。

元祖 LISP 1.5 は現在では古典ともいべき存在であり、LISP 1.5 で書かれたプログラムがそのままの形で稼動する LISP システムは余りない。しかも assoc や get のような比較的基本的と目される関数の仕様すら LISP システムによって異なる。

この LISP の多様性は多くの場合、ユーザの要求とインプリメンタの工夫とから生じている。先程述べたようなデータ構造そのものへの変更とか、評価方式自体の改良までが、LISP という土俵の上で議論されるのは、このような LISP の改良を全体としては好まし

い傾向として受け入れる体制が、LISP の開発者、利用者の双方にできているからである。

### ② 規格化は不可能である

この理由は現在インプリメンタの数だけシステムができていて、しかもその変化が大きく、おまけに①の理由で規格化への熱意が少ないという実際の認識からきている。

### ③ 規格化は不必要である

規格化の目的である流通と教育に関しては現状で十分である。移植が比較的容易だし、一つの LISP 言語を習得すればほかの LISP 言語も容易に扱えるという意見である。

実際、LISP ユーザは大抵この種の経験をしてきている。現状で LISP プログラムがかなり流通しているのも事実である。

以上のような規格化への反対意見のある中でお、規格化が議論されるのは、やはりユーザ層の変化である。新しいユーザ層にとって現在の LISP の混乱は迷惑なものである。言語の発達を促進するような規格化の議論が必要となろう。

### (2) LISP マシンの開発

すでに2社が LISP マシンを発売しているが、この傾向はさらに広がるだろう。知識ベース・システムの需要から90年代のパーソナル・コンピュータは LISP マシンで占められるという予想もある。

### (3) 他言語との関係

いわゆるコンパイラ言語との関係は今後もあまりないだろう。むしろ、パーソナル・コンピュータ上での対話的言語、特にグラフィックスを取り入れた SMALL TALK のような言語との関係が深まるだろう。その結果、表面上は現在の LISP とは全く異なった LISP が広まる可能性もある。

### (4) 応用分野

LISP の応用分野が主として記号処理、人工知能と呼ばれる分野であることは間違いない。ただ、ユーザの認識として、科学技術計算においても事務計算においても記号処理や人工知能的な部分を含んだ計算（記号計算？）が今後は要求される。向う4,5年の有望な応用分野としては、VLSI 設計、数式処理、コンサルテーション・システムが挙げられている。将来的には自然言語処理を含んだオフィス・オートメーションが最右翼であろう。

(注) LISP プログラムの移植性について

本稿の査読者から LISP の移植性について複数のコ

メントをいただいた。LISP は方言が多いにも拘わらず、移植性が良いというのは LISP ユーザの認識である。移植手法としては、変換プログラムを作る方法とエディタを用いた会話的な (incremental と称される) 手法とがある。

方言はあっても根幹が同じであること、会話的プログラム修正が容易なこと、当面使わない関数は未定義でも構わないこと、システム関数（組込関数）の再定義が容易なことなどが挙げられる。

他言語との比較は簡単ではないが、個人的印象としては JIS 規格のある FORTRAN より良いと思っているのだが、いかがであろうか？

### 参 考 文 献

- 1) McCarthy, J.: History of Lisp, SIGPLAN Notices, Vol. 13, No. 8, pp. 217-223 (Aug. 1978).
  - 2) Kurokawa, T.: Lisp Activities in Japan, Proc. 6th IJCAI (Aug. 1979).
  - 3) Conference Record of the 1980 LISP Conference, the LISP Conference (1980).
  - 4) McCarthy, J. et al.: LISP 1.5 Programmers Manual, MIT Press (1965).
  - 5) Barbacei, M. et al.: C.ai...A LISP Processor for C.ai, CMU-CS-71 (Aug. 1971).
  - 6) Hibino, T.: A Practical Garbage Collection Algorithm and its Implementation, Proc. Computer, Architecture Conference, pp. 113-120 (1980).
  - 7) Church, A.: The Calculi of Lambda-conversion, Annals of Mathematics Studies, Princeton Univ. Press (1941).
  - 8) Steele, G. Jr. and Sussman, G.: The Revised Report on SCHEME, A dialect of LISP, AI Memo 452, MIT (Jan. 1975).
  - 9) Cartwright, R.: A Constructive Alternative to Axiomatic Data Type Definitions, in 3).
  - 10) Kurokawa, T.: The Function Class, in 3).
  - 11) Kay, A. and Goldberg, A.: Personal Dynamic Media, Computer, pp. 31-41 (Mar. 1977).
- 以下に最近の LISP の教科書などを挙げる。
- 12) 中西: LISP入門——システムとプログラミング 第2版, 近代科学社 (1980).
  - 13) Winoton, P. H. and Hom, B. P.: LISP, Addison Wesley (1981).
  - 14) Charniak, E., Riesbeck, C. and McDermott, D.: Artificial Intelligence Programming, Lawrence Erlbaum Associates (1979).
  - 15) Stoyan, H.: LISP, 東独から出版中 (?) (1980).
  - 16) Newell, A. and Shaw, J. C.: Programming the Logic Theory Machine, Proc. of the WJCC (1957). (昭和56年1月26日受付)