コンピュータと教育　10－16
(1990. 4.26)

知的な数式処理システムＣＡＳＰと ユーザインタフェース

Computer Algebra System CASP and its user interface

対馬勝英，加賀英徳，中村勝則　（大阪電気通信大学）

Katsuhide TSUSHIMA, Hidenori KAGA , Katsunori NAKAMURA
Osaka Electro-Communication University

我々が既にＬＩＳＰを用いて開発した知的な数式処理システムＩＣＡＳをＰＲＯＬＯＧを用いて記述した。このシステムをＣＡＳＰと名付けたが，その特徴と構造を述べ，さらにこの上でＩＣＡＳに関する助言を行なうシステムを構築する方式を提案した。

数式処理，ユーザインタフェース，助言システム，ＰＲＯＬＯＧ

我々は従来の数式処理システムの機能を分析し，それが紙の上での人間の記号操作の際に持つのと同様のメンタルモデルを用いたのでは利用できないことを示した。[1),2)] この欠点を解消する数式処理システムＩＣＡＳは既に１９８５年に我々により作成されている。[3),4),5),6)]

数式処理システムの扱う数学的実体の持つ階層性を反映した情報構造を数式処理システムにもたせることで数式処理システムはオペレタ，汎関数，関数，微係数を整式を離れた抽象的な実体間の関係を取り扱う能力を獲得する。その機能により，数式処理システムを利用するユーザに紙上と同じメンタルモデルを用いた数式の処理が初めて，可能となる。[7),8)]

さて，数式処理は歴史的にはＡＩの一分野として出発したにも拘らず所謂，数式処理の高速アルゴリスムの様な特定の数式の処理に関する課題に関心が移り，人間の行なう数式に関する処理に対する関心は失われた。また，数式処理の研究分野としての独自性を強調するあまり，知的な因子を排除することが当然との狭いパラダイムを持つ研究者が増加した。わが国では，特にその傾向が強く，数式処理システムのヒューマンインタフェースについての研究は軽視され，数式処理の普及を疎外した。

また，数式処理と教育との関わりは，数式処理システムのユーザインタフェース及び知的インタフェースの改善に関わる重要なテーマであるが，多くの数式処理研究者はその意義すら認めない。これはわが国に特有の狭い専門主義の表れに他ならないが，この分野の活性化を妨げ，独創性のある研究の発展を疎外している。

数式処理システムを教育に展開する場合に，システムのユーザ支援機能の質が問題となる。特にシステムがＤＭＩとは言い難いヒューマンインタフェースをユーザに強要する場合にはまず，その改善が前提として欠かせないものとなる。我々が開発したＩＣＡＳ，ＩＮＴＣＡＳは

１）記号法に関するＤＭＩ
２）操作法に関するＤＭＩ

の双方の改善を行なった数式処理システムであり，教育において利用する準備の整ったものである。

最近，米国のメーカ主導による数式処理システムのヒューマンインタフェースの改善が進み，２）に関する改善はＤｅｒｉｖｅ，Ｍａｔｈｅｍａｔｉｃａ等により体験できよう。ＩＣＡＳの特徴は１）の実現を行った点にある。ＩＮＴＣＡＳは１）を前提として２）を実現した所にある。そして，両者が一体化されてＤＭＩとして機能することを実証した所に我々の一連の研究の意義がある。[9),10)]

この種のＤＭＩを持つ数式処理システムをそのまま教具として学習者に与えるだけでそれ以上の支援を行なわない環境で数理教育を行なう教育方式とより強い支援を行なうシステムを用いて教育を行なう方式がある。さらに，後者はアドバイザーによるそれと，特定のドメインに依存した教材

の知識を積極的に利用する方向に分けられる。

一貫して，数式処理システムを教具として用いる方向の研究に取り組んで来た我々は特定のドメインに依存しないアドバイザーを構築することを次の研究目標として設定した。

ＩＣＡＳは通常の数式処理システムと異なり，４種の数学的実体を持ち，それらに関する評価の抑止機能を具備している。これにより，関数，オペレタ，汎関数等に関する形式的取り扱いを可能としている。これが上述の記号法に関するＤＭＩを実現するが，操作に関して，利用者が新しいタイプのミスを犯す可能性が生じる。その意味で通常の数式処理システムとは異なるＩＣＡＳに固有のミスに対処する助言，支援機能が必要となる。

今回，我々がＩＣＡＳ－ＡＤＶＩＳＥＲの本格的構築を開始するに到った最大の理由がここにある。

整式の展開
微分
整式,関数の簡単化
(積分)
代数方程式の求解
ヒストリー機能

形式微分
数学的実体の評価抑止
高次パターンマッチング

1表 CASPの機能
---- の下はICASに固有の機能である.

２．ＣＡＳＰ

ＣＡＳＰ（Ｃｏｍｐｕｔｅｒ Ａｌｇｅｂｒａ Ｓｙｓｔｅｍ ｏｎ Ｐｒｏｌｏｇ）は１９８５年に我々により開発されたＰｒｏｌｏｇを用いて記述した数式処理システムである。それは，我々の開発したＩＣＡＳ機能を持つ点に最大の特徴を持ち，狭義の関数方程式の研究と数理教育用に於て利用することを目的として開発されたシステムである。ＩＣＡＳ機能とパターンマッチング機能の併用による問題解決記述能力の向上を強く意識してＣＡＳＰは開発された。

殆どの数式処理システムはＬＩＳＰを用いて記述されており，汎用のシステムを記述するにはＰｒｏｌｏｇよりもＬＩＳＰの方が適していることは衆知の事実である。さて，ＰＲＥＳＳに見られるように特定の分野の数式処理を行なうシステムをＰｒｏｌｏｇを用いて記述することは既に行なわれている。我々の作成したＣＡＳＰも解析に関する指向性を持つシステムであり，Ｐｒｏｌｏｇを用いて記述することにそれほどの困難はなかった。

［ＣＡＳＰの構造］

ＣＡＳＰは既に我々の開発したＩＣＡＳの標準機能である

１）数学的実体の評価の抑止機能
２）評価抑止機能付きの関数定義機能
３）記号微分，記号積分機能

等を，Ｐｒｏｌｏｇを用いて実現している。

その他の標準機能は１表に示した。

数式処理システムとして対話的に利用するためにＰｒｏｌｏｇで記述した数式処理機能の上に対話型インタフェースであるフロントエンド部ＵＳＲＩＴＰを構築し，処理結果を直接ＣＲＴに返す方式をとっている。この方式の利用をしているユーザはシステムがＰｒｏｌｏｇを用いて記述されていることを全く意識する必要はない。

一方，ユーザがＰｒｏｌｏｇを意識した問題解決を行なう際はフロントエンド部によらず直接，Ｐｒｏｌｏｇによる記述をすることができる。その際に

ｓｅｖａｌ（Ｅｘ，Ｒｅｔ）

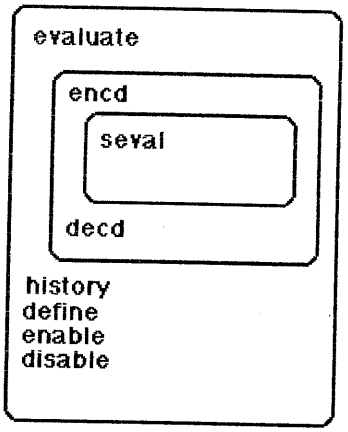
なる数式評価述語が中心的な役割を果たす。ｓｅｖａｌはＩＣＡＳに固有の評価の抑止を可能とするために１図に示した様にＣＡＳＰの構造のなかに埋め込まれているが数式処理機能を一つのＰｒｏｌｏｇ述語とすることで永田の提案した数式処理機能を持つＰｒｏｌｏｇを実現したことにもなっている。[1] ある種の問題解決においては数式処理システムを直接用いるより数式処理機能とＰｒｏｌｏｇの持つ推論機構の組合わせを利用する方が有効性を持つことはよく知られている。

数式処理アルゴリズムをＰｒｏｌｏｇで効率よく記述する問題はｓｅｖａｌの内部の問題であり，これとｓｅｖａｌを用いた問題の記述は別の問題であり，混同してはならない。

数式処理システムとしてＣＡＳＰを動作させるためには関数，オペレタ等の定義状況と抑止フラグを参照して整式としての評価，関数としての評価等の種々の数学的実体レベルでの評価を行なうＩＣＡＳに固有の評価機能を構築する必要がある。

これを実現したものがＵＳＥＲＩＴＦであり，この部分を数式処理部と独立に設計することが種々のヒューマンインタフェースを開発する際に仕様の変更にタフなシステム特性を与える。また，これはシステムの記述を容易にするという長所を伴う。また，この部分はヒストリ管理に関する述語を持つ。また，将来に於て，ＩＮＴＣＡＳ機能をＣＡＳＰに組み込む際にもこのフロントエンド部の構造を変更しなくてすむ。

以下にＣＡＳＰの各部分についてその概要を述べる。



1図 CASPの構造
CASPに於いては数式処理述語sevalをフロントエンド部がとり囲む構造をとっている.

［構文変換パッケージＰＡＲＳＥＲ］

数式はＣＡＳＰの内部ではリストとして格納され数式の処理はリスト処理として実行される。

ｅｎｃｄ（Ｅｘｏ，Ｅｘｉ）
ｄｅｃｄ（Ｅｘｉ，Ｅｘｏ）

は各々，入力Ｅｘｉのエンコード，出力Ｅｘｏへのデコードを行なう述語である。

［基本代数パッケージＡＲＩＴＨ］

実際の数式処理を行なう述語群よりなる。数式，有理数等の四則，簡単化，展開，同類項の処理，係数の取り出し，リストの要素の四則，等の多くの述語より構成されている。

［補助関数パッケージＥＸＦＵＮ］

ｍｕＳＩＭＰのｏｒｄｅｒにあたるもので数式が指定された順にならんでいるか否かを検出する。これは種々の数式処理のアルゴリスムを効率よく記述するために多用される。その他にリストに関する同様の述語や置換様述語が含まれる。

［リスト処理パッケージＬＩＳＰ］

リストの結合，分割，ａｓｓｏｃ，ｎｔｈ等のリスト処理用の述語を持つ。その他に

ｌｏｏｐ－ｗｈｉｌｅ（Ｐｒｏｇ，Ｃｏｎｄ）

なるＣｏｎｄの成立する限りＰｒｏｇの評価を続ける繰り返し述語を持つ。このパッケージにより通常のＬＩＳＰで記述した数式処理用アルゴリズムを継承することが可能となりシステムの機能の拡張が容易となった。

［解析関数パッケージＭＡＴＨＦＵＮ］

指数関数，対数関数，三角関数，逆三角関数等の処理，簡単化を行なう述語よりなる。また，これらの数値演算を行なう為の補助テーブルを持つ。

上記のパッケージ群は通常の数式処理を行なうためのものであるが，以下に述べるものはＩＣＡＳに固有の機能を実現するためのものである。

［数式の評価，定義ＤＥＦＵＮ］

ｓｅｖａｌ（Ｅｘ，Ｒｅｔ）

数式Ｅｘを評価した値がＲｅｔに単一化する。

ｅｖｓｔｒｃｔ（Ｏｐ，Ａｒｇｓ，Ｒｅｔ）

関数子（演算子）Ｏｐ，引数Ａｒｇｓの関数を評価した値がＲｅｔに単一化する。

ｅｖｓｗｏｎ（Ｆｎｃ）

関数子Ｆｎｃを持つ関数の評価を開始する。

ｅｖｓｗｏｆｆ（Ｆｎｃ）

関数子Ｆｎｃを持つ関数の評価を抑止する。

ＩＣＡＳに於ては関数，オペレタ等の評価の抑止は決定的な意義を持つ。上記の述語は内部に於てそれを実現する。通常の数式処理システムとしての利用においては後述するｄｉｓａｂｌｅ，ｅｎａｂｌｅの様なフロントエンドの持つコマンドによりＬＩＳＰで記述したＩＣＡＳと同様の利用方式をとる。

［記号微分パッケージＳＤＩＦ］

ｄｉｆ１（Ｅｘ，Ｖａｌ，Ｒｅｔ）
数式Ｅｘを変数Ｖａｌで微分したものがＲｅｔに単一化する。

ｄｉｆｐｔｒｎ（Ｆｎｃ，Ｒｅｔ）
関数Ｆｎｃをその引数で微分したときの結果がＲｅｔに単一化する。評価の抑止状況により，整式，または形式的微係数が返される。形式的微係数とはＦｘ，Ｆｘｘの様なものをいう。

これの反対の機能を持つ記号積分パッケージを作成しているが，積分機能が不十分であり，実用の段階にいたっていない。

［フロントエンドパッケージＵＳＥＲＩＴＦ］
このパッケージはユーザがＰｒｏｌｏｇを意識することなく通常の数式処理を行なう目的で設計されたものである。

ｅｖａｌｕａｔｅ　Ｅｘ
数式Ｅｘを評価するトップレベルのコマンド

ｄｅｆｉｎｅ　Ｄｅｆ　＝　Ｅｘ
関数Ｄｅｆを数式Ｅｘとして定義する。

ｃａｎｃｅｌ　Ｆｎｃ
関数子Ｆｎｃの関数の定義を解除する。

ｅｎａｂｌｅ　Ｆｎｃ
関数子Ｆｎｃの関数の評価を許す。

ｄｉｓａｂｌｅ　Ｆｎｃ
関数子Ｆｎｃの関数の評価を抑止する。

ｆｌｉｓｔｉｎｇ
ユーザ定義関数の定義状態を報らせる。

```
| ?-regist 2*X as double(X).
X       = X
yes
| ?-recognize a*2*b.
[recognition]   double(a*b)
[recognition]   double(b*a)
no
| ?-regist X+1 as inclement(X).
X       = X
yes
| ?-recognize a+1+b.
[recognition]   inclement(a+b)
[recognition]   inclement(b+a)
no
| ?-
|
|
|
|
|
|
|
|
|
    restor debug. trace. listin halt.

| ?-analyze 1+2*a.
[analysis]      1+2*a
[analysis]      1+a*2
[analysis]      1+double(a)
[analysis]      2*a+1
[analysis]      a*2+1
[analysis]      double(a)+1
[analysis]      inclement(2*a)
no
| ?-
|
```

2図 CASPに於けるパターンマッチングの例
この種のマッチングは多くの応用分野で必要とされるものである。

この他にヒストリーを管理，操作するコマンドを持つ。

［数式マッチングパッケージＰＭＡＴＣＨ］
整式の認知や公式の構造的な適用に関して
ｓｍａｔｃｈ（Ｅｘ，Ｏｂｊ）
がある。これはリスト形式で与えられたＥｘとＯｂｊの間のマッチングを試みる。

フロントエンドにおいては２図に示した様にｒｅｇｉｓｔ，ａｎａｌｙｚｅなるコマンドの形で利用される。これは，我々が既に開発した狭義の関数方程式の対話的解法を支援するＦＵＮＥＱに於て関数方程式のパターンを分類する際に利用されている。
［フロントエンドでの操作例］

・評価
ｅｖａｌｕａｔｅ　ｘ＋１＋ｘ．
－－－＞　１＋２＊ｘ

```
evaluate(x+1+x, Ret).
Ret=1+2*x.
```

evaluateはフロントエンドのコマンドとしてもPrologの述語としても定義されているので二重の使用が可能である。

・関数定義と評価の抑止

```
define f(X) = X 2+X+1.
define g(X) =log(X).
disable g.
evaluate f(g(x)).
--->1+g(x)+g(x) 2
enable g.
evaluate f(g(x)).
--->1+log(x)+log(x) 2
```

上記の様にCASPに於てもICASと同様に関数の定義と評価の抑止のコントロールが可能である。

・再帰定義

```
define fac(0) by 1.
define fac(X) by
          X*fac(X-1).

evaluate fac(5)
  ---> 120.
```

フロントエンドのレベルでの再帰定義が許されている。

・オペレタの取り扱い

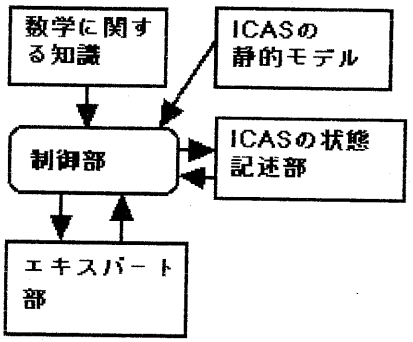
オペレタの前置型記述と評価の抑止が可能である。

```
evaluate d sin(x).
 --->cos(x)
```

の様にオペレタの作用を記述する。

```
evaluate (d 2+1) sin(x)
 --->0
```

ただし，dはオペレタとて事前に定義されている。



3図 ICAS-ADVISERの構造

3．ICAS－ADVISER

2や参考文献に示した様にICASは記号法に関するDMIを実現した使いやすいシステムであるが，実体の定義状態，レベルを意識して使用するので操作に関してICASに固有のミスを生じる場合がある。また，CASPにおいてはユーザは数式処理コマンドとsevalを持つPrologの述語の利用という二重の操作を行なえ，それによる操作ミスの出現も考えられる。この様な数式処理システム利用におけるユーザのミスを診断して助言を返そうとする試みはMACSYMA－ADVISERが著名である。[12] また，数式処理の誤答診断に関しては我々の研究がある。[13]

さて，ICASはシステムの扱う数学的な実体が多岐にわたる数式処理システムであり，それに関するアドバイザーを開発することは興味深いテーマである。

我々は2で詳述したCASPシステムの上にICASに関する助言システムICAS－ADVISERを構築することを意図している。現時点では3図に示した構造をとることとした。

数式処理システムのユーザの操作に関する助言を行なうためにはシステムは自らの行なう処理に関するモデルをもたねばならない。そこではICASのコマンドの実行に関する仕様が静的な形で記述されている。この部分はICASにおいては

多岐にわたるがルールとしての整合性に富み，記述は容易である。モデル部と数学的な知識を分離して記述することはシステムの記述の容易さの点よりも，ユーザの誤りに構造的に対応する点よりも当然である。数学的な知識はＩＣＡＳにおいてはシステム記述者にとりヒューマンフレンドリーな形で記述できる。特に，汎関数，オペレタに関する記述性の良さはＩＣＡＳの大きな特徴である。

制御部がユーザの操作列に関する診断を行なう際には変数の束縛状態，関数の定義状態，抑止フラグの状態等のシステムの状態を参照する必要がある。リアルタイムの助言を生成するためにはこの部分を独立させることは一つの方法といえる。

エキスパート部は誤りを含むユーザのコマンド列より正しいと思われる操作列を生成する機能を持つ必要がある。　二つの列を比較して助言を生成するが，その際にユーザに意図を聴くための問答を行なうと助言生成のための推論機構への負荷を軽減できるであろう。また，学習者に関するモデルを生成する方式をとることも考えられる。最初は，この部分の構造を比較的軽いものにし，助言の有効性が獲得できねば，改良する方式をとることとした。

参考文献

1）対馬：知的な数式処理を目指して１，２，数式処理通信，１－４，２－１，（１９８４）

2）対馬，加賀，中村：数式処理におけるＤＭＩ機能の実現　－－－ＩＮＴＣＡＳの開発－－－，ＣＡＩ学会誌（印刷中）

3）対馬：小型数式処理システムとその応用，情報処理，２７，３７９，（１９８６）

4）K.Tsushima et. al.:Teaching Mathematics Using a Computer Algebra System,Proc. of MCSE'86 ,407,(1986).

5）対馬：数式処理を用いた微分の教育，ＣＡＩ学会誌，５，５９，（１９８７）

6）対馬：ＣＡＩハンドブック（数式処理），フジテクノシステム，（１９８９）

7）対馬：数式処理におけるヒューマンインタフェースの改善，ヒューマンインタフェースシンポジウム論文集，３４７，（１９８７）

8）対馬：数式処理と教育工学，大阪大学大型計算機センターニュース（教育工学特集号），（１９８９）

9）対馬，広田：数式処理ワークステーションの開発と数学教育への応用，信学報ＥＴ８７－５，（１９８７）

10）K.Tsushima,H.Kaga and K.Hirota:Mathematical Education using a Workstation for Computer Algebra,Proc. of Asian-Pacific Conference of Education, , (1988)

11）永田：数式処理におけるＰｒｏｌｏｇ，情報処理，２５，１４０４，（１９８４）

12）Genesereth,M.R.:The Role of Plans in Automated Consultation,Proc. of 6th IJCAI,311,(1979).

13）対馬，阪野：数式処理テストに関する誤答診断システムＳＭＤＩＡＧの開発，ＣＡＩ学会誌，３，４，５，（１９８４）