

HLISPのImplementationとそのApplication

駒木, 悠二 / KOMAKI, Yuji / 福馬, 敏子 / 蜂谷, 博 /
HACHIYA, Hiroshi / FUKUMA, Toshiko

(出版者 / Publisher)

法政大学工学部

(雑誌名 / Journal or Publication Title)

Bulletin of the Technical College of Hosei University / 法政大学工学部研究集報

(巻 / Volume)

13

(開始ページ / Start Page)

1

(終了ページ / End Page)

12

(発行年 / Year)

1977-03

(URL)

<https://doi.org/10.15002/00004197>

HLISP の Implementation とその Application

駒木 悠二*・福馬 敏子**・蜂谷 博***

Implementation of HLISP and Application

Yuji KOMAKI, Toshiko FUKUMA and Hiroshi HACHIYA

Abstract

LISP (for LIST Processor) programming system has been developing in M. I. T. and so many universities, research institutes, makers, since John McCarthy had proposed first.

In researching of software-algorithm, non-numerical process problem, symbolic data process problem, theorem proving, metacompiler, network analysis, game, simulation, answers and questions system, etc. LISP is effectual tool from its own characteristics among various kind of information processing languages.

Especially LISP is indispensable language in researching the algorithm of theorem proving, artificial intelligence and intelligence of robot.

We implement HLISP (Hash coded LISP) on FACOM 230-45 S computer system of Hosei University Computing Center, and we also introduce some applied examples and evaluations of ability in order to contribute to study of computer science.

§1. はじめに

HLISP は LIST 処理語 LISP の拡張として東京大学情報科学研究室（後藤英一教授）で HITAC 5020 計算機システムによって当初作成された。その後 HITAC 8800/8700 システムに改良版がインプリメントされ、現在はさらにその改良3版とも言うべき版が稼動している。

この HLISP 第2版を Base にわれわれは、法政大学計算センターの FACOM 230-45 S に Implement した。東京大学大型計算センターでは、第3版が Batch の他 TSS モードでも稼動できるとの事であるが、本学の F 230-45 S のリソースは TSS 用の端末装置を持たないために、Batch でのみ稼動できる。

ここでは LIST 処理言語の開発とその背景、HLISP の特徴、HLISP の Implementation、HLISP の Application について述べる。

* 経営工学科助教授
** 経営工学科実験助手
*** 計算センタープログラマ

§2. LIST 処理言語の開発とその背景

MIT の John McCarthy によって 1960 年 LIST 処理言語 LISP が提案¹⁾され、1962 年に MIT の IBM 7090 に LISP 1.5 としてはじめて開発された²⁾。わが国においても、1963年に早稲田大学生産技術研究所で紹介され、慶応大学 (TOSBAC-3400; KLISP) 東京大学、電機大学 (NEAC-2230; LISP 1.5) などの大学や、日立、富士通、東芝、電総研などでも LIST 処理言語「LISP」が作製されてきた。

米国での LISP システムの開発は Raphael³⁾、Bobrow⁴⁾、Moses⁵⁾ などの仕事をもとにして、人工知能研究や記号処理システム作成の Tool として重視され、MACLISP⁶⁾、Stanford LISP 1.6⁷⁾、BBN-LISP⁸⁾、INTERLISP⁹⁾などに発展してきた。さらに高級言語である LISP と Machine Language 又は Assembler との間隔をうめるために、L⁸ (エル・シックス)¹⁰⁾などの低レベルのリスト処理言語、FORTRAN 言語との Linkage を考慮された SNOBOL および、SLIP, IPL, LEAF, LPL, BALM, LISPA¹¹⁾ など多様な LIST 処理機能を重視した言語が多く開発されてきた。さらに LIST 処理機能は、汎用言語 PL/I にもとりいれられている。

一方わが国においては LISP を「おもちゃ」の言語とみなす傾向が強く、それはわが国の計算機が、研究者にとって米国に比べ高価であり、LISP システムを開発あるいは利用するために、実用的に十分なメモリと高速で大型の計算機資源が与えられなかったためではあるが、結果的にはわが国では LISP 1.5 からの改良は、慶応大学などわずかな例にとどまった。HLSIP はそのわずかな例の一つであり、LISP 1.6 の機能をほぼ包含するなど、特異な存在である。

近年わが国のコンピュータも大型化し、価格も昔に比べ安くなり、高いコスト・パフォーマンスが得られるようになるにつれ、国外からの刺激もあって、最近ではわが国でも相ついで LISP システムの研究、改良、LIST アプリケーションの研究が活発になった。特に大容量のメモリや TSS 機能の実用化は、LISP システムをより実用化し、人工知能研究および人工知能向き高級言語の開発を加速化させた。その一つの例は Micro-Planner や Planner に見られる。このように LISP に対する評価はわが国でも見なおされ、その需要は質、量ともに増大している。

法政大学計算センターでもこれらの背景のもとに、今後 LISP の需要は増大するものとみられ、今回の HLISP の Implementation はそれにこたえ得るものと思われる。

§3. LISP および HLISP の特徴

LISP は binary Tree (2 進木) を特徴とする内部表現形式で処理を行い、これによってリスト構造の処理を可能とする。そのために、人工知能や、自然言語処理に必要ないくつかの特徴を持っている。

- (1) プログラムとデータが全く同じ形式 (リスト構造) をしている。

- (2) 記号を主体としたデータ・ベースを自由に扱える。
- (3) プログラムのリカーシブ・コール（再帰呼び出し）が可能。
- (4) 自由なデータ構造のデータ・ベースが扱える。そのため、複雑なデータ構造を扱いやすく、又複雑な属性をもつデータの取扱いが容易である。
- (5) データ・ベースの管理、特に不要となったデータの処理が自動的に行われる。（ガーベジ・コレクション）
- (6) 本質的にインタプリタによる実行であるため、会話形の処理に向いている。
- (7) スtring処理が FORTRAN などに比べ扱いやすい。
- (8) 文の構造が簡単であるため、プログラム学習が容易である。（反面、慣れるまでは理解しにくいところもある。）
- (8) Syntax が単純でエレガントである。

これらの特徴によって、LISP は人工知能、自然言語解析、自動翻訳プログラム、記号処理、オペレーティング・システム、プログラム言語（言語処理システム、コンパイラなど）、定理証明、メタ・コンパイラ、非数値処理問題、数値非数値混在問題、数式処理（微分、積分、ラプラス変換）、電子回路理論、迷路探索（その応用としての PERT, NETWORK など）、質問応答システム（Q-A System）、パターン・マッチング、String処理、情報検索などの研究の Tool となり、それらのアプリケーションが発表されている。

人工知能や、言語学、言語処理、情報検索のアルゴリズムの研究には特に不可欠で、ロボットの頭脳のソフトウェアにはほとんど LISP が関与している。

HLISP の特徴はこれら LISP の基本機能を持ちつつ、次の点に要訳される。（FACOM-230-45 S HLISP）

- | | |
|------------------|------------|
| (1) free storage | 11000 cell |
| そのうち H領域 | 7307 cell |
| L領域 | 3693 cell |

- (2) Stack 512個

但し Virtual storage 機能と、Garvage collection 機能とにより実質上無限。

- (3) 数値： $\left\{ \begin{array}{l} \text{整数} \quad -9999 \sim +9999 \\ \text{これを超える値は mod 1000 にとられる。} \\ \text{実数} \quad \text{取扱わない。} \end{array} \right.$

- (4) LISP 1.6 の機能をほぼ包含する。

- (5) Monocopy feature および Assocomp feature を持つ。

（単写機能と連想計算）

- (6) System file による virtual storage 機能
- (7) Grand Gabage Collector の追加。
- (8) 関数の追加
 - HASSOC, ASSOCCOMP, DZEROP, HCONS, HEADER, HDEFINE
 - mod 関数 MDCD, MDIN, MDPN など。
 - 関数をシステム・ファイルに登録する関数 BANK, FF など。
- (9) インター・プリタ方式であること。
- (10) 主として FORTRAN で書かれ, 7400 ステートメントに及ぶ。そのため他のコンピュータ・システムに Implement する事が容易であり, システムの互換性が大きい。

§ 4. HLISP の Implementation

4.1 HLISP の Run に必要なリソースの最小構成

本計算センターの FACOM 230-45S システム OS II の制御下で稼動する。

- (1) メモリー; ユーザ・エリア 118 KB 以上 (59 page), システム・レジデンス 56 KB
とすると, 合せて 174 KB 必要となり, 最低のメモリ容量は 192 KB
- (2) ファイル; U 20 ファイル (HLISP System file) 143 trk (但し, ディスク・バック装置 F 472 L を使うものとする)
 - VS 機能の処理
 - EXPR, FEXPR を S-Expression で格納
 - SUBR, FSUBR の関数—内部番号表を格納
- (3) カード・リーダー又はそれに代る入力装置
- (4) プリンタ

4.2 Implementation

HLISP はソース・プログラムを本学用のものに修正し, 内容としてはもとの LISP 文法をそこなわず同機能のものとした。

また東大大型計算センターの HITAC 8800/8700 HLISP とも互換性を保ち, 他の F 230-45S HLISP (例えば電機大), 青山学院大 IBM 370-135 HLISP とも互換性を保っている。

ソースプログラムは合計 7400 ステートメントあり, その内わけは表 2 に示す。HLISP 本体は S. HLISP および, S. EVALQT と云うブロックにあり, それぞれ, 2288行, 22サブルーチン, 2379行, 1サブルーチンから構成されている。この他に, S. BLK 368行, 4サブルーチン, S. VT, 2015行, 28サブルーチン (Virtual Tape 機能), S. FCB 52行, 1サブルーチン (ASSEMBLER, Virtual Tape の I/O), S. IFUNCS, 52行1サブルーチン (bit handling), S. INISET 325行, 1サブルーチン (LISP, Initial set cards) から構成されている。

EBは、主記憶上 59 page (118 kB) と、U20 file 143 track を要する。

さらに補助的な作業としてジョブ制御マクロの作成、システムファイル(U20)の登録と初期化、ELIBの登録などを行った。

§5. LISP の Application

次に LISP のアプリケーションプログラムのいくつかを紹介する。これらの作製と、Run Time の測定によって HLISP のおよそ性能比較ができるものとする。

- (1) 因数分解のプログラム
- (2) 微分・積分の数式処理プログラム
- (3) ゲームのプログラム

5.1 因数分解プログラム

このプログラムは、数式処理問題のモデルとしてとり上げた。FORTRAN で因数分解を行う場合には、数値計算問題として解くのであるが、LISP では関数の再帰呼出機能を利用した解法によるものである。

5.1.1 アルゴリズム

整式 $A(x)$ を $P(x)$ で割った時の商を $Q(x)$ 、余りを R とおくと、これは次の式で表現される。

$$A(x) = P(x) * Q(x) + R \quad (5.1)$$

$R=0$ のとき整式 $A(x)$ は $P(x)$ で割りきれ、 $A(x)$ は $P(x)=0$ に含まれる根をもつ。何故ならば $P(x)$ を一次式 $(x-\alpha)$ とすると、式 (5.1) は、次のように表現できる。

$$A(x) = (x-\alpha) * Q(x) + R \quad (5.2)$$

この時 $R=0$ ならば $A(x)$ は $x-\alpha$ で割りきれ、 $x=\alpha$ なる根をもつ。

[因数定理によると、整式 $A(x)$ が $x-\alpha$ で割りきれられるための必要十分条件は $A(\alpha)=0$]

$$A(x) = a_1x^{n-1} + a_2x^{n-2} + \dots + a_{n-1}x^1 + a_n \quad (5.3)$$

$$P(x) = (x-\alpha) \quad (5.4)$$

$$Q(x) = b_1x^{n-2} + b_2x^{n-3} + \dots + b_{n-2}x^2 + b_{n-2} \quad (5.5)$$

とおくと

$$A(x) = P(x) * Q(x) + R = (x-\alpha) * Q(x) + R \text{ より}$$

$$A(x) = (x-\alpha) * (b_1x^{n-2} + b_2x^{n-3} + \dots + b_{n-2}x + b_{n-1}) + R$$

係数比較によって、

$$\begin{aligned} a_1 &= b_1 & \therefore a_1 &= b_1 \\ a_2 &= -\alpha \cdot b_1 + b_2 & \therefore a_2 + \alpha \cdot b_1 &= b_2 \\ a_3 &= -\alpha \cdot b_2 + b_3 & \therefore a_3 + \alpha \cdot b_2 &= b_3 \end{aligned}$$

$$\dots\dots\dots \qquad \qquad \qquad \dots\dots\dots$$

$$a_n = -\alpha \cdot b_{n-1} + R \qquad \qquad \qquad \therefore a_n + \alpha \cdot b_{n-1} = R$$

であるから $R=0$ のとき整式 $A(x)$ は $(x-\alpha)$ で整除される。

したがって

$$\begin{array}{rcccc} a_1 & a_2 & a_3 \cdots a_n & (\alpha_1 \\ \vdots & & & \\ \vdots & \alpha b_1 & \alpha b_2 & \alpha b_{n-1} \\ \hline \vdots & & & \\ b_1 & b_2 & b_3 & R \end{array}$$

以上の方法を LISP でプログラミングした。

5.1.2 データの入力形式

一般形 関数名 ((リスト))

例 INSU ((a_1 a_2 $a_3 \cdots a_{n-1}$ a_n)) a : 整数

INSU ((3 4 5 6 \cdots 8))

(5.1.2) 出力形式

一般形 (α_1 α_2 $\alpha_3 \cdots \alpha_n$)

(α_1 α_2 $\alpha_3 \cdots \alpha_n$)(α_{21} α_{22} $\alpha_{23} \cdots \alpha_{2n}$) \cdots ($\alpha_{m1} \cdots \alpha_{mm}$)

(5.1.3) 例

外部表現	入力形式	出力形式	結果の外部表現
$x^4 + 4x^3 + 6x^2 + 4x + 1$	INSU((1 4 6 4 1))	(1 1)	$(x+1)^4$
$x^2 + 4x^2 + 4$	INSU((1 4 4))	(1 2)	$(x+2)^2$
$x^2 + 5x^2 + 6$	INSU((1 5 6))	(1 2)(1 3)	$(x+2)(x+3)$
$x^3 + 6x^2 + 11x + 6$	INSU((1 6 11 6))	(1 2)	
$8x + 12$	INSU((8 12))	(4(2 3))	$4(2x+3)$
3	INSU((3))	NIL	—
	INSU((10010001))	NIL	—

5.2 微分・積分プログラム

(5.1) 因数分解と同様、数式処理のサンプルとして、微分・積分プログラムをとりあげた。数値積分は FORTRAN でも扱うことができるが、不定積分は数式処理そのものであり、LISP のサンプルとしては典型的なものと言える。ここで紹介するプログラム INTEGRAL QF LISP は大学教養課程 1 年生程度の不定積分を解くことができる。

微分・積分プログラムは、微分のサブモジュールとそれを含む積分のプログラムとから構成される。不定積分の処理には必然的に微分が使われるため、従ってこのプログラムでは微分も可能である。

次にこのプログラムで処理を行っている主な変換を公式の形で示す。

5.2.2 アルゴリズム

$\int (3x^2+5)dx \dots (5.10)$ をリスト記法では次のように記述する。

$$(\text{INTEGRAL } (\text{PLUS } (\text{TIMES } 3 (\text{EXPT } X 2)) 5)) \quad (5.11)$$

ここで Args of INTEGRAL すなわち E は

$$(\text{PLUS } (\text{TIMES } 3 (\text{EXPT } X 2)) 5) \quad (5.12)$$

である。まずすべての積分する $f(x)$ すなわち E について $(\text{CAR } E)$ を実行する。

この例では (PLUS) となり、次にこのプログラム上で

$$((\text{EQ } (\text{CAR } E) (\text{QUOTE } \text{PLUS}))) \quad (5.13)$$

を捜し、

$$\begin{aligned} & ((\text{CAR } E) (\text{INTEGRAL } (\text{CADR } E) X) \\ & \quad (\text{INTEGRAL } (\text{CADDR } E) X)) \end{aligned} \quad (5.14)$$

に分けて実行する。

$$\text{ここで } (\text{CAR } E) \text{ は } \text{PLUS} \quad (5.15)$$

$$(\text{CADR } E) \text{ は } (\text{TIMES } 3 (\text{EXPT } X 2)) \quad (5.16)$$

$$(\text{CADDR } E) \text{ は } 5 \quad (5.17)$$

である。

$(\text{CAR } E)$, $(\text{CADR } E)$, $(\text{CADDR } E)$ を置きかえると、

$$\begin{aligned} & (\text{PLUS } (\text{INTEGRAL } (\text{TIMES } 3 (\text{EXPT } X 2)) X) \\ & \quad (\text{INTEGRAL } 5 X)) \end{aligned} \quad (5.18)$$

となる。

$$\text{即ちこれは } \int \{f(x) + g(x)\} dx = \int f(x) dx + \int g(x) dx \quad (5.19)$$

$$\text{によって, } \int 3x^2 dx + \int 5 dx \quad (5.20)$$

と変換された状態である。

次に (5.18) の argument 1 すなわち、

$$(\text{INTEGRAL } (\text{TIMES } 3 (\text{EXPT } X 2)) X) \quad (5.21)$$

を実行する。(再帰呼出機能)

その結果は上記と同様に $(\text{CAR } E)$ を実行し (5.21) のリストのうち最初の (TIMES) を価とする。

次に、関数

$$\begin{aligned} & ((\text{EQ } (\text{CAR } E) (\text{QUOTE } \text{TIMES}))) \\ & \quad \underline{\text{TIMES}} \end{aligned} \quad (5.22)$$

(TIMES) となっている

によって、その条件式 EQ を True とするので、次のプロセデュアを実行する。

$$\begin{aligned} & (\text{SUBT } (\text{TIMES } 3 (\text{INTEGRAL } (\text{EXPT } X \ 2) \ X)) \\ & (\text{INTEGRAL } (\text{TIMES } (\text{DERIV } 3 \ X) \\ & (\text{INTEGRAL } (\text{EXPT } X \ 2) \ X)) \ X) \end{aligned} \quad (5.23)$$

この過程は,

$$\int f(x) \cdot g(x) dx = f(x) \cdot \int g(x) dx - \left\{ \frac{d}{dx} f(x) \cdot \int g(x) dx \right\} dx \quad (5.24)$$

に相当する。

その結果, 例は

$$\int 3x^2 dx \Rightarrow 3 \cdot \int x^2 dx - \frac{d}{dx} \cdot 3 \left\{ \int x^2 dx \right\} dx \quad (5.25)$$

と変換される。

さらに (5.23) において

$$(\text{INTEGRAL } (\text{EXPT } X \ 2) \ X) \quad (5.26)$$

は, 関数

$$\begin{aligned} & ((\text{EQ } (\text{CAR } E) (\text{QUOTE } \text{EXPT})) \\ & \quad \underline{\hspace{1cm}} \\ & (\text{EXPT } \text{となっ} \text{ている})) \end{aligned} \quad (5.27)$$

によりその EQ を満足する (即ち True) ので, さらに次のステップ

$$((\text{EQ } (\text{CADR } E) \ X)) \quad (5.28)$$

にすすみ, その EQ をも満足し, 従って

$$(\text{DIVIDE } (\text{EXPT } X \ 3) \ 3) \quad (5.29)$$

となる。これは

$$\int x^n dx = x^{n+1}/(n+1) \quad (5.30)$$

により,

$$\int x^2 dx \Rightarrow x^3/3 \quad (5.31)$$

故に (TIMES 3 (INTEGRAL (EXPT X 2) X)) は (5.32)

$$(\text{TIMES } 3 (\text{DIVIDE } (\text{EXPT } X \ 3) \ 3)) \quad (5.33)$$

となり, 簡単にすれば

$$(\text{EXPT } X \ 3) \quad (5.34)$$

となる。

また

$$(\text{DERIV } 3 \ X) \quad (5.35)$$

は E (=3) が number であるため, 0となる。

$$\text{すなわち } \frac{d}{dx} - C = 0, \quad \therefore \frac{d}{dx} 3 = 0 \quad (5.36)$$

(5.35) によって (5.23) のうち,

$$\begin{aligned} & (\text{INTEGRAL } ((\text{TIMES } (\text{DERIV } 3 \ X) \\ & \quad (\text{INTEGRAL } (\text{EXPT } X \ 2) \ X)) \ X)) \end{aligned} \quad (5.36)$$

は 0 となる。

以上により argument 1 を整理すれば

$$(\text{EXPT } X \ 3)$$

となる。

次に (5.18) の argument 2 に着目し,

$$(\text{INTEGRAL } 5 \ X)$$

を処理すると, $E (=5)$ は NUMBER であるため,

$$(\text{TIMES } 5 \ X)$$

と変換される。

この過程は,

$$\int c dx = cx$$

であることによる。

以上整理すれば

$$(\text{INTEGRAL } ((\text{PLUS } (\text{TIMES } 3 \ (\text{EXPT } X \ 2)) \ 5) \ X))$$

は,

$$(\text{PLUS } (\text{EXPT } X \ 3) \ (\text{TIMES } 5 \ X))$$

と出力される。

このようにして

$$\int (3x^2 + 5) dx = x^3 + 5x$$

という処理がなされる。

以下同様にすべての不定積分はこの LISP プログラムによって処理される

5.2.3 Run time の比較

Run time の比較は表 3 に示すとおりであるが, 微分積分プログラムのフルセットが NEAC-2230 ではメモリ不足となって処理できず, 約 1/3 の縮小モデルで Run をしていたのに対し, FACOM230-45S ではフルセットの Run が可能であった。単純に比較しても約 6 倍, 縮小化を考慮すると約 15 倍の処理速度が上がったものと考えられる。

§6. おわりに

HLISP は、現在いくつかの改良および拡張機能を附加する計画がすすんでいる。主な点は次のとおりである。

- (1) 無限整数の取扱を可能とする。
- (2) コンパイラ・オプション機能を追加する。
- (3) double word handling
- (4) GBC においてH領域をファイルを使って VS 化する。

(1) は本質的に LISP の機能を満足させるためのもので、現在の HLTSP が整数が、-9999 ~ +9999 であるという制約によって、階乗 (FACTORIAL) の計算が 7! までしか行えないといった制限を解除するものである。

(2)は、インタープリタ構造の短所であった Run time を高速化するのに有効である。

(3), (4)はメモリの効率的利用に有効である。

これらの改訂は、1976年秋以降に行う予定であり、それによってより強力なリスト処理の Tool となるものと思われる。

システムの性能評価としては HITAC-8800/8700 に手持ちのプログラムをかけ、特にここで紹介した2本の時間比較を行うことによって、F 230-45 S と H8800/8700 HLISP どちらの比較データが入所できる予定で、現在その準備をしている。

最後に東京大学情報科学研究室、後藤英一教授、金田康正氏、寺島元章氏、東京工業大学情報科学研究室、井上謙蔵教授、木村泉教授、佐々政考氏に資料の提供と貴重なアドバイスおよび援助していただいたことに深く感謝する。

また、プログラムディバグに助力していただいた経営工学科学生井手一美氏にも謝意を表す。

付 録

表1 HLISP ソース・プログラム

NO	ブロック名	行数	サブルーチン数	言語	註
1	S. HLISP	2288	22	FORTRAN	HLISP Source Program
2	S. EVALQT	2379	1	FORTRAN	" "
3	S. BLK	368	4	FORTRAN	BLOCK Source Program
4	S. VT	2015	28	FORTRAN	Virtual Tape Source Program
5	S. FCB	52	1	ASSEMBLER	Virtual Tape I/O
6	S. IFUNCS	49	1	ASSEMBLER	bit handling
7	S. INISET	325	1	LIST	Initial set cards (data)

計 7471 58

表2 HLISP Executable Binary

NO	E. B. Name	Memory Size	c. f.
1	HLISP	59page(11812)	HLISP
2	HLISP B	63 (126)	HLISP + BLOCK
3	HLISP BV	73 (146)	HLISP + BLOCK + VT

表3 比較データ (Run Time および Gavage Collection)

		FACOM 230-45S	NEAC-2230	
Machine Configulation	Main Memory Size	256 KB	2.4 kw (14.4 kB 相当)	
	Sub Memory Size	143 trk (Disk) (1029 KB)	10 kw (Drum) (60 KB)	
LISP	free storage	11000 cells		
	Program Size	59 page (118 KB)		
微分積分	Run Time	20 M	120 M (以上)	NFACで は縮小モ デル
	Gavage Collection	—	50 回 (以上)	
因数分解	Run Time	38 s 890 ms	30 M	
	Gavage Collection	— (但し memory time 1M 10S 500)	5 回	

文 献

- 1) McCarthy J.: *Recursive Functions of Expressions and their Computation by McCarthy*, C.ACM 3 pp. 184~195 (1960).
- 2) McCarthy J. et al: *LISP 1.5 Programmer's Manual*, MIT Press (1962).
- 3) Raphael B.: *SIR: Semantic Information Retrieval, in Semantic-Information Processing*, MIT Press (1968).
- 4) Bobrow D. G.: *Natural Language Input for a Computer Problem Solving System, in Semantic Information Processing*, MIT Press (1968).
- 5) Moses J.: *Symbolic Intergration*, Ph. D. Thesis MIT. (1967).
- 6) Moon D. A.: *MACLISP REFERENCE MANUAL*, Project MAC-MIT (Apr.1974).
- 7) Quam L. H., Diffie W.: *Stanford LISP 1.6 Manual*, SAILON 28.4 (1970).
- 8) Teitelman W., Bobrow D. G., Hartley A. K.: *BBN-LISP TENEX Reference Manual*, BBN (1972).
- 9) Teitelman W.,: *INTERLISP MANUAL* (Feb. 1974).
- 10) Knowlton, K. C.: 'A Programmer's Descriptoin of L6', *Comm. ACM*, 9 616-625 (1966).
- 11) Jean E. Sammet: *Programming Languages: History and Future C. ACM No. 7 VOL. 15.*
- 12) 後藤英一: *Monocopy AND Associative Algorithms in an Extended LISP, Technical Report 74-03, Information Science Laboratories, UNIV. of Tokyo.*
- 13) 金田康正: *Implementation of HLISP and Algebraic Manipulation Language REDUCE-2. Technical Report 75-01 Information Science Laboratories. UNIV. of Tokyo.*
- 14) 寺島元章: *Algorithms Used in an Implementation of HLISP Technical Report 75-03, Information science laboratories UNIV. of Tokyo.*
- 15) Clark Weissman: *LISP 1.5 PRIMER 1967. DICKENSON PUBLISHING COMPANY, INC.*
- 16) 後藤英一: 『LISP 入門』 bit NO. 1, Vol.6—NO. 2, Vol. 7
- 17) 佐々政孝: 「Block と Vartual Tape を持つ LISP システム」 S49 情報処理学会大会 NO.324