John McCarthy et al. : LISP 1.5 Programmer's Manual

The MIT Press（1962）

1960 年代から 1980 年代にかけて LISP が人工知能分野の「標準語」であったことは，あらためて述べるまでもあるまい．このマニュアルは，LISP が普及する大きなきっかけを作り，1970 年代から 1980 年代にかけての人工知能研究の主役をつとめる MacLisp や Interlisp の礎となった LISP1.5 について述べたものである．

“Programmer's Manual”と銘打たれてはいるが，現代の視点から見ると，言語の設計，システム利用者のための手引，さらには特定のコンピュータ上の処理系の構成までを含んだ，はなはだ整理の悪いものである．近年では，言語の設計者と利用者との距離は開き，言語の設計と処理系の設計の間ですら必ずしも近くない場合が多いのに対し，コンピュータの利用者がまだ限定されており，言語と処理系と利用者とが混沌の中にあった時代背景を映し出している．

筆者がこの本を初めて目にしたのは，1970 年代半ば，学部学生の「工場実習」で，電総研の LISP システムの開発にあたっていた黒川利明氏もいらした東芝総合研究所のあるセクションを，数週間訪問する機会を得たときだった．Fortran しか知らない学生は，最初は面食らうばかりだったが，2 週間後には単一変数の有理式の微分，簡約化と，それらしい表示を行うシステムを作り，実習レポートとすることができた．簡潔で合理的な言語仕様と，高い生産性の賜物で，Fortran ではこうはいかなかったろう．その後卒業研究から博士課程まで，研究対象が LISP 一辺倒になったのは，これがきっかけだった．

LISP が及ぼした影響は多岐にわたる．まずデータを記号的に扱うという考え方自体が斬新だった．当時普及していたのは，Cobol，Fortran，Algol を典型に，数値データに対する演算を記述するための言語ばかりだった．これに対し，LISP は「記号アトム」という，なんらかの概念に名前をつけただけの，数値とは無関係な対象についての計算を主とする言語である．記号は同一か否かだけに意味があり，加減乗除に代わり条件判断が主役となる．このような「計算」がそれ以前に知られていなかったわけではないが，LISP は「計算」に対するこの見方を普及させる原動力となった．

「プログラム内蔵型」という概念を復権させたのも，LISP の果たした大きな役割の 1 つである．当時普及していた言語では（そして現在も多くの言語の主要な使い方では）プログラムは処理対象のデータとは独立した存在で，記述形式もまったく異なる．その限りでは，プログラムとデータを同じメモリ上に置くことには効率上のメリットしかない．これに対し LISP では，プログラムと処理対象データを，二分木を基礎とする同じ形式（S 式）で表現し，両者がある意味で同質のものであることを強く意識させる．このマニュアルでも，LISP プログラムをデータとして扱って実行するメタインタプリタを appendix の 1 章を割いて記述している．いわば高い抽象度の von Neumann 機械を与えているのである．もちろんその背後にラムダ計算という形式的枠組みがあるわけだが，それはきっかけでしかないようにさえ思える．

この高抽象 von Neumann 機械は，機械語命令の実行機構の代わりに，S 式プログラムの実行機構 eval を持つわけだが，その重要なポイントはメモリの仮想化にある．メモリを必要に応じて割付け，不要になれば自動的に回収・再利用するガーベジコレクション機構が，この機械には必須な前提である．これによって，複雑に絡み合うグラフ構造を用いても，プログラムは面倒なメモリ管理から解放される．複雑なデータを扱う局面の拡大に伴い，ガーベジコレクションは今や言語システムの重要な機能の 1 つとなってきた．

処理系にまで至る記述のおかげで，これさえ読めば処理系を作れる．これは，後続の多くの処理系の作成者にとってたいへん有益であった．逆に，具体的な機械べったりの記述を含むことが，LISP の本質を見失わせる面もあったかもしれない．しかし，こうした中から本質を見出すことを強いた記述が，次の時代の技術の芽吹きを促したのではないか，混沌こそ新たな展開の必須要件ではないか，とも考えさせられる．

（平成 16 年 10 月 21 日受付）

近山　隆／東京大学新領域創成科学研究科
chikayama@logos.ic.i.u-tokyo.ac.jp