

教育用記号処理システム

大久保 明 伸*

Educational Symbolic Manipulation System

Akinobu OKUBO

Abstract

This report discusses the symbolic manipulation system for engineering education. This system is constructed by ULISP, muSIMP and SmallTalk on Msdos. ULSIP is the symbolic manipulation language which is implemented by Pascal language and has graphic functions. This system has many capabilities of fundamental symbolic manipulation and symbolic mathematical operation such as LISP and muMATH. This system provides the student with fundamental knowleges for symbolic manipulation and nonnumerical programing methodes.

1. はじめに

本格的な高度情報化社会の到来をまじかにひかえて、情報処理技術者の不足がさげばれている。大学や高専では、この社会的要請に答えるべく、増科や改組等で対応しようとしている。それでも不足が予想されるため専門学校や工業高校でも本格的な情報処理技術者の養成が行われようとしている。大学や高専には、情報処理技術者の養成に対して2つの要請がある。一つは、情報処理を専門とする人材の養成であり、もう一つは、情報処理そのものを専門としないが、情報処理技術の応用力を持った人材の養成である。計算機が広い分野に浸透しつつある現在、特に後者の要請が急務になっている。特に、工科系の学科では、かなり前から、この目的に沿ったカリキュラムが実施されてきた。汎用計算機やパソコンの価格の低下がこれを可能にしてきた。

しかし、情報処理関係の科目は、一部を除いて機械語、FORTRAN や BASIC の言語教育が中心である。限られた単位数のなかでは、情報処理教育をおこなうにはこれらの言語教育しかできないのも現実である。ところが、

*宇部工業高等専門学校電気工学科

計算機を利用する分野は急速に拡大していて、FORTRAN や BASIC の世界からは連想しにくい非数値的処理の世界に、その比重が移りつつある。CAD/CAM や AI などはその典型的な例である。このような状況であるから、情報処理教育はこれらの新しい分野に関する入門教育も並行しておこなわれるべきである。この観点にたつて、いわゆる記号処理を中心にした教育用ソフトウェアの構成について報告する。

2. 情報処理教育の問題点

高専における情報処理教育は歴史的にみて、4つの段階がある。約20年まえまでは、座学による講義のみであった。そのため講義の中心はハードウェアよりの講義で、ソフトウェアの教育はほとんど皆無といってもよい時代であった。特に情報処理を専門にしていた教官もほとんどいない状態であった。その後各高専に計算機室が開設されて、バッチ処理ながら学生のプログラム演習が実際におこなえる状態になった。この時代は手続き型言語の全盛期（特に FORTRAN と COBOL）であった。まだ、LISP や PASCAL を国産機で使用できる環境になかった。しかし、このころすでに、情報処理教育で使用すべき言語に関する議論が盛んにおこなわれていた。このために

各種の言語の提案があったが、技術系の仕事や研究で実際に使用していた言語は FORTRAN が中心でその他の言語はあまり普及しなかった。とくに情報処理の中心は、数値計算が主体であったので、記号処理や高度なデータ構造に関する教育は特殊な分野としてあまり注目を集めなかった。また FORTRAN コンパイラのみが安価に入手できる時代でもあった。

このような時代をへて、小規模ながら安価なパソコンの時代に突入した。このパソコンには BASIC インタプリタが ROM で提供されたために BASIC ユーザが多量に、そして自然に養成されるようになった。高専や大学でも、この安価なパソコンを使用して情報処理がおこなわれるようになり、FORTRAN や BASIC が中心的な言語になった。

パソコンも汎用計算機なみのオペレーティングシステムをもち、8ビットから16～32ビットのCPUをもつにいたった現在、種々の分野で利用されるようになった。特に、数値計算処理が占める分野は極めて少なくなった。

このように、約20年間の電子計算機の発達は急速であったにもかかわらず、情報処理教育における講義内容はあまり変化が見られない。初期の段階で導入された FORTRAN 教育が、現在も中心をしめている。しかし、計算機の小型、高性能化は、利用分野を拡大して、単に計算処理に使われることは少なくなった。この利用分野の拡大につれて種々の言語がすでに開発されている。このような状況をふまえて現在の情報処理教育の問題点をあげる。

(1) 情報処理入門教育

高専の低学年における入門教育では、ほとんどの学校で BASIC が採用されている。BASIC が採用された理由は、初期のパソコンに BASI インタプリタが搭載されていたのが、その大きな理由である。また、BASIC プログラムの製作が容易であり、グラフィクス命令もあるので初心者に向いている言語であると言われてきた。しかし実際に入門教育をして感じることは、取り上げる例題は程度の差はあれ、FORTRAN 教育で取り上げる数値計算的なものになってしまうのが現状である。入門教育の目的は電子計算機の動作の原理とソフトウェアの概念基礎を習得することである。基本になるプログラム法やデータ構造の習得には、BASIC や FORTRAN では十分ではない。とくにプログラムの製作上重要なサブルーチンの教育には困難を感じる。このような困難をさけるため、

情報処理を専門にする学科の入門言語として使用されている PASCAL を採用すべきである。これにより、数値計算的な例題ばかりでなく、色々な分野の話題を例題として自然にとりあげることが可能になる。

(2) FORTRAN 教育

BASIC で入門教育を受けた学生が、つづけてこの言語の教育をうけている。提示される問題は、やはり数値計算的なものになる。このために数値計算法の知識は広がるが、非数値処理に対するプログラミングに対する認識、データ構造や制御構造に対する認識はあまり進歩しないのが現状である。もし教師がデータ構造や制御構造をテーマにした演習を提示した場合、かなりのプログラムトリックを使用しなければならず、プログラム嫌いを増やす結果の一つになっている。逆にこれらの演習を全くやらなければ、情報処理は数値的な処理であると誤解させる結果になる。数値計算の知識も必要であるが、データ構造の概念や再帰処理など数値計算の世界では現れにくい概念や手続き等も習得させるべきである。これらの学習の過程で、情報処理に対する応用技術の基礎ができる。

以上の様に、情報処理教育の中心になっている二つの言語教育の問題点をあげてみた。これらの言語がほとんどの高専で教育されている理由には、前述した歴史的な環境による理由以外に、専門課程の教官にも問題がある。教官自身の研究で必要な情報処理は、ほとんど FORTRAN や BASIC で行われてきたので、卒業研究もこれらの言語さえ使えれば十分であると考えられている。このため新しい言語を使った情報処理教育にたいしてコンセンサスを得るのはむつかしいのが現状である。しかし、情報処理の応用分野が広がってきた現在、情報処理教育の在り方を考え直す必要がある。本来、情報処理教育の目的は数学や物理と同じような、計算機による情報処理の基本概念を習得させるのが目的である。基本概念を習得するために、実際に使用する計算機言語の選択は特に重要である。FORTRAN は FORTRAN の世界が LISP は LISP の世界が存在する。特に、高専の情報処理教育に不足している記号処理（非数値処理）にたいする基本概念の講義が必要である。CAD や AI の分野に直接たずさわる学生も現れてこようとしている。この観点にたつて著者は、パソコン上で動作する記号処理システム製作を行っている。このシステムの概要をつぎにのべる。

3. 教育用記号処理システムの概要

このシステムは、情報処理教育の入門用として開発した。記号処理の基本的な概念と簡単な応用の学習が可能である。このシステムの概要を図1に示す。このシステムを記述するのに使用した言語はPASCAL, muSIMP, Smalltalk とアセブラン言語である。実際、著者の卒業研究に配属された学生も、ほとんど記号処理の経験をもっていないので約1月程度の時間をつかい、基礎的な知識の習得に使用している。また、処理系自身も色々な言語で記述されているので、学生が製作しようする記号処理システムの参考になっている。

(1) ULISP 系

記号処理の入門として最初に講義がおこなわれるのはLISPであろう。LISPの基本関数であるCAR, CDR, CONS, EQ, ATOMPを使用してプログラムをつくる。この時、プログラムがどのように解釈実行されるのを理解するためこのULISPと全く同じ評価手順をこれ自身で記述してある。これにより初心者には無味乾燥なS式や

基本関数の意味が解理できる。また直接アクセス可能なグラフィック関数をもっているので、図形処理とS式を関連つけて理解もできる。このインタプリタはPASCALで記述されているので高学年ではこれを教材にして、インタプリタの改造演習も可能である。

(2) muSIMP 系

muSIMPは数式処理システムmuMATHの記述言語である。この系は、単にLISPの講義を聞いただけでは、記号処理のもつ重要性が理解できない学生に最適である。日頃手計算で行う数式計算がS式と基本関数で可能であることを強く印象づける効果がある。特にこの数式処理能力強化のために種々のプログラム群が付加されている。とくに数式処理と数値処理を混合して行うハイブリッド処理の機能もある。次節に示すような演習が提示でき、記号処理の重要性を理解させられる。またこの系にはmuSIMPで数式処理可能なPROLPGインタプリタが記述されている。PROLOGそのものの違った分野からのアプローチやPROLOGインタプリタの理解も可能である。詳しい内容は後述の文献を参照してほしい。さらにこの系に数式処理のための知的CAIシステムを構築中である。

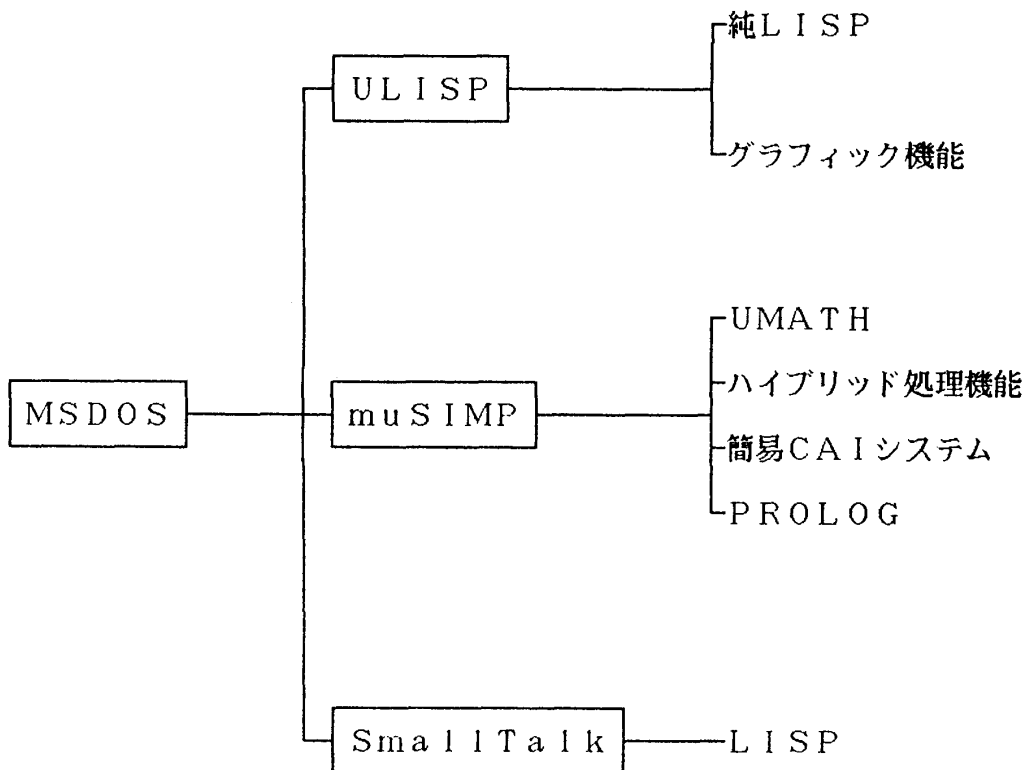


図1 教育用記号処理システム

(3) Smalltalk 系

この系の上に現在 ULISP インタプリタが動作している。この LISP のクラスリストの動作中の画面を図 2 にしておく。ULISP インタプリタを Smalltalk で記述した理由は、今後記号には特に必要なオブジェクト指向の概念の理解と、Smalltalk の優れた操作環境を利用するためである。またハイブリッド処理の観点からも重要であると考えられる。

```
Object subclass: #Lisp
  instanceVariableNames:
    'text atomDictionary sexper position anil t lambda quote typemiss'
  classVariableNames: ''
  poolDictionaries: '' !

!Lisp class methods !

new
  ^super new init:open! !

!Lisp methods !

eval
  | result sel |
  sexper:=text selectedString.position:=1.
  sel:=text selection.sel selectAfter:sel corner.
  result:=self readSexper.
  (result = false ) ifTrue:[result:=typemiss].
  self skipSpace:(sexper size < position) ifFalse:[result:=typemiss].
  result:=self eval:result env:anil.
  sexper:=''
  self printSexper:result.
  text replaceWithText:sexper. !

eval:aSexpr env:alist
  aSexpr atom
    ifTrue:[^self evalAtom:aSexpr env:alist].
  aSexpr car atom
    ifTrue:[^self apply:aSexpr car arg:aSexpr cdr env:alist].
  ^self evalis:aSexpr env:alist!
```

4. 情報処理教育への利用例

ここでは、このシステムをどのようにして、情報処理教育へ利用するかを簡単な例を用いて説明する。FORTRAN や BASIC の例題としてよく使われる「ニュートン法を用いて 2 の平方根をもとめよ。」とする。普通学生は、ニュートン法の説明をきき、プログラムで数値計算

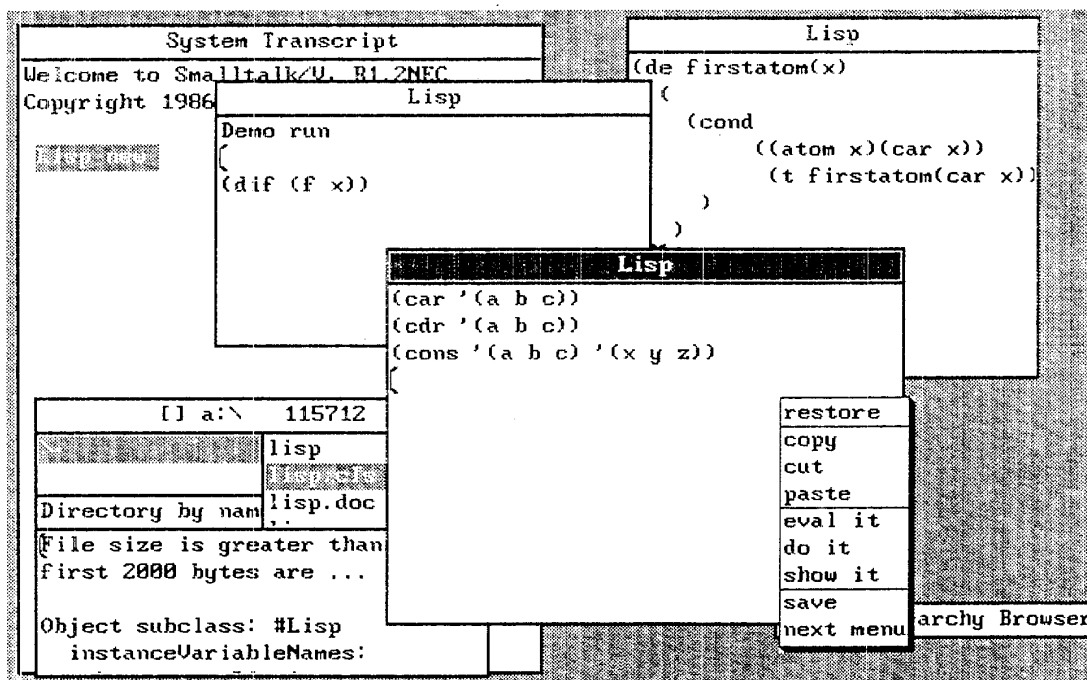


図2 Smalltalk 上での LISP の動作例

をするがこのシステムを利用した教育では次のような多種多様な方法が可能である。この例は数値計算から自然に LISP の世界に誘導する。

ステップ 1 ニュートン法による平方根のアルゴリズムを説明する。

ステップ 2 FORTRAN か BASIC で数値計算をする。

ステップ 3 UMAHT (数式処理システム) の微分機能を利用してステップ 1 で得た知識から種々の公式を発見し、実際にその公式を利用してみる。公式が複雑な場合はハイブリッド機能を用いて計算する。

このような過程で記号処理の威力を認識することができよう。

ステップ 4 数式のまま微分が可能なプログラムを実際につくる。ここで FORTRAN や BASIC ではあらわれてこない再帰プログラムを体験する。

ステップ 5 数式処理システムの内部で使用される S 式を自分が求めた公式と比較して理解する。

ステップ 6 S 式の基本操作である CAR, CDR, CONS 等の必要性を理解する。

ステップ 7 LISP のプログラムを学習する。

ステップ 4 で利用する言語は muSIMP であるが、S 式を直接使用しなくても記述できる。またプログラミングに必要な命令は 4 個である。またこのサンプルプログラムのリストを与えれば、簡単に理解できる。LISP から教育する場合、学生にとって具体的な例題を提示するにはかなりの時間が必要である。数値計算を講義するのに 2 進数の話からはじめるようなものである。この種のプログラムで必然的に現れる再帰処理も比較的較自然に導入できる。また、簡単な例を通じて行うハイブリッド処理

により色々な言語の特徴を理解させるこしがかのうである。

5. むすび

情報処理教育の改善がさげばれている。とくに専門学科以外の学科での情報処理教育の改善が必要である。本稿では、FORTRAN と BASIC が中心になっている高専の情報処理教育の改善の提案とこのために必要な記号処理システムの概要をしめした。著者の研究室に配属された学生はこれらのソフトウェアを使用して、ある程度まで記号処理の基礎ができるようになった。また、卒業生のなかには知的なソフトウェアの製作に従事しているものも多い。文部省の主催で毎年開催される情報処理関係の研究会でも AI やエキスパートシステムの講義への導入が議論されるようになった。パソコンの機能強化や EWS の普及につれて環境は整いつつある。

今後、このシステムの整備と講義法を検討していく予定である。

参考文献

- 1) muSILP/muMATH は The Soft Warehouse に所有権がある。
- 2) Smalltalk/v は digitalk inc. に所有権がある。
- 3) 大久保：パソコンによる数式処理システムの利用，電気四学会中国支部連合大会講演論文集，63年。
- 4) 大久保：パーソナルコンピュータ用数式処理システムの拡張(3)，電気四学会中国支部連合大会講演論文集，63年。

(平成元年 9 月 25 日受理)