

K. E. Iverson の A P L について*

清水川 緋紗子

1. 始めに

A P L (A Programming Language) は、最初、K. E. Iverson が定義し (Wiley, 1962) それ以来彼が A. D. Falkoff と共に改良を続けている汎用プログラミング言語であって、現在多くの機械に使われている。これは記述用言語として優れ、更に設計用言語としても役立つ⁽¹⁾。また教育用言語として、初歩の数学の訓練⁽³⁾、A P L 文法の訓練等⁽⁴⁾に役立っている。これは統計計算用言語としても有用である。従って、yale 大学の統計学部から“A P L の統計計算への使用について”という技術報告が出されているし、カナダの Alberta 大学の計算機科学科では A P L の統計計算用パッケージが作られ利用されている。K. E. Iverson は、“統計での A P L 使用について”という⁽⁶⁾ペーパーを出版している。以下それを要約し更に A P L の特徴を若干述べる。

2. Iverson があげた A P L 3 つの特徴

- i 演算素子は広く役立つものでありその種類が豊富である。
- ii 関数の定義でその引数が他の関数であることを許す。
- iii 配列を直接とり扱える。

Iverson は、i, ii, iii の特徴をあげて A P L が Special Purpose Language を定義するのに大変役立つ言語であると述べている。また、多項式、正規方程式、行列の逆転等の数値計算は無論、位置を示す平均値のメディアン、モ

* 原稿受領 1971年6月11日

ード等も APL で表現できることを示した。

Iverson の ペーパーは、APL の 簡潔な解説であり、統計計算のみならず、非数値計算用言語として APL が役立つことを示した。従って我々は APL が非数値計算用言語としても表現力に富んだ言語であることを知る。

また A. Carraciolo de Forino は、APL が配列やファイルや木構造を含めた高度の構造をもつデータを扱うのに便利であり、大変綿密で総合的な記法システムにもとづいていると述べている。⁽⁸⁾

筆者は APL が会話型言語としても優れた言語であると思うので以下その点を指摘したい。

3. 会話型言語としての APL

現在、APL は、ユーザーが端末装置の鍵盤から命令やデータを打ちこみ、中央のコンピューターや他の端末装置と情報を交換してコンピューターを使用するのに用いられている言語であって、電話回線によるコンピューターの有料使用が可能になった我国でも APL 的言語の重要性は増しているであろう。

まず APL の式の計算順序は、多くのプログラム言語とは異り右から左へ行われるが、右から左へを採用した理由は4つあげられており、その1つは、右から左へ計算される式は左から右へ読むのに楽だということである。⁽³⁾

次にこれは、ユーザー自身がタイプをするのに効率的であり、タッチも少なくて済む言語である。

ユーザー自身がタイプをするので次の2つの特色がある。

- i だれもがタイプし易いよう予め鍵盤が配列されている。即ち特殊文字ができるだけ記憶に便利なよう工夫されていて、ギリシャ文字は対応するローマ文字の上へ、意味の連想がつくものはその文字の上におかれ、関係記号も順序よく並んでいる。例えば () は [] の上へ ? は Q の上に置かれている。また、 π 倍を表す \circ は \bar{O} の上へ並んでいるが、これと同じことを電々公社の電話計算システムでは XF80 とボタンを押

さねばならない。⁽²⁾

- ii 記号が極めて簡単化された。即ち，上下，左右の回転で意味が異なるため，他人を介するシステムでは採用できない記号の使用や二重打ちによる記号の増加が可能である。

例えば $_ _ _ | / \backslash$ や $\Phi \Omega$ や $\Delta \Psi$ がある。

タイプのタッチが少なくなるように APL は次のような特色をもつ。

- i 豊富な記号がある。ALGOL 等の記号の他に

イ．常数倍が 1 記号で表わされている。

○ と * が π, e 倍を表す。

ロ．頻度の多い関数が 1, 2 の記号で表わされている。

「 L * ⊕ | ! ○ 等。

ハ．記号処理に便利な記号がある。

! ρ, ∪ Δ Ψ / \ Φ Ω ε ⊥ T 等。

- ii □ スペースを打つことに意味がある。

□ は関数名と引数との区切り，配列の次元の区切り，データの区切り等に用いられており，FORTRAN が H field 以外の □ を無視するのと全く異って，無駄な □ を打たない。

- iii 関数サブプログラムの定義が簡単である。

一般に $\nabla \dots \nabla$ は，関数サブプログラムの宣言と END を書いたのと同じである。

例えば幾何平均は次のように定義される。

$\nabla Z \leftarrow \text{GMEAN } V$

$Z \leftarrow (\times / V) * \div \rho V \nabla$

但し V はその成分が正のベクトルを表す。ρ V はベクトルの成分の数である。× / V は $\prod_{i=1}^{\rho V} x_i$ である。V = (x₁, x₂ ... x_n)

- iv ループ記法の簡単化。

行列の内積，外積の定義や低減の定義とか行列の要素毎に演算する記法により，通常の代数的行列演算が簡単に行われて，プログラミングの

ループ記法は全てそれらで行われる。即ち FÖRTRAN の DÖ 記法を追放できる。

例えば $+ / V$ は $\sum_{i=1}^{pV} x_i$ に同じ。

v 条件文の簡単化。

ALGÖL の条件文は算術式で表わしている。⁽⁷⁾

4. 最 後 に

以上簡単に APL の使われ方についてのみ述べた。APL の内部等については、またの機会にする。

- (1) 萩原・黒住, 「計算機設計言語」『情報処理』(Feb., 1971) p. 93.
- (2) 広田, 「電話計算システム」『情報処理』(Mar., 1971) p. 129.
- (3) 和田 弘 訳, 「アルゴリズムによる初等関数」(1969) p. 194.
- (4) *APL\360 User's Manual* (IBM Corporation, Sep. 1969).
- (5) *APL\360 Primer* (IBM corporation, Sep. 1969).
- (6) Roy. C. Milton and John A. Nelder (ed.), *Statistical Computation* (1969).
- (7) S. Rosen (ed.), *Programming systems and Languages*, (1967) p. 167.
- (8) Julius T. Tou (ed.), *Advances in Information Systems Science*, (1969) p. 91.