

# 会話処理用教式処理言語 AL-C

神原慎一, 池田義則, 角田俊晴  
(電々公社 横須賀通研)

## 1. まえがき

計算機による計算といえば、もっぱら数値計算を意味する程であるが、記号変数を含む式の計算という点、式の変形や微積分計算の様な計算があり、教員全体から見ればこちらの方がより重要である。例えば、 $\int \frac{dx}{1+x^2}$  を求めるとき、人間は、まず被積分関数の不定積分  $\tan^{-1} x$  を計算し、それによって上下限値を入力して  $\pi/2$  という答を求める。計算機による教式処理では、上例の様々、人間が記号的に行なう計算を計算機が行なうのである。もちろん、誤差関数  $e^{-x^2}$  の不定積分や5次以上の代数方程式の根の様々に解析的に解を求める様なものについては、数値計算が有用である。しかし、教式処理では、記号変数を含んだままでも計算でき、もともと非数値解を求めることも目的とした計算を行なうことができる。

計算機による教式処理への要望は早かったが、その手がかりが得られぬ間に数値計算が著しく進歩し、解析的な計算の一部を数値計算で済ませてしまおうようになった。しかし、1960年以降、米国を中心として計算機による教式処理が実際に試みられるようになった。たとえば、IBMのFORMAC<sup>1)</sup>、MITのMACSYMA<sup>2)</sup>、ユタ大のREDUCE-2<sup>3)</sup>等がよく知られていゝ。また、国内でも、東大に於けるHLISPによるREDUCE-2の導入<sup>4)</sup>、東北大でのFORMACの開発等、教式処理に対する研究、開発が活発化してきた。

さて、解析的な教式計算は、極論すれば、全て公式を順に並べるだけの事であり、答とは、与えられた問題と同様に、その場で都合のよい形をした一公式にすぎない。この様な計算をプログラム化する事は、実は非常に困難な事である。すなわち、公式はもともと可逆的であって、一つの公式は二通りに適用する事ができる上、同じ式に対して、いくつもの公式が適用できて、その結果が異なる様な場合もある。これに対して、人間は、試行的な思考を得意とし、パターンマッチング能力に優れているので、上述の計算機の欠点を補える立場にある。難しい計算であっても、中間結果の式を眺める事により、その次に適用すべき適切な公式(変形規則)を見つけた事が可能である。この様な意味から、計算機は、指示された公式を適用して式を変形する事に努め、それを操作する人間が、公式の選択という判断を行なう、両者の協調形態が考えられる。これを実現する事が会話型の教式処理システムである。

今回筆者等が開発した教式処理用の言語AL (Algebraic Language) は、公社のTSSサービスであるDEMOS-Eのライブラリとして、一括処理用パッケージ(AL-B)に加えて会話処理パッケージ(AL-C)を提供する事により、インタラクティブな形態で教式処理が行なえるばかりでなく、言語仕様、制御方式、入出力形式においても利用者の立場に立った教員の工夫がなされていゝ。

## 2. ALの言語

ALシステムと利用者のインタフェースをツカサどる手段がAL言語である。AL言語は、教式処理に直接関係はもたないが、ALシステムの初期化、終了、手順書の実行指示等を使用されるALサブコマンドと、教式処理を行なうプロ

ラムを作成するための言語であるALプログラミング言語がある。

## 2.1 ALサブコマンド

ALではサブコマンドとして6種類のコマンドを持っている。これは、DIPS-100の「基本コマンド」と区別するため、特にサブコマンドと呼んでいる。サブコマンドには会話処理時のみ使用できるもの、一括処理時のみ使用できるもの、および、両処理形態に共通に使用できるものがある。表-1にALサブコマンドの機能と使用区分を示す。

ALのサブコマンドは、冗長な表現を避け、すべて4文字以内の短縮形を用い、入力が簡単なようになっている。また、パラメータも必ず最小限にとどめ、/RUNのプログラム名、/PR0のファイル名を除くパラメータは、デフォルト値の扱いが可能である。また、ALは、システムとして簡単に設計されているので、操作法に対するHELP機能は特に設けていない。

## 2.2 ALプログラミング言語

ALでは、一括処理用の言語をALプログラミング言語と呼んでいるが、会話処理時にも一部の文を除いて、そのまま、インタラクティブな言語として使用することができ、ALプログラミング言語は、会話形式で数式処理を行なうことを考慮して種々の工夫がなされている。

### 1) ALステートメント

ALステートメントは、代入文、計算文、プログラムの構造文、プログラムの制御文、メモリ制御文、宣言文、定義文、入出力文の8種類に分類できる。表-2にALステートメントを一覧表にして示す。表中( )印のものには、一括処理時のみ使用できるステートメントがある。ALステートメントの形式は、PL/Iに似ているので、数値計算用の言語を使いながら利用者がスムーズに数式処理に慣れることが可能である。

### 2) 組込関数

ALの組込関数をその用法に着目して分類すると、3種類の関数に分類することができる。数式処理関数は、数式処理に必要なアルゴリズムを関数の形式にまとめたものである。数学関数は、数式表現あるいは算術表現で使用されるものであり、数式表現中で使用されると名詞的な意味を持ち、関数自身が値となる。一方、算術表現で使用されるものは、動詞的な要素として、その値が計算される。算術演算関数は算術表現でのみ使用される数値計算専用関数である。表-3にALの組込関数を一覧表にして示す。組込関数のうち、数式処理関数は、他の関数と比較して引数の数が多く入力が複雑である。このため、ALの数式処理関数の引数には、いくつかの省略を認め、省略時解釈としてデフォルト値が定められている。これにより、利用者は、ALの数式処理関数を、より簡単に使用することができる。表-4にALの数式処理関数のうち、引数の省略が可能な関数と、その引数の意味を示す。

### 3) システム変数

例えば、数式を展開するときなど、展開するべき式の最高次数、あるいは、最低次数を関数の引数として与えることはプログラミング上、複雑である。ALでは、このような冗長さを避けるため、システム変数という概念を設け、算術代入文の実行時に一括して値を与えられるようにしてある。ALにはシステム変数として、次の3変数がある。

- MAXPOW — MEXPAND, DIST, EXPAND等, 展開を行う関数 (MIN) 数において, 展開すべき中の上階(下階)を与える。デフォルト値は +6 (-6) である。
- SMPLE — 数式代入文実行時, 適用される自動簡単化のレベルを与える。レベルは, 0, 1, 2 の指定が可能であり, デフォルト値は 1 である。

#### 4.) 属性の宣言

インクティブに数式処理を行なうとき, 使用する変数の属性を, 前もって宣言しなければならぬとすれば, それは困難なことが多い。このため, ALでは, 文脈宣言, あらゆる暗黙宣言を可能にしてゐる。配列を除くと, 特別変数の使用法をとりなす限り, 変数の属性を宣言しなすむかゝらなっている。

- 代入演算子 — 代入演算子として, 数式代入文では ::=, 算術代入文では = を使用することにより, 両辺の変数の属性を文脈的に決定してゐる。
- 関数呼び出し — 組込関数の引数として使用された変数は, その関数呼び出しとして許される属性に従って, その属性が確立される。
- 整数属性 — 算術変数の 1 文字目が I, J, K, L, M, N で始まるとき, 整数属性が暗黙的に確立される。

サブコマンド名	パラメータ	使用区分		機能
		会話 AL-C	一括 AL-B	
/INI	なし	○	×	処理の途中ですべてのデータを破棄し, 初期状態にする。
/CAN	〃	○	×	DEBグループの入力途中でDEBグループを破棄する。
/RUN	プログラム名 (エリアの文)	○	○	中間言語ファイルになった利用者手続を実行する。
/END	なし	○	○	ALシステムを終了させる。
/PRO	ファイル名 リスト有無	×	○	一括型コンパイラを起動して, ソースを中間言語に翻訳する。
/;	なし	×	○	/PROで入力されるソースプログラムの終了を示す。

表1 ALのサブコマンド

文の区分	文の名称
代入文	数式代入文, 算術代入文
計算文	SOLVE文
プログラム構造文	(PROC文), (END文)
プログラム制御文	GOTO文, IF文, DO文, CALL文, (RETURN文), (STOP文), END文
メモリ制御文	SAVE文, ATOM文
宣言文	DECL文
定義文	FUNC文, DERIV文
入出力文	(GET文), (PUT文), PRINT文, HPRINT文, 出力要求文

表2 ALの文 ( )内は, AL-Cで使用できない文

分類		関数名	分類	関数名
数式処理関数	展開	MEXPAND, DIST, EXPAND	数学関数	SIN, COS, TAN SINH, COSH, TANH ASIN, ACOS, ATAN ASINH, ACOSH, ATANH EXP, LOG SQRT FACT, COMB
	分数	NUM, DENOM, CODEM, FRAC		
	簡約化	SIMP		
	再配置	ARNG		
	恒等性	IDENT		
	変換	FORMLT, NUMRT		
	数値化	EVAL	算術演算関数	ABS, CEIL, FLOOR GCM, LCM, MOD LOG10
	置換	REPLACE		
	対数	LSUM, LEXPAND		
	多種表現関係	RIGHT, LEFT, EXCH		
	微分	DIFF		
	積分	INTGRT		
和	SUM	抽出 数式解析	COEFF, CNSTT, <sup>HIGHPOW</sup> <sup>LOWPOW</sup>	
抽出	COEFF, CNSTT, <sup>HIGHPOW</sup> <sup>LOWPOW</sup>			
数式解析	LOP, ARGNO, PART, SUBPART			

表3 ALの関数

関数名	省略可能な引数	省略時の意味
SIMP	簡約化のレベル	レベル=2 (AL最高レベルの簡約化)
ARNG	再配置指示要素	ALで定める標準に従い、並べ替える。
EVAL	変数の組	数値計算できる部分を計算する。
REPLACE	代入法(並列/逐次)	P(並列)代入を行なう。
DIFF	微分回数	微分回数は1回とする。

表4 引数を省略できるALの関数

### 3. 会話型言語処理

ALの言語処理プロセッサは、インタプリタ方式であり、処理単位ごとにコンパイルと実行が繰り返される。AL-Cでの処理単位として、単一ステートメントとFORグループがある。単一ステートメントは、一文ごとにコンパイルと実行が行なわれるが、FORグループでは、グループと単位としてコンパイルと実行が行なわれる。

### 3.1 システムの状態遷移

AL-Cの言語処理プロセスの状態遷移について述べる。

1) ALステートメント待ち状態 — 会話処理プログラムとしてシステム起動後、この状態となる。

2) 文入力待ち状態 — 1)の状態での文を投入した時、この状態となる。

3) 言語処理状態 — 1)の状態での文以外でALステートメントを投入したとき、この状態となり、入力文の構文チェックがなされる。誤りを検出したとき、1)の状態となる。2)の状態のとき、END文以外の場合、入力文の構文チェックの後、2)の状態に戻る。

4) 実行状態 — 3)の状態での構文解析が正常に終了したとき、あるいは、2)の状態でのEND文が投入されたとき、3)の状態を経由してこの状態となる。

図-1に会話処理時の状態遷移図を示す。

### 3.2 システムの構成

AL-Cシステムの構成を機能上、大きく分割すると、主制御部、コンパイラ部、実行管理部、ライブラリ部になる。各処理部の機能は次のようになる。

1) 主制御部 — システム全体の制御、初期設定、終了関係メモリ制御を行なう。また、サブコマンド処理を行なう。

2) コンパイラ部 — ソースステートメントの解釈と中間言語の作成を行なう。

3) 実行管理部 — 中間言語の解釈実行を行なう。

4) ライブラリ部 — 教式処理関数、数学関数、算術演算関数等の処理ルーチンより構成されている。

AL-Cの言語処理は、端末より入力された一文単位ごとに主制御部のコントロールのもとに、コンパイラ部に渡され、ここで中間言語に変換される。次に、実行管理部に制御が渡り、中間言語を解釈実行する。ワイルドグループの場合、コンパイラ部は、END文が入力されるまで中間言語を順次作成し、END文入力後、エンバール等を解決して一括して実行管理部に渡す。ライブラリ部は、実行時、必要の都度呼び出される。図-2に、AL-Cシステムの構成を示す。

### 3.3 エラー処理

システムで生じる誤りには軽微なものも、強度のものがある。前者の場合、システムにより適切な解釈がなされ、処理は正常に終了するが、警告メッセージ(ワイルドエラー)が出力される。後者の場合、エラーメッセージ(フレベルエラー)が出力され、原則として、その文が入力される以前の状態に戻る。ALステートメントの誤り処理条件は、コンパイル時と実行時で分けられる。ALステートメントのコンパイル単位は、一文、あるいは、グループ単位であるが、エラー処理の単位も同様である。入力ステートメントに誤りがある時、コンパイラは、エラーメッセージを送出した後、それを破棄し、ステートメントの再入力を要求する。ワイルドグループの場合、入力ステートメントの構文チェックは、一文単位に行なわれ、誤りのとき、その一文のみを破棄する。また、エンバールの未解決等、グループとしての意味的エラーは、END文入力後、チェックされ、誤りがあればグループ全体が破棄される。また、/ENDがサブコマンドにより、作成中のワイルドグループを破棄するこゝも可能である。一方、実行時の誤りは、回復可能なエラーは、文の投入前の状態に戻るが、回復不能の誤りについてはALシステムの初期化が行なわれる。

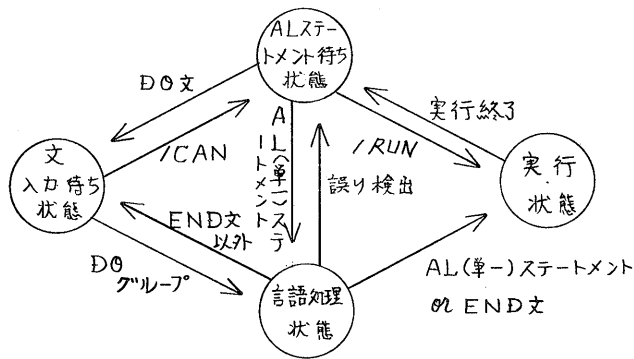


図1  
会話実行時の  
状態遷移図

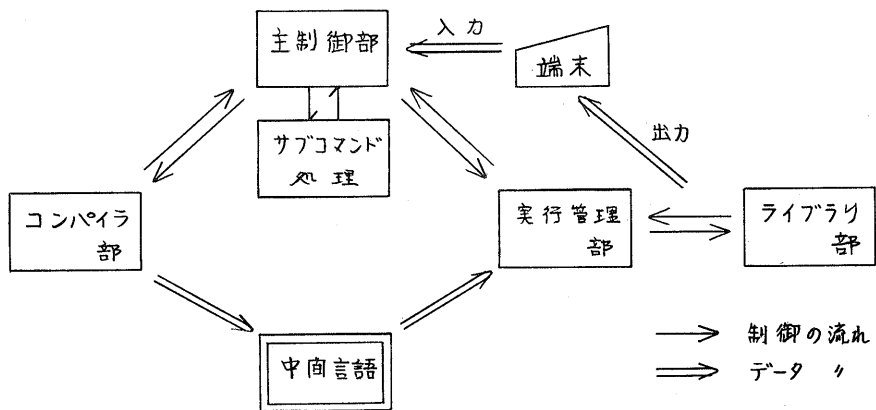


図2 AL-Cシステムの構成

#### 4. 入出力機能

この様子種類のシステムであっても、システムへデータを投入する事と、システムからデータを出力させる事は重要な事柄であるが、教式という種類のデータも会話的にサリヒリするシステムにとっては、特に重要である。この場合は、教式処理システムは教通のみでなく、教式もデータとして投入するにせ、出力される式は相当複雑な場合が多いこと、出力される式の意味を人間が見て迅速な判断を行なわねばならぬこと等、システムと人間のインタラクシヨンが高度で、おぼろげ、頻繁に行なわれるためである。しかも悪いことには、教学上の記法は、タイプライタやライノプリクタ等の行スキヤン型出力機器向きには決められておらず、逆に、出力機器は、教学記法上の特殊文字や、ギリミナ文字等をほとんど持っていない。この様な条件下でALでは、端末タイプライタを通して、次の機能を入出力の手段に提供している。

##### 4.1 GET機能

変教を指定して、実行時にデータとして教式を読み込み、その変教に代入する。データとしての教式の形式(構文則)は、プログラマステートメントとしての教

式，および PRINT 機能による出力式のそれとほとんど同じである。GET 文自身は，一括処理においてプログラムを作成するときのみ使用できる文であるが，一括形式で作成されたプログラム（中間言語）を会話処理で実行する時，AL-C でも GET 機能を使用できる。

#### 4.2 PUT 機能

改行，改頁を指定して一行を単位とする文字列定数を出力する。AL では文字列の演算を導入して  $\backslash$  の  $\backslash$  で，タイトル等，固定の文字列のみを扱う。なお，PUT 文の使用制限は，GET 文と同じである。

#### 4.3 PRINT 機能

変数を指定して，その時点で，該当変数に代入されている値を出力する。出力の形式は，左端に指定した変数を，次に代入演算子を，その右に一次元編集された変数の値を順に配置する。一次元編集の方式は，プログラム中の数式の構文則とほとんど同じで，演算子の優先順位は，高い方から，中 (\*\*)，乗除 (\*，/)，加減 (+，-) である。また，数式の中の数値と，算術変数の値の編集と変換は，すべて E 形式としてやる。なお，PRINT 機能では，次に述べる HPRINT 機能と同様，指定した変数と代入演算子を出力するが，これは，FORTRAN の WRITE，FORMAT 機能の代替機能であり，手軽で使いやすくなる。

#### 4.4 HPRINT 機能

HPRINT は，“hand-print”の意味で，計算機からの横一行ではなく，数学的表記法をとり入れた出力機能である。複雑な数式でも，それを見た人間がすぐ理解できるように出力することが重要であるが，これは，相当に難しい問題である。まず，人間が慣れ親しんでいる慣用的な数学の表記法を分析してみる。

- 使用文字，使用記号が豊富で，大文字の連用も含めて字体が多種である。
- E 型，横方向ともに文字や記号が規則正しく等間隔に並ぶが変化する。
- カッコの大きさや，分母のバーの長さの様に，囲み式の大きさや，分母の長さによって依存して定まるものがある。
- 簡単な記法に複雑な意味を持たせたり，同じ記法で何種類もの意味を持たせている。

次に，AL でとり入れた “hand-print” 機能について述べる。

- 1) べき演算 — 底を中央の行に配置し，指数はその上りの行のすぐ右のカラム以降に配置する。ただし，指数自身が分数式や，さらに，中演算子を含んでいる場合，指数は一行で出力する。
- 2) 除算演算 — 中央の行にハイフンと並べてバーとし，その上の二行で分子を，下の二行で分母を出力する。分母の中の中演算子は “hand-print” するが，分母が更に除算演算を含んでいる場合，それは，一行で出力する。

図3参照

- 3) 演算子の前後のスペース — 数式のリードオペレータが “+” であるとき，その演算子の両側に空白を置く。これは，多項式の様式データを考えると，慣用的な表記法では不要な積演算子 “\*” が出現して，各頂が横に長目になるので，頂の区切りを解りやすくするためである。

- 4) 特殊記号 — 図4 に示す様式で特殊記号を表現しやる。

ここで，HPRINT による二次元的出力の例を示し，PRINT 機能による一次元的出力との相違を示す。図5参照。

PRINT機能による一次元的出力が見にくい理由は、主に分母で、分母子(特に分母)の範囲が不明確で、カッコの数が多くなること、べき演算子“\*”が文字列を包む上、“\*”とまじりわしい事であった。この点で、HPRINT機能は、大きな効果と上げてくる。

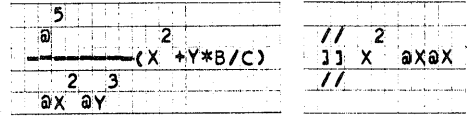
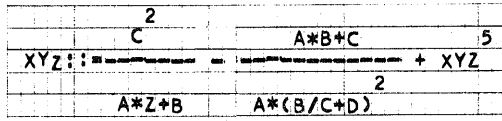


図3 HPRINTの基本形

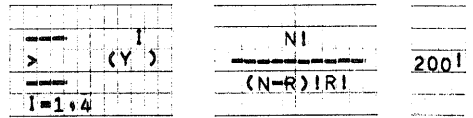


図4 HPRINTの特殊記号

$$\frac{(H+D(H))^{(1/2)} - 1)^2 * D1(H)}{2 * D(H)^{(1/2 * 3)}}$$

図5 HPRINTとPRINTの出力比較

$$-(H+D(H))^{(1/2)-1} * 2 * D1(H) / 2 / D(H)^{(1/2 * 3)}$$

### 5. エリア管理

教式処理を含めて、リスト処理では、実行時にエリアの割り付け、解放の管理を行うことが常識とされてきたが、その原因は次の二点にあるように思われる。第一に、扱っているデータが重要とするエリアの小ささが、第二に、その様なデータを扱うべく能率よくエリアに詰め込もうとすること、の二点である。

教式処理は、データの増加速度や寿命の点から考えると、エリア管理にあって相当きびしい条件を要求する処理である。以下に、AL-Cにおけるメモリの管理方法について述べる。

#### 5.1 主記憶領域の管理

AL-Cで式の計算を全部実行するとき、利用者と直接関係するデータは、二種類ある。第一は、プログラムのステートメントを翻訳した中間言語、第二は、変数テーブルや定数テーブル、第三は、不構造の教式データである。エリアの管理は、それぞれ別のエリアについて行なわれていたが、ここでは、最も重要な教式データ用エリアについて述べる。

- 1) まず、教式データエリアをセルと呼ぶ最小単位に分割し、全アウセルを順にポイントで結び、空きセルリストとして一本のリストにしておく。そして、リストの先頭セルと最終セルのアドレスと空きセル数を記憶しておく。
  - 2) セルに対する使用要求が来ると、空きセルリストから必要とする個数のセルを切り出す。また、不要となった領域は、不要となった時点ですぐ、空きセルリストに回収する。
- エリアの管理方法は、扱うデータの性質、すなわち、同一セルを多重に参照す



るか否か、あるいは、不要セルの回収機構を何とするか、ということも深くかがわっている。AL-Cでは、セルを多重に参照しない方式をとっている。この方式は、空きセル数が正しく把握できる点で優れているが、セルを多く消費する傾向がある。また、不要セルは、その時点で回収されるので、明示的にガーベジコレクションを指示する機能は設けていない。

## 5.2 ファイルとデータの退避

ALでは、主記憶上の数値データをファイルに退避しておく機能がある。

### 1) SAVE機能

会話実行中の利用者が、ある変数に代入されている値を当面の処理において使用しないと判断したとき、SAVE文により値をファイルに退避するとよい。この機能により、その数式が占めていた主記憶領域を回収して、エリアの有効利用を図ることが出来る。ファイルへ数式を退避するとき、主記憶上で未構造となっている数式をReverse Polish化してファイルに退避する。そして、値がファイルにある変数テーブルには、その表示とファイルアドレスを記憶しておくことにより、後に、この変数の値を参照すると、利用者が意識することなく値が復元される。

### 2) 自動SAVE機能

急激なデータ量の増加により、処理途中で空きセルが無くなってしまふと処理は続行不能となる。その場合、利用者に事態を通知し、利用者の指示を受けるとにより、システムが前述のSAVE機能を利用してその時点ですべての変数の持っている値をファイルに退避し、その後、中断していた処理を継続して実行することが出来る。この機能は、すでに終わった計算の結果を保証しながら処理の続行ができるという点で、会話処理には必要不可欠なものと言える。

## 6. AL-CFによる処理例

5次の分母を持つ $x$ の有理式の積分  $\int \frac{dx}{(x+1)^2(x^2+x+1)}$  を会話処理で計算する。計算方針としては、与式を有理式部分と超越部分に分解し、各係数を未定係数法により求める。その後、超越部分の積分を部分分数法により計算する。図-6参照。

## 7. おとがき

現在の数式処理システムは、歴史も深く、まだまだ不完全なものであるが、人間の判断力によりシステムの不備な部分を補なうことにより、かなり高度な機能を発揮することが可能である。この意味で会話型の数式処理システムは非常に有用なものと言える。また、端末は、現在のところ、汎用的な端末を使用しているが、専用端末、ディスプレイ等の使用はミニマニオンフェースの立場からは望ましい。ALは、DEMOS-ENVIRONMENTとして、今後よりサービスされることになっている。最後に、本研究の機会を与えられ、御指導いただいた横須賀通研、高原所長、高島部長、戸田統指役、藤原室長、そしてAL実用化グループ、日本電気株式会社関係各位に深謝する。

### <参考文献>

- 1) IBM SYSTEM/360 PL/I FORMAC User's Guide, IBM
- 2) MACSYMA REFERENCE MANUAL Version Six 1974, MIT
- 3) A. C. Hearn: REDUCE 2 USER'S MANUAL 2nd Edition
- 4) 滝沢, 阿部, 桂: TSSによる数式処理 第16回プログラミングシンポジウム報告集
- 5) 後藤, 金田: 記号及び数式処理システム HLISP-REDUCE ミニマニオンシステム研究会資料21

```

*** AL-C カイシ 75-12-25
:000010 /*INTEGRATE 1/POLYNOMIAL*/
:000020 INTGRT(1/((X+1)**3*(X**2+X+1)),X)?;

```

$$\frac{1}{(X+1)^3(X^2+X+1)}$$

```

:000030 U:=:(X+1)**2;
:000040 V:=:A*X+B;
:000050 P:=:(X+1)*(X**2+X+1);
:000060 Q:=:C*X**2+D*X+E;
:000070 S1:=:DIFF(V/U+'INTGRT(Q/P,X),X);
:000080 S1:=:SIMP(S1);
:000090 S1:=:CODEM(S1);
:000100 S1:=:SIMP(S1);
:000110 S1?;

```

$$\frac{(X+1)^2(E+D*X+C*X^2)-2*(B+A*X)*(X+X+1)+A*(X+1)*(X+X+1)}{(X+1)^3(X+X+1)}$$

```

:000120 S1:=:EXPAND(NUM(S1));
:000130 SIMP(S1)?;

```

$$A + E - 2*B + D*X - A*X^3 + C*X^2 + C*X^4 + D*X^3 + E*X^2 - 2*B*X + 2*E*X - 2*B*X^2 + 2*C*X^3 + 2*D*X^2$$

```

:000140 DCL F[4] ASSIGN2;
:000150 F[0]:=:CNSTT(S1,X)=1;
:000160 DO I=1 TO 4;
:000170 F[I]:=:COEFF(S1,SIMP(X**FORMLT(I)))=0;
:000180 PRINT(F[I]);
:000190 END;

```

```

F[1]:=:2*E+D-2*B-2*A+A=0
F[2]:=:E+2*D+C-2*B-2*A+A=0
F[3]:=:D+2*C-2*A+A=0
F[4]:=:C=0

```

```

:000200 SOLVE F[0],F[1],F[2],F[3],F[4] FOR A,B,C,D,E
: AND ASSIGN F[0],F[1],F[2],F[3],F[4];
:000210 V:=:REPLACE(V,A,RIGHT(F[0]),B,RIGHT(F[1]));
:000220 Q:=:REPLACE(Q,C,RIGHT(F[2]),D,RIGHT(F[3]),E,RIGHT(F[4]));
:000230 HPRINT(V,Q);

```

$$V := (-X) + \frac{-3}{2}$$

$$Q := (-X) + (-1)$$

```

:000240 ANS:=:V/U-INTGRT(1/(X**2+X+1),X);
:000250 SIMP(ANS)?;

```

$$-\frac{X+1/2+3}{(X+1)^2} - \frac{2*ATAN(1/3 \sqrt{2*X+1})}{3}$$

```

:000260 /END
*** AL-C 終了

```

図-6 AL-C の処理流れ

$\int \frac{dx}{(x+1)^3(x^2+x+1)}$   
 そのままでは INTGRT 関数で積分できない。

分母が square-free に分解できるから、未定係数 a ~ e を用いて、

$$\text{与式} = \int \frac{N(x)}{D(x)} dx \text{ が}$$

$$\frac{V(x)}{U(x)} + \int \frac{Q(x)}{P(x)} dx$$

となる事が解っている。

未定係数 a ~ e を求める為、上式を微分した形で恒等である、と置く。

通分と簡約化をし、未定係数を含む分子を抽出する。

この式が与式の被積分式の分子、1 と恒等である。

X の次数別に係数を比較して、5元1次の連立方程式を得る。

これを解いて、その結果を Q(x), V(x) のそれぞれに未定係数に代入(置換)。

確認の為、その両式を出力させてみる。

与式を変形して得られた形の積分を計算する。

ANS に代入された容を出力させてみる。

以上で完了。