

FORTRAN IV 言語の概略*

西村 恕彦**

1. まえがき

FORTRAN (Formula Translator) は元来 IBM 704 のために作られた自動コーディング方式である⁶⁾。その後、共通言語の性質を備えていることが見出され、IBM 社のすべての計算機について用意されるようになった^{7,8)}。

FORTRAN 言語は 4 種に大別され、そのうち 2 種だけが共通言語として用いられている。そしてこれらは非公式にはあるが、I, II, III, IV と区別されることがある。

FORTRAN I は、Basic FORTRAN と呼ばれることもあり、カードシステムの計算機で普通に用いられる⁷⁾。FORTRAN II は、Full FORTRAN などとも呼ばれ、SUBROUTINE, FUNCTION, COMMON, CALL, RETURN の五つの命令 (いわゆるサブプログラムのための命令) を追加したものであって、テープシステムの計算機で翻訳される^{4,9)}。FORTRAN III は、2 進法の機種にのみあり、FORTRAN II を少し拡張したものである。FORTRAN IV は現在のところ、IBM 7030, 7090/94, UNIVAC III についてのみ計画されている¹⁰⁾。翻訳ルーチンの完成予定は 1962 年前半といわれ、言語の仕様の概略はすでに発表されている¹⁾。FORTRAN IV は I および II の命令をすべて含んでいるが、多少、質的な変更もあって、さらに柔軟かつ強力な書き方ができるようになった。

すなわち形式的には、データの型の宣言と、条件文とが追加され、入出力命令が整理された。内容的には、2³⁵ までの整数演算、2³⁴ の精度の倍精度演算、複素数演算などが含まれる。また IOCS を内部的に採用して、全体の時間を短縮している (翻訳時で約 2 倍の速さが見込まれている)。

2. 用語の説明および制限事項

この解説では FORTRAN IV に関して次のような用語を使う。ただし、これは全く便宜的なものであり、

FORTRAN 言語自体は、このような定義の仕方をしていないし、またそれを提供することが、この解説の目的でもない。したがって各項の詳細なことは参考文献によらる^{1,8,9)}。

型 (type): データが扱われる形式。次の六つの型がある。整数 | 実数 | 倍精度数 | 複素数 | 論理数 | 文字。説明は定数の項を見よ。

データ: 定数 | 変数 | 関数。

定数 (constant): 六つの型がある。

整数: 通常の整数。2³⁵ まで扱う。

実数: 小数点を含む数。精度は 2²⁷ まで。

倍精度数: 実数の定数のあとに D をつける。精度は 2³⁴ まで。

複素数: (実数の定数, 実数の定数)

論理数: .TRUE. | .FALSE.

文字: 字数 H 任意の文字。字数は 120 字まで。

記号: 英字で始まる、1~6 字の英数字。変数 (variable), 関数 (FUNCTION), サブルーチン (SUBROUTINE), 入出力装置 (unit), 変数のブロックなどの名称として用いる。

リスト (list): 幾つかのデータをコンマでつないだもの。

算術式 (arithmetic expression):

1) データを演算子 (+|-|*|/|**) およびカッコで組み合わせ、最終的に一つの算術的な値に還元できるようにしたもの。

2) 原則として一つの式に型の異なるデータを混ぜてはいけない。ただし、倍精度の式に実数を、複素数の式に実数を、それぞれ混ぜることができる。

論理式 (logical expression): 論理データ | 二つの算術式を比較関係でつないだもの | 論理式を演算子 (.OR. | .AND. | .NOT.) およびカッコで組合わせたもの。なお論理演算子として +|*|- は使えないようである。

比較関係: .LT. | .LE. | .EQ. | .GE. | .GT. | .NE.

(注. <|≤|=|≥|>|≠)

命令番号 (statement number): 5 桁以内の整数

* An Outline of FORTRAN IV Language, by Hirohiko Nisimura (IBM Japan, Ltd.)

** 日本 IBM 社教育部

数、上限が 2^{15} であるか 10^5 であるか不明。

3. データの記述

この節で説明する命令 (statement) はいずれも、計算としては実行されず、データの性質を記述するだけである。原則としてプログラムの最初にまとめて書く。

3.1 型の宣言

次に述べる五つの命令がある。一つの記号は、ただ一つの型にしか属することができない。

INTEGER 記号のリスト

REAL 記号のリスト

DOUBLE PRECISION 記号のリスト

COMPLEX 記号のリスト

LOGICAL 記号のリスト

これらの命令はリスト中の記号がそれぞれ、整数、実数 (浮動小数点数)、倍精度数、複素数、論理数の値をとる変数または関数であることを指定している。変数名あるいは関数名として用いられる記号は、必ずその属する型を明示せねばならない。宣言されなかった記号は、従来の規則によって、INTEGER または REAL とみなされる。

文字データについては、特別な指定はない。

3.2 その他の宣言

EXTERNAL 記号のリスト

記号が関数名 (FUNCTION) またはサブルーチン名 (SUBROUTINE) として用いられることを示す。しかもこの記号は他の関数、サブルーチンの呼出しパラメータとしても使うことができる。

DIMENSION 変数 (正整数のリスト), ……

変数が添字つき変数として用いられることを示す。3次元までの添字がつけられる。カッコのなかは定数であって、添字の上限を示す。

DATA 変数 (定数), ……

定数域 (CONSTANT SECTION) を設定する。定数は文字、あるいは8進数 (Oで書き始める) でもかまわない。変数に対してその定数の値が予め与えられる。論理データは扱えない。

BLOCK DATA

はっきりしたことは不明であるが、END 命令によって cancel されない定数域を設定するようである²⁾。

COMMON /ブロック名/変数のリスト……

END 命令によって cancel されない記憶場所をブロックごとに設定する。ブロック名は blank であり

得る¹⁾。

EQUIVALENCE (変数のリスト), ……

同一の記憶場所を幾つかの異った変数名で取扱うことができる。

FORMAT (……………)

変数の入出力型式を指定する。整数 (I), 浮動小数点数 (E|F), 文字 (A), FILLER (X), 8進数 (O) を入出力することができる。

論理データは扱わない。

命令番号を必ずつける。

4. 演算命令

変数 = 算術式 が一般形である。次の四通りの書き方ができる。

$\left\{ \begin{array}{l} \text{整数} \\ \text{実数} \end{array} \right\}$ の変数 = $\left\{ \begin{array}{l} \text{整数} \\ \text{実数} \end{array} \right\}$ の算術式

倍精度の変数 = 倍精度の算術式

複素数の変数 = 複素数の算術式

変数 = 文字の変数

また下記の書き方もできるらしい。

論理変数 = 論理式

FORTRAN III の Bool 演算 (マスキングなど) は含まれない。

5. 制御命令

次の三つ以外の制御命令は従来どおりである³⁾。

STOP

プログラム実行上の終点であり、モニタを呼び出す²⁾(従来の CALL EXIT)。

IF (算術式) 命令番号 1, 命令番号 2, 命令番号 3
算術式の値の負, 零, 正に従って, 1, 2, 3, の行先へ分岐する。算術式の型は、整数、実数、倍精度のいずれかに限る。

IF (論理式) 命令 1

論理式が成立 (.TRUE.) ならば命令 1 を実行する。論理式が不成立 (.FALSE.) ならば次の命令を実行する。命令 1 は実行できる (executable) 命令であり、しかも DO または IF (論理式) であってはならない。

論理式に含まれる算術式は、整数、実数、倍精度、複素数のいずれであってもよい。(ただし、異った型の式の比較関係、複素数の大小の定義などについては不明)。

文字データの比較については不明。

6. 入出力命令

入出力命令として、次の四つがある。もちろん、従米どおりの書き方をしてもかまわない。

READ (記号, 指定) 変数のリスト

WRITE (記号, 指定) 変数のリスト

通常の入出力。記号は入出力装置の名称、指定は FORMAT 命令の命令番号、または 1 次元の添字を宣言された文字の変数。後者の場合、プログラム実行時に FORMAT の指定を読みこむ。変数のリストについては、FORMAT 命令の項を参照せよ。

READ (記号) 変数のリスト

WRITE (記号) 変数のリスト

外部記憶へ中間結果を一時的に、書き出し、また読みこむ。

7. むすび

FORTRAN IV の言語は、単に従来の FORTRAN 命令のレパートリを拡張したものとのみはいきれない。幾つかの面で、思想の変化を伴っているようである。それは端的にいえば、ALGOL, COBOL といった近代的な言語の影響であろう。にもかかわらず FORTRAN IV の言語に含まれていない feature を、COBOL からひろってみれば、次のようなものがある。

データの階層的構造 (ファイル, レコード, 集団項目, 基本項目)/自動小数点演算/文字データの比較/入力 End of File/自由な編集。

参考文献

- 1) "Advance Specifications: 7090 FORTRAN and FORTRAN Assembly Program (FAP)", IBM Bul. J 28-6132, (1961-7), p. 20
- 2) "7090 FORTRAN", IBM Tech. Newsletter N 28-0001, (1962-1-2), p. 2
- 3) W.P. Heising and R.A. Larnar: "A Semi-Automatic Storage Allocation System at Loading Time", Com. ACM. Vol. 4, No. 10, (1961-10), pp. 446-449
- 4) "Comparison of All FORTRAN System" IBM Systems Letter, Technical Bulletin No. 5, (1961-10-17), p. 7
- 5) "32 K 709/7090 FORTRAN: Double Precision and Complex Arithmetic", IBM Bul. J 28-6114, (1961-4) p. 8
- 6) "Programmer's Primer for FORTRAN Automatic Coding System for the IBM 704 Data Processing System", IBM G.I.M. F 28-6019/32-0306-1, (1959-10-1), p. 64
- 7) D.D. McCracken "A Guide to FORTRAN Programming", Wiley, (1961), p. 88
- 8) "FORTRAN", IBM G. I. M. F 28-8074-1, (1961-12), p. 104
- 9) "709/7090 FORTRAN Programming System", IBM R.M. C 28-6054-2, (1961-1), p. 112
- 10) "UNIVAC III FORTRAN PROGRAMMER'S GUIDE", UNIVAC Bul. U-3517, (1962-6-1), p. 112

(昭和 36 年 6 月 11 日受付)