数式処理システムの開発（第１報）

3X-4

村端晋一
（トヨタ自動車㈱ 東富士研究所）

１．はじめに

近年、MACSYMA やREDUCEなど世界的に有名な数式処理システムが国内に流布し始め産業界の各方面から注目を集めている。本論文では今回開発した数式処理システムについて述べる。本システムは現在のところ多倍長の整数、有理数、多項式および分数式までを演算の対象とし、その処理には四則演算による式の代入、置換、微分、積分、ＧＣＤ演算などがある。開発言語はUTILISP である。

２．機能及び仕様

システムの入力には直接の処理の対象である数式の演算式とシステムを操作するシステムコマンドがある。図１にシステムの構文規則、図２に演算の種類を示す。システムコマンドについては省略する。

(1) 演算式はその処理の単位を括弧で括る。
(2) 書式は自由形式であり、行概念はない。数の桁、文字数に制限はない。変数名は英字で始まる英数字で、予約語は現在のところない。
(3) 演算式の表現は入出力ともにFORTRAN などの数式表現と同じである。このため出力を再び入力として取り込むことも可能である。
(4) コマンドファイル実行機能を備えている。ネスティングは自分自身以外ならば自由にできる。

３．システム構成

本システムは一種の言語処理系であり、インタプリタ本体と数式処理を行う演算ライブラリから構成されている。図３にシステムの基本構成を示す。インタプリタのメイン部は入力部、解釈実行部、出力部の三つから成り、これら全体をトップレベルルーチンが管理している。トップレベルルーチンは全体のサイクルを回す大きい逐次ループである。またコマンドファイル実行には再帰ループでファイルのネスティングに対応している。入力部は字句解析、属性変換、構文解析の三つからなる。解釈実行部はツリーのトレーサである。出力部は２段階の表現変換と印字処理の三つからなる。変数表はシステム内部のデータベースでLISPの属性リストを利用している。変数名を検索キーにし解釈実行未処理の演算式ツリー（式）とその解釈実行結果（値）の三つのデータで構成する。演算ライブラリは整数、有理数、多項式、有理式の各種演算ルーチンの集まりである。

４．処理の流れ

次ページ図４に簡単な例で処理の流れを示す。

入力＞　　：入力待ちの状態
入力＞（演算式）．．．：括弧で括る。
入力＞　コマンド．．．

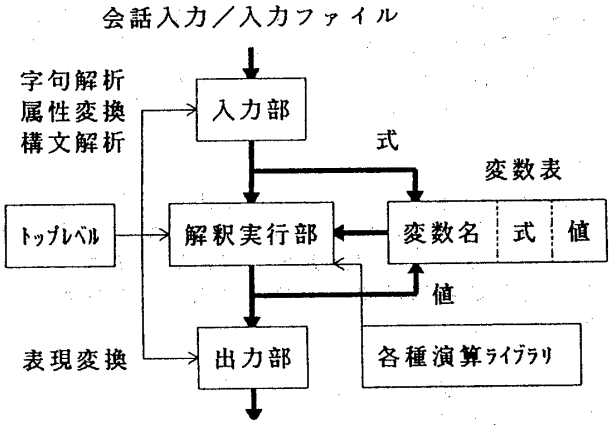
図１．構文規則

(1) 代入　　変数　＝　演算式
(2) 演算式　　演算式１　OP　演算式２

OP ⇨					
＋	和	**	巾乗	¥	剰余
－	差	／	商	¥¥	ＧＣＤ
＊	積	//	整商	$$	ＬＣＭ

(3) 微分　　ＤＩＦ変数名　演算式
(4) 積分　　ＩＮＴ変数名　演算式
(5) 引用　　ＥＶＡＬ　演算式
(6) 引用中止　ＱＵＯＴＥ（変数名）

図２．演算の種類



変数名　：検索キー　　←：データの流れ
式　　：演算式ツリー　　←：制御の流れ
値　　：解釈実行結果

図３．システム構成

COMPUTER ALGEBRA SYSTEM
Shinichi Murabata
TOYOTA MOTOR CORPORATION

(1) 入力部

会話的な操作で端末から、あるいはコマンドファイルからシステムに取り込まれた入力文字列は、字句解析により変数名、数、演算子など基本要素に切り出される。各要素は属性変換を経て、構文解析により演算式ツリーにまとめられる。この過程で数は多倍長整数、多倍長有理数に変換され、変数名はシステムアトムとして登録される。構文解析は演算子順位文法による通常のプッシュダウン論理により処理している。演算式ツリーは数式の演算順位を表現したツリーで下位ほど順位が高い。ノードは演算子を表しブランチはその被演算子を表す。終端要素は変数と定数である。

(2) 解釈実行部

入力部で作成された演算式ツリーのトレースを行い、システムコマンドの実行あるいは各種数式処理演算を行う。往きのトレースではツリー中の変数の処理を行う。変数に遭遇したならば変数名をキーにして変数表の検索を行う。存在したならば変数を式（演算式ツリー）で置き換えて更にトレースを続ける。あるいは変数に値を代入して戻る。値の場合はその先のトレースをしない。式か値かの制御は関数EVALで行う。この時関数QUOTE を使うとこれらの変数表参照は行われなくなる。変数表に存在しない場合には、変数を係数が１、次数が１の単項の１変数多項式とみなして変換する。戻りのトレースではノードの演算子に対応する数式処理ルーチンを呼び出し下位のブランチへ適用する。下位ブランチはこの時既に数式処理可能なデータになっている。このように演算式ツリーは下位ほど順位が高いため下からの戻りの際に演算を行う。なお入力部から出力された演算式ツリーがアトムの場合には、システムコマンドかどうかチェックを行っている。

(3) 出力部

解釈実行部により得られた結果を内部の代数的な表現からFORTRAN 形式の外部表現に変換する。一度LISP形式の演算子前置表現を経由して変換している数の外部表現は整数、小数、分数の三通りあるが分母、分子をみて適当に選択している。

5．数式処理について

(1) データ形式

UTILISP は多倍長整数をサポートしていないため次のように千進法で整数、有理数を表現している。

整数　（識別子　符号　桁１　．．．　桁Ｎ）

有理数　（識別子　符号　分子整数　分母整数）

符号は正の場合１で、負の場合－１である。多項式のデータ構造は多項式環の再帰性を利用しており、MACSYMA 等が取っているデータ形式と同じである。

（変数名　次数１　係数１．．．次数Ｎ　係数Ｎ）

次数　：　整数

係数　：　有理数あるいは下位多項式（多変数）

次数は降順で係数が０の項は表現しない。多変数の場合多項式のネスティング順序は変数名で昇順である。有理式は次のように多項式のペアで表す。

（識別子　分子多項式　分母多項式）

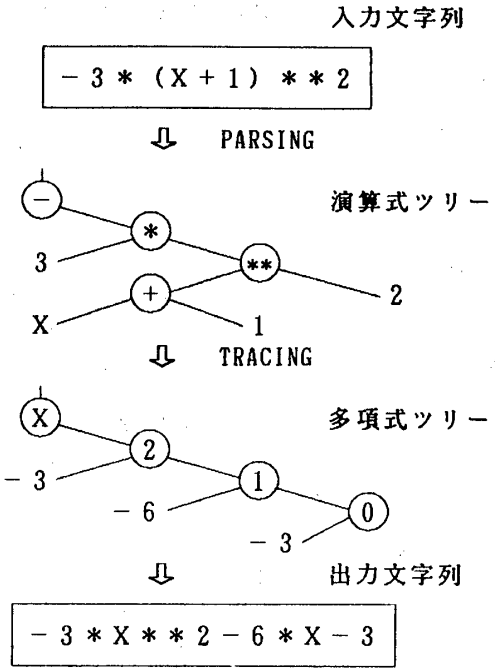


図４．処理の流れ

入力＞（　(X**10-Y**10) ¥¥ (X**15-Y**15)　）
X**5-Y**5

図５．使用例

(2) 数式演算

多項式の和・差は変数名と次数の二つをキーにした２本の多項式ツリーのマージである。キーが一致したならば係数間で和・差をとる。係数が下位多項式ならば再帰処理をする。キーが不一致ならばダミー項を挿入して再帰処理をする。多項式の積も同じような方法である。ただしこの場合マージのキーは変数名だけである。多項式のＧＣＤは係数ＧＣＤと原始多項式ＧＣＤに分かれる。係数ＧＣＤは再帰で処理し最終的に整数のＧＣＤに帰着する。原始多項式ＧＣＤは擬似除算を用いた拡張EUCLIDアルゴリズムで処理する。有理式の演算は多項式演算に帰着する。図５にＧＣＤ演算の例を示す。

6．おわりに

自動車の研究開発においても制御、流れ解析などの分野で大きな数式を扱う機会が増えており数式処理に対する期待が高まってきている。本システムはユーザオリエンティドで使いやすいが、機能が少ないため未だ本格的な利用には至っていない。今後も実用化に向けて機能の拡張を続けていく予定である。

参考文献

Winston & Horn「ＬＩＳＰ」
一松・広瀬「計算機概論ー数式処理(12)~(15)」数学セミナー
佐々木健昭「数式処理」情報処理学会