

パーソナル・コンピュータで稼働する日本語プログラミング言語の開発

自然言語の利用でマン・マシン・インタフェースを改善する「日本語 AFL」

上田 謙一 松下技研研究開発部主任技師 菅野 淳 松下技研研究開発部技師 本田 邦夫 松下技研研究開発部

- データやコマンドを日本語化したのではなく、自然な日本語で記述できる本格的な日本語プログラミング言語「日本語 AFL」を開発し、試験稼働させている。
- 一般の人へのなじみややささという特徴を生かし、パーソナル・コンピュータ上で動かしているが、実用性は高い。日本語 AFL のコンパイラも日本語 AFL で記述した。
- 算術演算、論理演算、ファイル処理などだけでなく記号処理も強力で人工知能に適している。

半導体技術の進歩に伴い、パーソナル・コンピュータの性能は大幅に向上し、漢字がなまじりの日本語情報処理もパーソナル・コンピュータで可能となってきている。しかし、これまでの日本語情報処理は在庫管理や給与計算などにおける品名や氏名などのデータを漢字がなまじりの日本語で表現することにどまっていた。最近になって、この日本語による情報処理をプログラミングの分野にまで応用しようとする動きがでてきた。

これまでに開発されたプログラミング言語は BASIC、FORTRAN、COBOL などだが、これらは英語の構文が基礎となっている。欧米人にとっては親しみやすいものでも、日本人にとってはなじみにくく、これらのプログラミング言語を使うことからコンピュータは難しいものと感じさせてしまうことが多かった。

このため、パーソナル・コンピュータが安価で入手しやすくなり、各家庭で利

用できるにもかかわらず、普及が、いままではなかった。そこで、パーソナル・コンピュータをだれでも使いこなせるようにするため、日本語でプログラムする言語の開発が期待されていた。

これまでに、日本語プログラミング言語として、いくつか開発されているが、これらは COBOL 言語を基礎とし、COBOL の文法に導かれた言語となっており、自然な日本語とほゞ違い欠点があった。このような理由から筆者らは「日本語 AFL (A Fundamental Language)」と呼ぶ新しい日本語プログラミング言語を開発した。現在、当社でこの試験システムが順調に動いている。

自然な記述

図 1 は日本語 AFL による簡単なプログラム例である。最初のスタートメントで、面積の算出式が「縦×横」と定義される。②のスタートメントで縦と横の値

図 1 ●日本語 AFL プログラム例①
①は空白を示す。

```
スタートメント
① 面積=縦×横
② 縦=100, 横=50
③ 面積を計算する。
```

が決まり、③のスタートメントで面積が計算される。計算結果の 5000 は「結果」という名前がついて保持される。

図 2 の例についても読者は容易にこのプログラムが何をしようとしているかわかるであろう。①と②のスタートメントはそれぞれ、直方体の体積と円柱の体積の算出式を定義している。③-⑤のスタートメントで直方体の体積を計算し、⑥-⑧のスタートメントで円柱の体積を計算している。

このように日本語 AFL では算出式の定義およびデータがプログラミングの実行段階で設定できるので、自由度の高いプログラム表現が可能となっている。



図2 ●日本語 AFL プログラム例②

```

ステップ番号
① ぬ方体は□ (縦*横*高さ)
② 円柱□は□ [3.14*半径*半径*高さ]
③ 体積□は□ぬ方体
④ 縦=100, 横=200, 高さ=100
⑤ 体積□を□計算する
⑥ 体積□は□円柱
⑦ 半径=20, 高さ=40
⑧ 体積□を□計算する
    
```

図3 ●日本語 AFL プログラム例③

```

ステップ番号
① N□は□
② 階乗□は□
③ 「階乗□は□N□を□求める」
④ 「N □は□」を□挿入する
⑤ □を□10□回□繰り返す
    
```

図2の③と⑤のスタートメントでは、体積をぬ方体あるいは円柱として「体積を□計算する」という同一のスタートメントで答えを求めている。

図3は10! (階乗) を求めるプログラムである。①と②のスタートメントはデータの初期値を設定するものであり、こ

こではNと階乗に1という初期値を与えている。③と④のスタートメントは次のように、
階乗=階乗*N、
N、N=N+1
と表記することもできる。⑤のスタートメントは「0!、0!」で表された機能を10回繰り返すことを示す。これを実行して10!の値が求められる。

このように日本語AFLではGOTO文に相当するスタートメントは存在してい

プログラミング言語となっている。

また、日本語AFLは漢字かなまじりの文字列を操作する記号演算処理が簡単に行える。表1では日本語AFLとほかのプログラミング言語とを比較して、それぞれの言語の特徴を記述した。

- 文法仕様
- 日本語AFLは構造的プログラミング言語となっている。その基本文として以下の4種類で構成される。
- (1) 代入文
 - (2) 実行文
 - (3) 選択文
 - (4) 繰り返し文

これらの具体的なスタートメント表現は図4に示す。図4において①の代入文における「」は直接定数演算子であって、この例の場合には「りんご みかん 桃 梨」というデータが「果物」という名前の場所に格納されることを示している。もう1つの例の場合は「私」という名前の場所に格納される。この場合、「果物」は「りんご みかん 桃 梨」となっているの

ない。日本語の文章を書く場合と同様に、機能を前に記述していくことによりプログラマができる。これらの例からわかるように、日本語AFLによるプログラムのコードは日本人にとって自明に近いものであり、コンピュータに対して特別な知識がなくとも容易に習得できるプ

表1 ●汎用プログラミング言語と日本語 AFL との比較表

プログラミング言語の種類	代表的な言語	特徴
英語プログラミング言語	広く普及している言語 FORTRAN COBOL BASIC	現在のコンピュータ利用者は大部分がこれらの言語を用いている。文法は比較的簡明で容易に習得可能である。言語によって用途が限られている。
	豊富な機能を持っている言語 PASCAL LISP PROLOG	人工知能の研究などに利用されている機能の豊富な言語である。文法がかなり高度であり、習得には専門知識が必要である。
	パソコン向けの容易な言語 VISICALC PIPS EPOCALC	定型的な処理について、ユーザーがいちいちプログラムすることなく、コマンド入力によって、コンピュータが自動的に処理を行うもので、その分用途が限定される。
日本語プログラミング言語	COBOL言語の日本語表記 PRECOBOL (三菱電機) PAGE (富士通) 漢字COBOL (キヤノン、日立製作所)	日本語表記により、COBOL言語に比べて読みやすくなった。しかし、COBOL言語に継承されていること、全部が日本語表記になっているわけではないなどの問題点がある。大規模コンピュータ用に開発されている。
	自然言語に近い日本語表記 日本語AFL	自然言語に近い日本語表記で読みやすい。文法は簡明で容易に習得可能であり、文字操作を中心に豊富な機能を持つ。パソコン用に開発されている。

図4 ●日本語 AFL の基本文

①代入文	……果物 ₁ は「りんご ₁ みかん ₁ 桃 ₁ 梨 ₁ 」 果 ₁ は果物 ₁ である。
②実行文	……N ₁ に1 ₁ を1 ₁ 加える。 果 ₁ を1 ₁ を1 ₁ 計算する。 果 ₁ を1 ₁ を1 ₁ 実行する。
③選択文	……品名 ₁ が果物 ₁ に含まれる ₁ か ₁ 表示し。 正しいならA ₁ を1 ₁ 実行し。 間違いないならB ₁ を1 ₁ 実行する。
④繰り返し	……A ₁ を1 ₁ を1 ₁ 回繰り返す。 A ₁ を1 ₁ を1 ₁ 回繰り返す。

んご みかん 桃 梨」ということとなる。この例において「である」という述語動詞が使用されているが代入文の場合にはこの述語動詞を省略できる。「ステートメントの区切りは、または、」の句読点を用いられる。単語と助詞の区切りにはブランクが使用される。図4②の実行文において、最初の例の「N に 1 を 1 加える」で、数値データ「1」は直接変数として取り扱われ「[1]」と同等の意味を持っている。

「計算する」と「実行する」との違いは次のようである。「計算する」という場合はその目的語の「果上げ」が「単価×数量」と、単独のステートメントの一部を構成し、演算結果は「結果」という名前をついた場所に格納される。これに対し、「実行する」の場合は目的語の「定義文」が複数のステートメントから構成されたプログラム・イジュールとなっている。

③の選択文で、最初のステートメントが実行されると、「品名」が「果物」に含まれるかどうかを調べ、「品名」が「果物」に含まれる場合は「A」という名前のプログラム・イジュールが実行される。「品名」が「果物」に含まれていない場合は「B」

という名前のプログラム・イジュールが実行される。選択文は通常、この3つのステートメントで構成される。「正しいなら……および間違いないなら……」のステートメントはどちらか一方を省略することができる。省略した場合、判断結

果が省略した方のステートメントを満足するときは何もしないで次へ進む。

さらに、この例では「正しいなら」、「間違いないなら」の後は「A を 実行する」、「B を 実行する」というようになっているが、これらの代わりに「[]」でくくられたプログラム・イジュールを直接記述できる。

図4の④の繰り返し文において、最初のステートメントの例では繰り返しの条件として「N が 5 と 等しい まで」となっている。このためには N が A

というプログラム・イジュールの中で値が定められるようにしていなければならない。

以上が日本語 AFL の基本文の仕様である。

データはバイト単位で扱う

日本語 AFL はバイトを単位とした文字データを処理の基本単位として扱っている。漢字、ひらがなを半数字と同等に取り扱うことができる。このため、日本語 AFL は文字かなまじりの文字列を結合したり、消去したり、検索したりという記号演算が簡単にできる。このため、地盤動向も数多く用意されている。

図5はこの記号演算ステートメントについての例をいくつか示したものである。

日本語 AFL では記号演算の単位として「文字単位」、「項単位」、「行単位」の3種類がある。「文字単位」とはバイトごとの文字データを単位とするもので、「追加する」、「削除する」などの記号演算を文字単位で行う。「項単位」とは、ブランク記号などの特殊な記号で区切られた項

図5 ●日本語 AFL の記号演算記述ステートメント

① A ₁ にB ₁ を1 ₁ 追加する。	A=[ABC], B=[DEF], とすると → A=[ARCDFF], となる。
② A ₁ からB ₁ を1 ₁ 削除する。	A=[ABCD], B=[BC], とすると → A=[AD], となる。
③ A ₁ からB ₁ の1 ₁ 前1 ₁ を取り出す。	A=[ABCD], B=[CD], とすると → 結果=[AB], となる。
④ A ₁ から[C] ₁ の1 ₁ 個1 ₁ を取り出す。	A=[ABCDEF], とすると → 結果=1, となる。
⑤ A ₁ のB ₁ 番1 ₁ を取り出す。	A=[ABCDEF], B=5, とすると → 結果=[E], となる。
⑥ A ₁ の2 ₁ 項目1 ₁ から1 ₁ 項目1 ₁ をB ₁ で1 ₁ 置き換える。	A=[ABC, DEF, GHIJK], とすると B=[LMN], とすると → A=[ARCLMN, GHIJK], となる。

寄稿

図6 ●日本語 AFL の算術演算記述ステートメント

- ①AにBを加える。 A=A+B.
- ②AからBを引く。 A=A-B.
- ③AにBを掛ける。 A=A*B.
- ④AをBで割る。 A=A/B.

図7 ●日本語 AFL の比較演算記述ステートメント

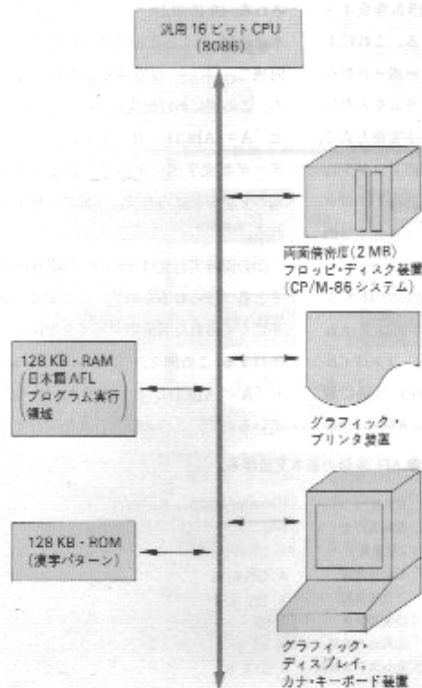
図7-A ●算術比較演算

- ①AがBより大きい場合に判断し、・・・
(A=100,B=50,とすると、結果=1(正しい)となる)
- ②AがBより小さい場合に判断し、・・・
(A=100,B=50,とすると、結果=0(間違い)となる)
- ③AがBと等しい場合に判断し、・・・

図7-B ●論理比較演算

- ①AがBに含まれる場合に判断し、・・・
(A=[ABCDE],B=[ABCDE],とすると、結果=1となる)
- ②AがBと一致する場合に判断し、・・・

図8 ●日本語 AFL の実行環境



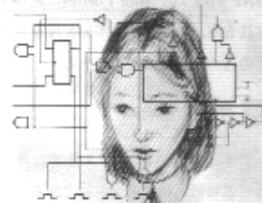
数の文字データを項と呼び、この項を演算処理の単位とするものである。1行単位はLf:コドなど、行区切り記号の取り出した行データを単位とするものである。①-③は文字単位の記号演算の例であり、④は項単位でデータを置き換える例である。

日本語 AFL ではこのほかに算術演算、比較演算としてそれぞれ、図7に小さな記述ができる。算術演算は「+」、「-」、「*」、「/」などの演算子を使用してステートメントを記述することができ、「加える」、「引く」といった述語動詞を使用する必要はない。

算術演算に使うデータは整数データや実数データの区別がなく、符号や小数点が付加された文字データそのものが数字データとして算術演算に用いられ、結果も文字データとなる。このために、算術演算の処理速度は遅くなるが、データタイプを意識してプログラムを作成する必要がない。演算精度は上記値での処理が許されるまで自由に設定可能である。

以上が日本語 AFL の文法仕様で

夢のような話を実現する「ベーシック・イン」



1台が日本語ワープロにも、A/D、D/Aを使って計測・制御装置からのデータ取り出しにも、さらに取り出したデータを本社の研究室に送るオンラインにも、これらの完成までのサポートして、コンピュータ2セット分も含めて月のリース料が全部で15万円です。



大巾値下げ

¥1,058,000

243M/20 価格大巾値下げ旅行/業務用機種
ピップスのデータ9日本語ワープロ作成したデータがBASICで読めるソフトが付きます。
●BASIC(標準) ●グラフィック(標準) ●漢字BASIC ●日本語ワープロ ●BASIC-II ●高速漢字BASIC ●漢字ワープロ ●漢字PIPS ●フォント/コピ/ホリなど ●会計(簿記) ●給与(当社開発) ●マルチである ●通信ができる ●各種インターフェース(A/D、D/Aなどがつなげられる) ●CPU-192KB(漢数字) ●KS282C-4ch ●5-100バス ●その他243シリーズ20機種

SOROを買うなら
コンピュータの付加価値を高める

ベーシック・イン BASIC-inn

本社 ●ニッゾー貿易株式会社 ●☎045(662)8552
ベーシック・イン 東京 ●東京駅前東横線1344ビル ●☎03(436)3091
ベーシック・イン 神奈川 ●横浜市中区新港1-10-1 ●☎045(641)0985
ベーシック・イン 大阪 ●大阪市東区1-10-1 ●☎06(271)6521
●資料請求番号1199

日本語AFL

あり、自然言語に近い記述でプログラムが作成できる。なお、日本語AFLのBNF (Backus Naur Form) 記法による文法表現、および、基本形表現を筆者執筆の後の付表にまとめている。

実行環境

日本語AFLはだれでもすぐに使いこなせる言語という開発目的があることから、日本語AFLを実行できるシステムはパーソナル・コンピュータ・システムに限られる。日本語AFLの実行環境を図8に示す。これは現在の標準的なパーソナル・コンピュータ・システムの構成となっている。

CPUは現在の標準的な16ビット・マイクロプロセッサであるインテル社の8086を使用し、主記憶は128KB(1キロバイト)のRAM(ランダム・アクセス・メモリー)を装備しており、文字ハ

ード・インタリー・メモリーに格納されている。周辺装置としては両面倍速率のフロッピー・ディスク装置が2台(合計容量2MB)、漢字入出力のためのグラフィック表示可能なCRT(ブラウン管)ディスプレイ、それにプリンタが装備されている。以上が現在動かしている日本語AFLの実行システム構成である。

このシステム構成での日本語AFLの実行手順は図9である。まず、電源ONによるスタートでCP/M-86が稼働する。CP/M-86はパーソナル・コンピュータの標準的なOSとしてよく知られており、日本語AFLはこのOSのもとで稼働する。

CP/M-86が動けばコマンド入力により、日本語AFLのプログラム作成モードになる。これにより、キーボードからプログラムを入力し、カナ漢字変換方式で日本語のプログラムが作成され、フロッピー・ディスクへ格納される。

作成した日本語AFLのプログラムは次に、コンパイルしてAFL言語に翻訳する。AFL言語は

後で説明する。この翻訳されたAFL言語のオブジェクト・プログラムがAFLインタプリタ・システムにより実行される。

以上が日本語AFLの実行環境と具体的なプログラムの実行手順である。このように、現在、日本語AFLは標準的なパーソナル・コンピュータ構成である16ビット・マイクロプロセッサ8086およびOSとしてCP/M-86を使用したシステムのもとで稼働するようになっているが、AFLインタプリタ・システムを修正することにより既存のシステムで稼働させることも可能である。

コンパイル後にインダプリットする

AFLインタプリタで実行されるAFL言語は図10に示すような基本文法で構成される。図10において、10の代入文は日本語AFLの文法仕様で述べた代入文と同様で、「:=」は直線定数演算子である。この例にあげた代入文が実行されると「A」[ABCD]、「B」[EFG]、というデータが文字データの形でPという名前のタブが付けられて、P配列へ格納される。

12の演算実行文は「<>」が関係演算子と名づけられるもので、この関係演算子に与えられる名前がデータを参照して実行する。この例でPは10の代入文により「A=[ABCD]、B=[EFG]」となっているので、この関係実行文を実行する

図9 ● 日本語AFLの実行手順

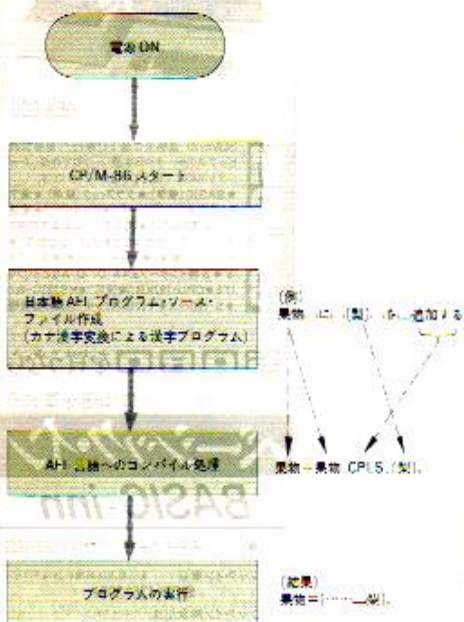


図10 ● AFL言語の基本文法体系

(1) 代入文	P := (A=[ABCD], B=[EFG], ...)
(2) 関係実行文	(P)
(3) 2項演算文	
① 加算	A + B
② 減算	A - B
③ 乗算	A * B
④ 除算	A / B
⑤ 乗算	A * B
⑥ 除算	A / B
⑦ システム制御	A .SYS B

寄稿

と、結果として「ABCD」というデータがAという名前のところに格納され、「EFG」というデータがBという名前のところに格納される。

関数演算子の機能は、このほかに文字間換算符としての機能がある。これは、たとえば、

S=A+C+B*D、というステートメントが実行時の「の内容により展開され、Iが1のときは

S=A+1*B1、

Iが2のときは

S=A+2+B2、となって実行されるものがある。

3項の2項演算文は2つの演算符「+」を持つ2項演算子で構成される。機能向から分類すると、図10に示す3種類に分かれる。最初の3項の例では、「CPLS」がキートラック・プラスと呼ばれる型写地

使用の2項演算子で、Aが「ABCD」、Bが「EFG」とすると、この2項演算文の実行結果は「ABCDEFG」となる。

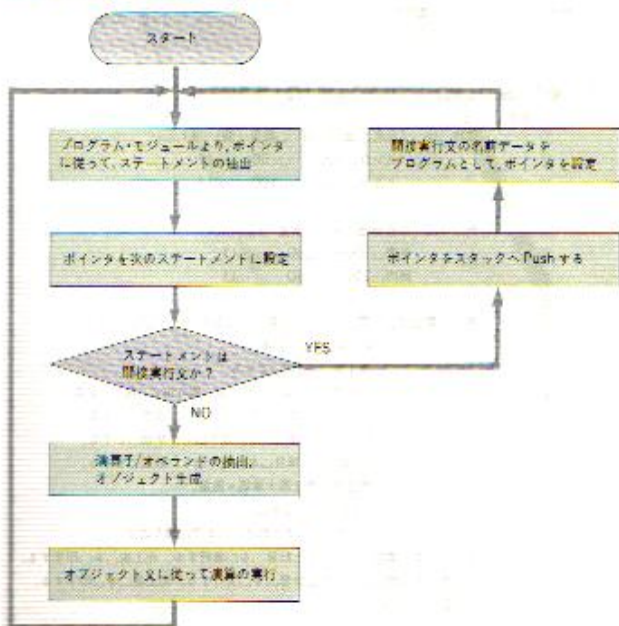
この場合、C=A,CPLS,B、というステートメントとすると、計算結果がCという名前のところに格納される。さらに、

D=A,CPLS,B,CPLS,C、のように2項演算子を1つのステートメント中に複数使用することができる。以上がAFL言語の基本構文である。

AFLインタプリタの処理方式の概要は図11に示すようになっている。図11からわかるように、このインタプリタの特徴は通常のインタプリタがソースプログラムをいったん中間コードに変換し、中間コードをインタプリットするのに対し、ソースプログラムを直接インタプリットして実行していることである。

日本語AFLのプログラムはコンパイル

図11 ● AFL言語インタプリタの概要



我が
ビジネスに
勝算あり。



【いま、他社に先がけて】
PC-9800用ソフト、
高電社から続々登場

高電社
マイレター98 (日本語ワード
プロセッサ)
PC-9800 価格 ¥69,000

高電社
PARAM K2・K1
PARAM K1 (価格 ¥49,000) PC-9800 ¥49,000
PARAM K2 (価格 ¥69,000) PC-9800 ¥69,000

システム販売からソフト開発
ソリューション教育まで
高電社
株式会社 高電社
NECマイコンショップ! システムイン 高電社
〒100 東京都千代田区千代田1-10-10 高電ビル3F
大阪駅前店: システムイン 高電社
〒542 大阪府大阪市淀川区西淀川1-10-10 高電ビル3F

日本語 コストレジ

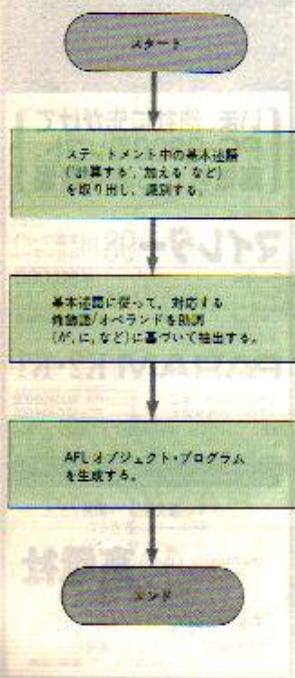
ル処理によって、このAFL言語のプログラムに変換され、インタプリタで実行されるが、このコンパイラはAFL言語で作成されており、図12に示すような処理を行っている。図13はコンパイラにより、日本語AFLのプログラムがどのようなAFL言語のプログラムへ変換されるかの1つの例である。初めに、基本文語「判別し」を識別し、この条件語である「含まれる」を取り出す。次に、2つの判別データ名である「品名」および「果物」を取り出し、

結果=果物、CNUM、品名、というAFL言語を生成する。

次に、「正しい なら ー」、「間違いない ー」の構文を識別し、

C1=A、C0=B、<C<結果>>、というAFL言語を生成する。正しい場合は実行すべきプログラム・モジュール名

図12 ●日本語AFLのコンパイル処理



解説

AをC1に代入し、誤りの場合は実行すべきプログラム・モジュール名BをC0に代入し、同様実行文

<C<結果>>、で「結果」の値によりC0またはC1が実行される。

以上が日本語AFLの処理系についての概要である。このように、日本語AFLはAFL言語を使用することにより、パーソナル・コンピュータで稼働できる簡単なシステムとして実現されている。

機能説明のためのプログラム例

日本語AFLはGOTO文の存在しない構造的プログラミング言語で、4種類の基本文から構成されている。このような日本語AFLを用いてプログラムをど

のように作成したらよいかについて、以下、具体的なプログラム例に基づいて説明する。

図14はスーパー・マーケットなどの売上高の算出プログラム例である。図14に示すステップ番号①~④は果物について、1行ごとにリンゴ、ミカンといった品目の単価がいくらか、数量がどのくらいか、の情報を格納しているデータである。⑤はデータ中の各項目がどのような内容かを示している。ここでは1項目が品名で、2項目が円を単位とする単価であり、3項目が個数を単位とする数量であることを示している。具体的な果物に関するデータは2行目以後に格納されている。

同様に⑤~⑧は肉に関するデータと格

図13 ●日本語AFLプログラムをコンパイルし、AFLのオブジェクト・プログラムをつくる

図13-A ●日本語AFLプログラム

```
品名C1が果物C0に含まれるC1を判別し、
正しい ならC1をC1実行し、
間違いない ならC0をC0実行する。
```

図13-B ●AFLオブジェクト・プログラム

```
結果=果物、CNUM、品名、
C2=A、C0=B、<C<結果>>。
```

図14 ●日本語AFLによるプログラム例「売上高算出プログラム」

ステップ番号

- ① 果物C1はC1 [品名C1 単価 (円) C2 数量 (個) C3] C1
- ② リンゴC1 100 C2 50 C3 C1
- ③ ミカンC1 90 C2 100 C3 C1
- ④ ...
- ⑤ 肉C1はC1 [品名C1 単価 (円) C2 数量 (kg) C3] C1
- ⑥ 牛肉C1 3000 C2 80 C3 C1
- ⑦ 豚肉C1 2000 C2 50 C3 C1
- ⑧ ...
- ⑨ 売上計算C1はC1 [データC1のC1行C2をC1取り出し、C1Nとする、
- ⑩ K1をC1とする。
- ⑪ 売上単価をC1とする。
- ⑫ [データC1のC1行C2をC1取り出し、S1とする、
- ⑬ S1のC1C1項目C1をC1取り出し、品名C1とする。
- ⑭ S1のC1C2項目C1をC1取り出し、単価C1とする。
- ⑮ S1のC1C3項目C1をC1取り出し、数量C1とする。
- ⑯ 売上高=C1単価×数量、
- ⑰ K=C1+1、C1をC1
- ⑱ KがC1Nより大きいC1までC1繰り返す。]
- ⑲ データC1をC1果物C1とし、売上計算C1をC1実行する。売上高C1をC1印字する。
- ⑳ データC1をC1肉C1とし、売上計算C1をC1実行する。売上高C1をC1印字する。

(C1は実行コード)

寄稿

納している部分である。この例では「牛肉が単価 3000 円で 80 kg」の売り上げがあり「豚肉が単価 2000 円で 50 kg」の売り上げであったことを示している。

図 14 の①～③は売上高の算出方式が売上計算という名前で定義されている。①はデータの数を取り出し、N と定めており②の繰り返し回数の判定に使われている。③は必要なデータを取り出すためのポインタ K に初期値として 2 を与えている。④は売上高の初期値を 0 に設定する。⑤では売上高の計算するデータを K 行目から取り出し S という名前にしていく。最初は K が 2 であるから、果物の 2 行目のデータである「リンゴ 100 50」が取り出されて S となる。⑥～⑧では S のデータから品名、単価、数量の値が取り出され売上高が計算される。最初は単価が 100 円で数量が 50 個であるから、売上高は 5000 円となる。

⑨では次に参照するデータを取り出すために、ポインタ K を 1 増加している。この⑤～⑧で示される売上高の計算を繰り返すまで繰り返すことにより全体の売上高が計算される。⑩はこの売上高の算出式に従って果物、肉といったデータについて売上高を計算し、リストに印字している。以上が日本語 AFL による売上高の算出プログラム例である。

図 15 はクラスの各人について身長、体重のデータをもちに、各人の肥満度を算出し、新しいクラスのデータを作成するプログラム例である。日本語 AFL では、任意の形式で、データを格納できることを示した。

図 15 の①～③は A 組というクラスの各人の名前、身長、体重についてのデータである。この例でわかるように、おのおのデータ長は何文字以内でなければ

図 15 ●日本語 AFL によるプログラム例「肥満度のデータを作成するプログラム」

ステップ番号	プログラム
①	A組(身長) (青山) 160 70 50 LF
②	長谷川 170 80 50 LF
③	肥満度(身長) (身長 - 100) * 0.9 * 100;
④	A組の各行の③の数値を取り出し、N とする。
⑤	K は 1;
⑥	新A組(身長) ;
⑦	[A組から K 行目を取り出し、S とする。
⑧	S から 1 項目目を取り出し、名前 とする。
⑨	S から 2 項目目を取り出し、身長 とする。
⑩	S から 3 項目目を取り出し、体重 とする。
⑪	肥満度を計算する。
⑫	S と A 組の 1 項目と結果を結合し、S とする。
⑬	新A組の K 行目に S を入れる。
⑭	K に 1 を加え、K とする。
⑮	N と比較し、N < K なら繰り返す。
(結果)	新A組 = (青山) 160 70 50 129 LF
	長谷川 170 80 50 125 LF

GENERAL キャラクタ ディスプレイ 装置

●拡張性を持った高級機

LPD-400形



- 特徴
- システム拡張可能な構造の設計になっている。
 - 表示文字は 7 × 10 ドット・マトリックスの高解像度高級機です。
 - 使用条件の改善は、コストのない物出しの付添いになっています。
 - メニュー・スクロール機能がつまみやすい。
 - ファンクション・キーが標準装備になっており、アプリケーション・キーにも使えます。

機能の豊富な高級機

KDE-830形



コンパクトな汎用タイプ

KDE-820形



- 全機種共通仕様
- 画面: 1024 × 1024 (70 CRT) 液晶表示
 - インターフェイス: RS-232C/20mA/シリアル・ポート
 - 印刷文字数: 漢字、カタカナ、英数字、半角カナ
 - 印刷速度: 1000文字/分

▲対応機種: 富士通 FT-100

■販売代理店

◆ 兼松エレクトロニクス株式会社
東京都中央区日本橋2丁目1番1号 兼松ビル
TEL: 03-3542-1111 FAX: 03-3542-1112

◆ 住商電販光株式会社
東京都中央区日本橋2丁目1番1号 兼松ビル
TEL: 03-3542-1111 FAX: 03-3542-1112

◆ 日商エレクトロニクス株式会社
東京都中央区日本橋2丁目1番1号 兼松ビル
TEL: 03-3542-1111 FAX: 03-3542-1112

ゼネラル
CRTディスプレイ

株式会社 ゼネラル 札幌支店 営業部
〒050 札幌市中央区南一条西五丁目1番1号
TEL: 011-247-1202 FAX: 011-247-1203



ならないといった制限は不要で、自由に与えることができる。

①は肥満度の算出式を定義している。②はA組のデータの総数を求めて、それをNということに格納する。この場合、1行が1人のデータとなっている。③④は初期値設定である。⑤-⑧が各人の肥満度を算出して、新しいクラスのデータを作成するプログラム・モジュールである。⑨-⑬のストートメント全体はNの値の回数だけ繰り返して実行されることにより、求められる。

⑭は各人のデータを順次取り出し、Sということへ格納するもので、例えば最初は①において、K=1であるので①のデータから、Sには「青山16070」が格納される。⑮⑯はこのようにして取り出したデータから、さらに、名前、身長、体重と、各データを取り出すものである。

⑰はこのようにして取り出したデータを使って、①で定義した算出式に基づいて、肥満度を計算する。⑱は⑱の計算結果である肥満度をSに追加するもので、例えばSが「青山16070」になると⑱の実行により、Sは「青山16070129」ということになる。

⑲はこのようにして、肥満度が追加された各人の新しいデータを新A組のデータとして順次登録していく。⑳Kをインクリメント（増加）することによって、次の人のデータを取り出す。㉑はこのようにしてA組の全員について、繰り返し実行することによって、新A組のデータを作成する。このようにして図15に示す新A組のデータが求められる。

数値データも文字データとして扱う

図16は π （円周率）を計算するプログラムの例である。データ・タイプを区別せず、簡単に算術演算が行え、また、演



算精度を自由に設定できる。FORTRAN、BASICなど従来のプログラミング言語では算術演算において、整数型や実数型というデータ型を意識してプログラムをつくらなければならない。さらに、演算精度は通常、有効桁数が7桁で、倍精度でも15桁に制限されている。

これに対し、日本語AFLでは文字データを処理の基本としており、整数型や実数型のデータ・タイプは存在せず、数値データも文字データの一部として取り扱われる。また、演算精度は上記以上のユーザー領域の範囲内で自由に設定できる。現在のシステムでは約1万桁までの演算精度が可能である。

図16で示されるプログラム例ではマチンの式を用いて π を計算している。つまり、

$$\frac{\pi}{4} = 4 \times \arctan\left(\frac{1}{5}\right) - \arctan\left(\frac{1}{239}\right)$$

の式をもとにして、

$$\arctan\left(\frac{1}{m}\right) = \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)m^{2k+1}}$$

の展開式を用いて π を計算する。

図16の①は算術演算における有効桁数

を宣言するストートメントであり、ここでは有効桁数が100桁であることを示している。②-④はデータの初期値を設定し、 π の値が1つ前の値と等しくなるまで⑤-⑧で示される π の計算式を繰り返すことにより、指定された有効桁数の π の値を求めている。

実務への適用

これまで、プログラム例の述べてきたように、日本語AFLは自然言語に近い表記で簡単にプログラムを書くことが可能である。そこで、日本語AFLの実用性をみるため事務処理への応用と、コンパイラ作成への応用という2つの違った分野への適用例を示す。

事務処理

具体的な事務分野への応用として、従来目録立貯金システムを取り上げてみる。このシステムはかつて、AFL言語を用いてIBM 370/125の上で稼働していたシステムであるが、これを日本語AFLで書き換えたプログラム例で説明する。従来目録立貯金システムには口座開設、日

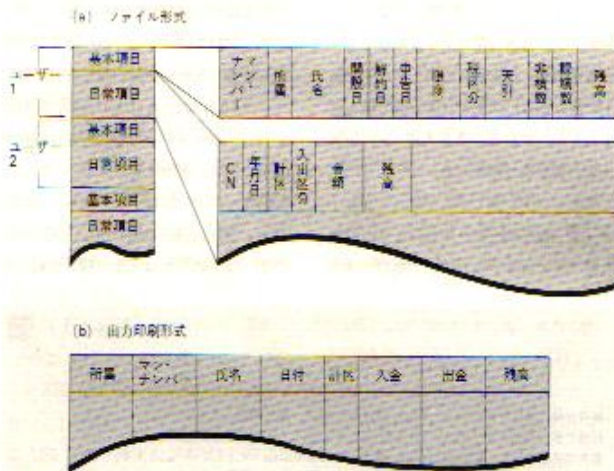
常入出金、伝票、残高表、台帳、利息計算、など多くの種類がある。ここでは台帳について取り上げ、そのプログラムの説明を行う。

台帳は各人の入出金の動きを各月ごとに見るもので、ディスク上のファイルから各人の日常の入出金の動きを読み、出力形式を作成してリストに出力するも

図16 ●日本語AFLによるプログラム例「 π を計算するプログラム」

ステップ番号	
①	100桁の演算精度で演算する。
②	$\pi = 0$
③	$N = 1$
④	データ1 = 1/5
⑤	データ2 = 1/239
⑥	$\pi = \pi + 4 \times$
⑦	$\pi = \pi - (16 \times \text{データ1} - 4 \times \text{データ2}) / N$
⑧	データ1 = データ1 / (5 * 5)
⑨	データ2 = データ2 / (239 * 239)
⑩	$N = N + 2$ (1を付)
⑪	π が10桁と等しいまで繰り返す。
(結果)	$\pi = [3.141592 \dots]$

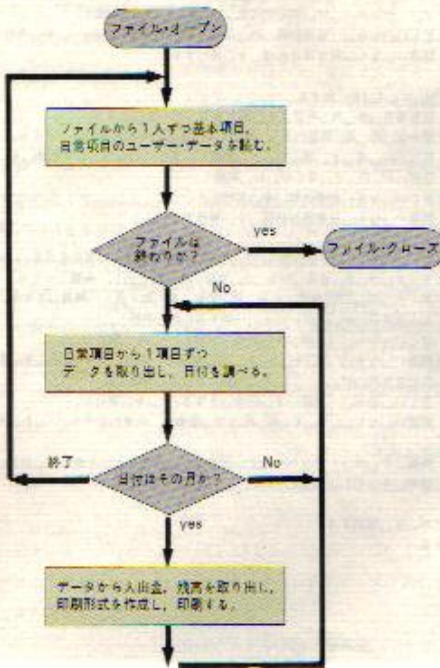
図 17 ●積立貯金システム



のである。ディスク上のファイル。出力の印刷形式は図17のようになっている。処理内容はまず、基本項目、日次項目を読み、日次項目からその月のデータだけを取り出し、入出金区分に応じて入金が出金を判定し、それによって出力の印刷形式をつくり、印刷し、これをファイルの終わりまで繰り返す。ファイルの終わりは特殊なマン・ナンバー・コードで検出する(図18)。

このプログラムは図19のようになる。ステップ①～④が台帳処理の全体を示す。

図 18 ●積立貯金システムのフロー・チャート



第4世代コンピュータのソフトウェアとして注目されているリレーショナル・データベース管理システムをパソコンで実現し、しかも「データ・クック」自由

DATA ACE

1. 処理できる本格的な日本語情報処理システムで、専門知識を一切必要としない完全なプログラマレス機能により、複雑なデータの条件検索、リストの作成、さらにデータの追加・削除・変更が簡単な操作で思い

DATA ACE

通りに実行できます。また、ユニークなアプリケーションシステム開発機能により、オフィス業務を効率よく迅速にオートメーション化、あらゆるビジネス・コースに低コストで威力を発揮します。

DATA ACE

運用機種 N5200-05、COMPORT、FM-11、MULTI116、PC-9800、その他OH/M-86パソコン
 ●ソフト工学研究所 (株)SOFELが販売代理店を専任して、
 ●CP/M3.2のシステム・リサ、データ管理技術で、
 ●DATA-ACEは Computer Software Design 社が専
 売品です。

SOFEL
 株式会社
 ソフト工学研究所
 〒101 東京都千代田区外神田3-11-2
 ロックビル2F TEL.03(261)1195

資料請求番号125

ステップ⑩-⑬が個々の日常項目の処理を、ステップ⑭-⑯は印字処理の具体的な内容を示している。

このように日本語 AFL プログラムは全体から部分へとモジュール化された構造的なプログラムとなっており、プログラム・リストがそのままコメントの一部となりうる。

ステップ⑱で経理台帳ファイルがオープンされ、ステップ⑳で台帳処理が実行される。経理台帳にはあらかじめ作成さ

れている CP/M 86 の下でのファイル名が (A:KEIRI-SYS) のように格納されている。ステップ⑰、⑱でフロッピー・ディスクの経理台帳ファイルから 1 人分の基本項目、日常項目のデータを上記順序に読み込み、ステップ⑳-㉑で読み込まれたデータから、マンナンバー、所属、氏名の値が取り出される。

ステップ㉒-㉓でマンナンバーが終了マーク (99999999) と等しいときは経理台帳ファイルをクローズして処理全体を

終了し、通常のマンナンバーのときは日常項目処理を実行する。ステップ㉔、㉕で日常項目から日付のデータを取り出し、ステップ㉖で該当する日付から印字処理を実行する。ステップ㉗-㉘で日付が終了マーク (00) のときは次の人の台帳処理を行い、終了マークでないときは継続する日常項目の処理を行う。

ステップ㉙-㉚はデータから、入出金区分、金額、残高を取り出し、入出金区分が「+」のときは入金の欄に金額を書

図 19 ●経理プログラム

ステップ番号	台帳処理 (H)	印字処理 (H)
①	経理台帳のファイルを開く。読み、基本項目とする。	
②	経理台帳のファイルから 58 行目を読み、日常項目とする。	
③	基本項目の 11 番目から 17 文字を取り出し、マンナンバーとする。	
④	基本項目の 18 番目から 23 文字を取り出し、所属とする。	
⑤	基本項目の 12 番目から 15 文字を取り出し、氏名とする。	
⑥	N を 0 とする。	
⑦	マンナンバーが 99999999 と等しいかを判断し、	
⑧	正しいならば経理台帳のファイルを終了する。H を実行し、	
⑨	間違いないならば日常項目処理 H を実行する。	
⑩	日常項目処理 H: [N を 0 とする、	
⑪	日常項目の N 行目を取り出し、データとする。	
⑫	データの 4 番目から 7 文字を取り出し、日付とする。	
⑬	日付が 00 と等しいかを判断し、正しいならば印字処理 H を実行する。	
⑭	日付が 00 と等しいかを判断し、	
⑮	正しいならば台帳処理 H を実行し、	
⑯	間違いないならば日常項目処理 H を実行する。	
⑰	[データの 8 番目から 10 文字を取り出し、入出金区分とする。	
⑱	データの 11 番目から 16 文字を取り出し、金額とする。	
⑲	データの 17 番目から 23 文字を取り出し、残高とする。	
⑳	入出金区分が「+」と一致するかを判断し、	
㉑	正しいならば金額を入金とする。H を実行し、	
㉒	間違いないならば 0 を 0 とし、H を実行し、入金とする。H を実行する。	
㉓	入出金区分が「-」と一致するかを判断し、	
㉔	正しいならば金額を出金とする。H を実行し、	
㉕	間違いないならば 0 を 0 とし、H を実行し、出金とする。H を実行する。	
㉖	所属とマンナンバーと、氏名と日付と入金と出金と残高とを結合し、	
㉗	結果 H を実行して出力する。	
㉘		
㉙	経理台帳のファイルを開始する。	
㉚	台帳処理 H を実行する。	

寄稿

き込み、大用金区分が「***」のときは用金の欄に金額を書き込んで印字するプログラムである。印字のフォーマットはステップ⑧のように、印字すべきデータを出力フォーマットに従って結合することで行われる。

以上が従来用金システムを例にとった日本語 AFL の事務分野への応用例である。これは日本語ドキュメントを兼ねた日本語プログラムとなっており、プログラムの変更、修正、機能追加などの作業が容易である。

ステマ記述

——コンパイラの作成——

日本語 AFL は豊富な記号演算機能を持つ。その機能を利用した例として日本語 AFL プログラム言語を AFL 言語のオブジェクト・プログラムへ変更するコンパイラを取り上げた。コンパイル処理の概要は図 12 で説明したが、図 20 で示すコンパイル・プログラムはこの基本部分を実用し書き替えたものである。

図 20 のステップ①～⑧では日本語プログラムをスクリュー「 \rightarrow 」や「 \leftarrow 」計算するなどの基本となる処理および読取制御を取り出し、それぞれの基本単語に応じた処理を行っている。同じ基本単語が 1 つの文章の中に複数あった場合でも、誤りなく処理が行えるよう、1 つの文章中に存在する基本単語の数を取り出し、その回数だけ同 処理を繰り返している。ステップ⑨は基本制御「 \rightarrow は \leftarrow 」の処理である。「 \rightarrow 」の文字列を「 \leftarrow 」に置き換えている。例えば、

「N は 1。」

というスタートメントは

「N=1」という AFL 言語に変換される。

ステップ⑩、⑪の計算処理は基本制御「計算する」を持つ文章を処理する部分である。例えば、

「体積を計算する。」
という日本語プログラムを

「結果=体積。」

という AFL 言語のオブジェクトに変換される。このオブジェクトが実行されると「体積」という名前でおかれる演算式。例えば、「縦*横*高さ」が実行され、結果が「結果」という名前で作納される。

ステップ⑫～⑭は判断処理のプログラムをコンパイルする部分である。例えば、「A が B より大きい か」判断し、

「正しいなら C を実行し、
間違えたら D を実行する。」

という日本語 AFL プログラムから

「Z1=A, GT, B,

Z11=C,

Z10=D,

<Z1><Z10>。」

という AFL 言語のオブジェクト・プログラムをつくり出す。ステップ⑯はこの判断処理が呼び出れるごとに Z1, Z2, ... と異なる名前が選択文をつくる識別子を求めている。ステップ⑳～㉑では選択文から左右の比較データを左端、右端に取り出す。ステップ㉒で比較条件「大きい」が取り出され、ステップ㉓で、AFL 言語での比較演算子「GT」を含む文章が「条件文」という名前がつくられる。ステップ㉔～㉖で判断結果が正しいときの実行文「Z11=B」がデータの名称でつくられ、同時にステップ㉗～㉙で間違いのときの実行文

Z10=C」がデータの名称でつくられる。ステップ㉚は条件に応じた実行文をつくり出す部分であり、

「<Z1><Z10>。」

というスタートメントがデータの名称でつくられる。最後にステップ㉛で条件文、データ 1、

新製品

ロジックアナライザ GL1600



100MHz・16ch ロジックアナライザ
+20MHz パターンジェネレータ
+デスター機能

ソフト・ハード解析に
対応したパワフルな機能。

- 高速・マルチチャンネル
 - 100MHz クロック
 - 1ch データ + 2ch クリアファイブ
 - 8ch クロック・クリアファイブ
- 総合デバッグに対応したマルチ表示
 - タイミング表示
 - スタート表示 (7進、8進、16進)
 - マップ表示
- パターンジェネレータ標準装備
 - 1024ビット/8ch×16または2048ビット/8ch×8
 - 20MHz
- デスター機能
- 磁気バブルメモリ搭載(オプション)
- 使い易さを上げたソフトウエアキー
- プリンタ出力標準装備

この性能で
¥2,250,000

グローバル電子株式会社
〒100 東京都千代田区千代田 1-10-10
丸の内線丸の内駅 丸の内ビルディング 10F
丸の内線有楽町線有楽町駅 丸の内ビルディング 10F
丸の内線茗荷谷駅 丸の内ビルディング 10F
丸の内線茗荷谷駅 丸の内ビルディング 10F

（各社代理店あり）

データおよび、データを結合し、前述のコシパイルされたオブジェクト・プログラムが「文」という名前でつくられる。

ここで説明した日本語 AFL のコンパイル過程を図 21 のようになる。ある問題が与えられたときに、その問題を解くプログラムを自然に近い表記

の日本語 AFL で記述し、コンパイラにより AFL 言語に変換され、それが実行されるということである。最終的に実行されるオブジェクト・プログラムの形態

図 20 ●日本語 AFL コンパイル・プログラム

ステップ番号	文処理	説明
01	文処理	文を入力し、文とする。
02		N を読み出す。
03		文から [] の数をとり出し、処理を結果に送り出す。
04		文から [] を計算する。この数をとり出し、計算結果を結果に送り出す。
05		文から [] を判断し、この数をとり出し、判断結果を結果に送り出す。
06		文から [] を送り出す。この数をとり出し、結果を結果に送り出す。
07	処理	文の [] を [] に置き換える。
08	計算処理	文から [] を読み出す。 [] と [] を結合し、文とする。
09	判断処理	[] と [] を結合し、識別子とする。 N を読み出す。 文から [] を読み出し、条件文とする。
10		条件文から [] の項目を読み出し、左項とする。
11		条件文から [] の項目を読み出し、右項とする。
12		条件文に [] が含まれるかが判断し、正しいなら大処理を実行する。
13		条件文に [] が含まれるかが判断し、正しいなら小処理を実行する。
14		条件文に [] が含まれるかが判断し、正しいなら大処理を実行し、間違いないなら小処理を実行する。
15		文から [] を読み出し、結果から [] の項目を読み出す。
16		結果が [] と同じか判断し、正しいなら [] を実行する。
17		識別子と [] と結果と [] を結合し、データとする。
18		文から [] を読み出し、結果から [] の項目を読み出す。
19		結果が [] と同じか判断し、正しいなら [] を実行する。
20		識別子と [] と結果と [] を結合し、データとする。
21		[] と識別子と [] と [] と [] を結合し、データとする。
22		条件文とデータ1とデータ2とデータ3を結合し、文とする。
23	大処理	識別子と [] と左項と [] と [] と右項と [] を結合し、条件文とする。
24	小処理	識別子と [] と [] と左項と [] と右項と [] を結合し、条件文とする。
25	含処理	識別子と [] と [] と右項と [] と [] と [] を結合し、条件文とする。

寄稿

がAFL言語という高水準なプログラム言語であり、コンパイラの記述が日本語AFLという記号演算機能の豊富な言語を使用しているため、図20のプログラム例で示したように、コンパイラは簡潔なものとなっている。

以上2つの応用例で、ファイルの読み書き、出力データの編集、文字列の操作といった実際の業務で頻繁に使われる機能が日本語AFLで自然な日本語の表記で簡単に記述できることを示した。日本語AFLではこのように従来のプログラム言語ではわかりにくい記述となる処理がわかりやすい記述となる。このため、マン・マシン・インタフェースやデータベース、さらに医療をはじめとする各種のエキスパート・システムなど、発達が期待される分野への応用が考えられる。

日本語AFLの今後の展開

これまで述べてきたこととわかるように、日本語AFLは日本人にとってかなり自然言語に近い表記の日本語プログラ

ムになっている。英語中心のプログラム言語と比較すると、わかりやすく、第三者でもプログラムを見て内容を理解することができ、また、初心者になじみやすくなっている。

日本語AFLは標準的なパーソナル・コンピュータ上で稼働するシステムであるので、これからコンピュータを学習し、プログラミングを学ぼうとしている人に適したプログラミング言語と思われる。このような日本語プログラミング言語によって、コンピュータ人口が広がってゆくと期待される。

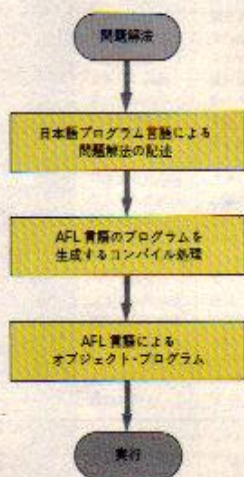
情報化社会では今後、ますます、コンピュータによる大量のデータ処理が必要となってくるが、最近ではパーソナル・コンピュータといえどもディスク装置を接続し、高度な処理を行うようになってきている。このようなシステムは内部の核となる高速処理のハードウェア部分と、高速処理の操作を容易に行うためのマン・マシン・インタフェースのソフトウェア部分で、構成されている。

したがって日本語AFLをマン・マシン・インタフェースの部分に位置させると、だれもが内部の高速処理と連結した高性能なパソコンを容易に使いこなすことができるようになる。半導体を中心とするハードウェア技術の進歩により、日本語AFLの動作環境は着実に広がって来たといえよう。

引用参考文献

- 1) 上村：巻頭く日本語プログラミング時代、日経コンピュータ、no.14、pp.40-53、1982年4月5日号。
- 2) 神田、杉本、武井：エンドユーザのための日本語によるプログラミング。情報処理、vol.21、no.3、pp.208-213、1980。
- 3) 上田、青野、野田：高水準言語AFLとその処理システム、日経エレクトロニクス、1977年10月3日号、pp.94-117。

図21 ●日本語AFLのコンパイル実行過程



本格的なシステム時代へ

大型コンピュータのDBMSのスペシャリスト、基本ソフトウェアのシステムプログラマ及び上級システムエンジニアのブレイクが、MDS 2886HDを駆使して、いま新しいシステムの時代を拓きます。

KOZU*SD
マイクロプロセッサラボラトリ

が拓きます。



MDS 2886HD

標準構成
CPU: 8086-10
128KRAM
フロッピーディスク
(2HD)
ハードディスク
(5.25/10.4)
5.25/10.4
UNIX (5.4)
10/15/20
標準プリンタ

標準構成

●マイクロプロセッサ用基本ソフトウェアの開発 (DOS/UNIX/DBMS/ファイル・システム/ネットワーク) ●マイクロプロセッサ用システムの開発 (オフライン、アプリケーション/パッケージ、ソフトウェア開発環境/ツール等) ●マイクロプロセッサ駆動システムの開発 (ハードウェア、水中設置装置等)

株式会社 神津システム設計事務所
東京都港区新橋1-2-3-705 千180
電話 03(232)1350

共に、未来を拓く技術者を募集しております。
(詳細はお問い合わせ下さい。)

〈資料請求番号129〉

プログラム	::= 宣言文
宣言文	::= 文 文
文	::= 節 節 節終端
節	::= 目的節 空白 物節 空白 (修飾節) 空白
修飾節	::= 目的節 (目的節) (節終端) (注脚)
目的節	::= 名節 量修飾節
空白	::= 空白
名節	::= 直接名 (間接名)
直接名	::= 英数字 日本語 (英数字 日本語)
間接名	::= (名詞性) (動詞性) (修飾的)
量修飾節	::= (数) (数) (文字) (文字) (文字) (文字)
英数字 日本語	::= 英数字 日本語
文字	::= 英数字 日本語 (記号)
日本語	::= カタカナ (カタカナ) (漢字)

項目	小項目	基本形表現
代入文		A ₁ はB ₁ である。 A ₂ をB ₂ にする。 A ₃ はB ₃ 。
選択文		A ₁ がB ₁ より大きいと判断し、正しいならばP ₁ を実行し、間違っているとQ ₁ を実行する。 A ₂ がB ₂ より小さいと判断し、正しいならばP ₂ を実行し、間違っているとQ ₂ を実行する。 A ₃ がB ₃ より大きいと判断し、正しいならばP ₃ を実行する。 A ₄ がB ₄ より小さいと判断し、間違っているとQ ₄ を実行する。
繰り返し文		A ₁ をN ₁ 回繰り返す。 A ₂ をB ₂ がN ₂ と等しいまで繰り返す。
実行文	基本実行文	A ₁ を実行する。体積 ₁ を計算する。 [A ₂ ×C ₂ +D ₂]を実行する。 B×C+D ₃ を計算する。
	算術比較文	A ₁ がB ₁ より大きいと判断する。 A ₂ がB ₂ と等しいと判断する。 A ₃ がB ₃ より小さいと判断する。
	文字比較文	A ₁ がB ₁ と一致すると判断する。 A ₂ はB ₂ に含まれるかと判断する。
	ファイル文	A ₁ のファイルからB ₁ の行を読み取る。 A ₂ のファイルにB ₂ を書き、A ₃ のファイルを開閉(終了)する。 A ₄ のファイルを作成(書き)する。
	入出力文	1文字 ₁ を入力する。 1行 ₁ を入力する。 A ₂ を出力する。 A ₃ を印刷する。 A ₄ を表示する。
制御文	メモリー文	A ₁ の番地 ₁ からB ₁ 文字を読み取る。 A ₂ の番地 ₂ にB ₂ を書き、
	基本制御文	A ₁ をB ₁ で割る。 A ₂ をB ₂ で割る。システム ₁ へ戻る。 B ₃ の演算精度で制御する。 B ₄ の行区切り記号で制御する。

項目	小項目	基本形態
実行文	算術演算文	AにBを加える、AからBを引く、AにBを掛ける、 AをBで割る、Aの10進数(16進数)を18進数(10進数)に変換する、 Aの2進数(16進数)を18進数(2進数)に変換する。
	記号演算文	AとBを結合する、AのBを挿入する、(AからBを削除する、 Aの右からBの左端をとり出す、Aの右からBの左端をとり出す、 Aの右からBの左端をとり出す、Aの左からBの右端をとり出す、 AからBの左端をとり出す、 AのB番目をとり出す、AのB番項目をとり出す、 AのB行目をとり出す、 Aの文字数をとり出す、Aの行数をとり出す、Aの行数をとり出す、 AからBの行数をとり出す、AをB回取り出す、 AからBで取り出す、AをB回取り出す。
	記号演算文	AのB番目からC文字をとり出す、 AのB番項目からC番項目をとり出す、 AのB行目からC行目をとり出す、 AのB番目をCに入れ、AのB番項目にCを入れる、 AのB行目をCに入れる、 AのBをCで置き換える、 AのB番目からC文字をDで置き換える、 AのB番項目からC番項目をDで置き換える、 AのB行目からC行目をDで置き換える。
演算	代入演算	である(であり)とする(とし)
	実行演算	実行する(実行し)、計算する(計算し)
	比較演算	相等する(相等し)
	繰り返し演算	繰り返す(繰り返し)
	算術演算	加える(加え)引く(引く)、掛ける(掛け)、割る(割り)、変換する(変換し)
	ファイル演算	読む(読み)書く(書き)開始する(開始し)終了する(終了し)作成する(作成し)消去する(消去し)
	入出力演算	入力する(入力し)出力する(出力し)印字する(印字し)表示する(表示し)
	メモリー演算	読む(読み)書く(書き)
	基本制御演算	動かす(動かす)制御する(制御し)戻る(戻り)
	記号処理演算	結合する(結合し)挿入する(挿入し)削除する(削除し)取り出す(取り出し) 置き換える(置き換え)置換する(置換し)入れる(入れ)
制御部	算術比較演算	大きい/小さい/等しい
	文字比較演算	一致する/含まれる/空
	選択制御部	正しい/間違ひ
	算術演算部	10進数、16進数、2進数
	ファイル演算部	ファイル、行
	入出力演算部	文字、行
	メモリー演算部	単位
	基本制御演算部	システム、演算モード、行外制御部
記号処理演算部	右から、左から、前、後、長さ、番目、項目、回、文字数、項数、行数、行目、文字、行、空	
演算	基本演算	は、を、から、に、の、で、
	制御演算	まで、が、より、なら、へ、
記号	文記号	、() / () () () ()
	算術記号	+ / * / ^ / () / ()

MC

