

ミニ FORTRAN の提案*

大 駒 誠 一**

JIS FORTRAN¹⁾が制定されて2年以上たち FORTRAN コンパイラの機能の比較や用語の統一などに大いに役に立っている。ところが、いまある3つの水準のうち、一番低い水準 3000 の FORTRAN でさえも、最近はやりの超小型計算機のプログラム言語として、あるいはコンパイラとはどんなものか、ためしに作ってみようなどという場合の言語としては機能が大きすぎる。JIS は制定されて3年たつと改訂できるようなので、言語仕様をできるだけ制限して馴れた人なら1人で1箇月くらいでコンパイラが作れそうな、ミニ FORTRAN (JIS 1000?) というのを提案したい。

このミニ FORTRAN では、水準3000の FORTRAN から他の文で同様のことができるものや、あまり使わないのにコンパイルが面倒な機能を削除し、ちょっとしたこと、コンパイルがずっと簡単になるような制限を加えたものである。

この提案は3種類に分類してある。

A は現在の JIS FORTRAN に矛盾しないで、コンパイラ作製がかなり簡単になり、しかも、プログラムを組むにあたって、それほどきびしい制約とはならないような提案である。ここで矛盾しないという意味は、このミニ FORTRAN で書いたプログラムは、JIS FORTRAN のコンパイラで完全にコンパイルでき、結果も同一となることが期待できるということである。

B は JIS には矛盾しないが、ここまで機能を削ってしまうと、FORTRAN とはいがたいといった感じのするものである。しかし、このくらい思いきって機能を縮小した方が、かえって実用上は効率がよいかも知れない。

C は JIS には矛盾するが、ミニ FORTRAN ではぜひ実現したい変更である。

A-1 空白を区切り記号とする

名前や定数の中に空白を入れてはいけないことにする。したがって、DO 文や GO TO 文と代入文との

区別が簡単につく。

区別がつけにくい例:

```
DO 15 K=L(I, 3)
```

```
DO 2 I=1, 20
```

```
GOTO(5, 20), JACK
```

```
GOTO(3, 1)=JACK
```

A-2 DO, CALL, STOP, IF などを予約語とする

READ や GO を変数名や配列名として使ってもかまわないために、FORTRAN コンパイラは文の型を決定するのにかなり苦労している。

例: READ(L)=L*(L+6)+5

```
IF(I)=3*(I+K)
```

これは IF や READ が配列名なら代入文で、配列名でなければ文関数定義文である。組込み関数名と基本外部関数名も予約語とすれば、この両者に区別をつける必要がなくなりすっきりする。

A-3 DO 文は必ず対応の CONTINUE 文を必要とする

DO の文末文にはいろいろうるさい制限があり、DO の末端が何重にも重なるとかかなりやっかいである。

A-4 EQUIVALENCE 文の削除

EQUIVALENCE 文が、COMMON 文や DIMENSION 文と関係するとその処理が非常にむずかしい。

A-5 添字は整数型変数と整数のみとする

このような制限をつけると、コンパイラが簡単になるばかりでなく、添字に複雑な式を許すよりも、実行時間が早くなることが多い。

A-6 実定数は小数点の両側に数字を必要とする*

小数点は代入文や FORMAT 文などにおいて、数

* Proposals on Mini FORTRAN, by Seiichi Ohkoma (The Faculty of Engineering, Keio University)

** 慶応義塾大学・工学部

* 和田英一氏(東京大学)のシラブルリーダーはこうなっている⁽²⁾。

字と数字との間だけにしか表われない。

例: 0. や .314 E + 1 のような実定数は使えない。

A-7 入出力並びに配列名の使用禁止

配列名に添字をつけないで使えるのは、COMMON 文の中と副プログラムの引数のみとする。入出力文では、配列名 (entire array) の代わりに、DO 形並びを使う。

A-8 DO 文の増分パラメタを必ず指定する

入出力文の DO 形並びも含めて、DO 文の書き方は一とおりにする。

例: DO 100 I=1, 100
を DO 100 I=1, 100, 1

と書くのは苦痛ではない。

B-1 文関数の削除

文関数は文が一つだけの関数副プログラムと考えられるから、文関数は関数副プログラムを使うことにする。

B-2 計算型 GO TO の削除

算術 IF 文で代用が可能である。

B-3 書式なし入出力文の廃止

書式つき入出力文で代用する。実行速度は犠牲になるが。

B-4 COMMON 文または引数の廃止

現在の FORTRAN では主プログラムと副プログラムとの間のパラメタの受け渡しは、COMMON 文で行なう方法と、副プログラム呼び出しのときの引数による 2つの方法があるが、これをどちらか一方だけに制限しようというわけである。仮引数が実行文中に入っていると処理が面倒なので、引数の使用を禁止したいが、禁止すると関数の引用は引数を 1 個以上必要とするという JIS の文法に違反することになるので、問題であるが。

B-5 変数名と配列名の区別

水準 3000 では I, J, K, L, M, N で始まる英字名は整数型、それ以外の英字で始まる名前は実数型であるが、たとえば、N と P で始まる英字名はそれぞれ整数型と実数型の配列名だけに使うものとし、変数名

にはこの 2 字で始まる名前は使えないものとする。

コンパイラは名前の先頭の 1 字を見ただけで、変数名か配列名か区別がつけば、文法解析がかなり楽になる。

B-6 単項作用素の禁止

単項作用素 (unary operator) は使用頻度が少ない割に文法の検査がむずかしい。どうしても単項作用素を使いたいときには、つぎのようにする。

例: A=0. 0-B

C-1 STOP, PAUSE 文の変更

いまの JIS FORTRAN では、STOP または、PAUSE の後は「空」か、「8 進数字の列」がくることになっているが、FORTRAN の中で、この 2 つの文だけに 8 進数が現われるのはおかしい。どうしてこのような盲腸みたいなものを、いまだに ISO や JIS の FORTRAN に残しておくのか理解に苦しむ。ここは当然「空」または「整定数」とすべきである。

C-2 入力の際の文字欄記述子

FORMAT 文中の文字欄記述子 (nH 形) は、入力の際書式仕様自身のところへ読み込ませることになっているが、このようにプログラム自身を自分のプログラムで実行中に書き替えるような高級な機能は、小さな FORTRAN には不要である。動的部分のほとんどないといっている FORTRAN にこのような機能が入っているのは奇妙である。FORTRAN は ALGOL と違って、自然発生的にできてきた言語なので、設計思想が不統一なためかも知れない。

このようにプログラムで自分のプログラムを書き替えるような機能は非常に面白く、かつ有効に使えられるので、もっと水準の高い言語仕様にはどんどん取り入れることにして、小さな FORTRAN では入力の際の nH 型記述子は空白欄記述子と同じとするか、または未定義で十分である。

参考文献

- 1) 日本工業規格、電子計算機プログラム用言語、FORTRAN(水準 3000)、JIS C 6203-1967.
- 2) 和田英一: 電子計算機(V), 問題向き言語とコンパイラ, 電子通信学会誌, Vol. 52, 7, 1969 年 7 月.

(昭和 44 年 9 月 10 日受付)