

C5 フォートラン・テスト・プログラムについて

恒川純吉・森美千穂（日本科学技術研修所）

高橋義造・三好 彰（東京芝浦電気）

1. はじめに

一般的なプログラム評価の一環として、FORTRAN処理プログラムのテストを取上げてみた。FORTRANの場合、計算機用の言語としてはもっとも広く使用され、かつ統一化されている。とくにJIS規格が制定されて以来、国産の電子計算機の場合、中型以上の機種ではだいたいJIS規格水準7000のFORTRAN処理プログラムを用意している。外国機種といえども、多少の仕様の拡張があってもだいたいJIS規格水準7000の規格をみたしている。そこでJIS規格水準7000の文法に準じて文法をテストするプログラムを作成し、いくつかの機種についてテストしてみたので、その結果を報告する。

2. プログラム作成の方針

- 1) JIS規格水準7000のFORTRAN処理プログラムをテストの対象とする。
- 2) テストの目的は、文法どおり正しく書いたプログラムが、正確に処理されるかどうかを確認することにある。
- 3) 対象とするFORTRAN処理プログラムはいちおう完成しているものとする。（開発途中の処理プログラムを段階的にテストするものではない）
- 4) 文法上処理系にまかされている部分については、容易にプログラムの変換ができるようにする。
- 5) 処理プログラムにおける誤りに対しては、テストプログラム内でエラー・メッセージを打たせる。
- 6) プログラムはすべてFORTRAN言語で書く。

3. プログラムの内容

でき上がったプログラムは29本で、カード枚数が約15000枚（内約2000枚がデータ）である。処理時間は15分から3時間ぐらいまでであった。

29本のプログラムの内容は表1にまとめて記しておくが、だいたいJIS規格に記された順番に文ごとく調べて行くことになる。文法的に関連している事項のテストはまとめていく。とくに関連のない文の組合わせに対するテストは行っていない。そのようなテストは実際のプログラムでテストするよりないであろう。プログラム作成時に考慮した点のいくつかを記しておく。

表1 FORTRANテストプログラムの内容

(最初の番号はプログラム番号, 題名のあとの番号はJIS規格中のセクション番号を示す)

01 GENERAL TEST

JIS 7000 FORTRANに含まれる文を一通りテストする。

02 プログラムの形式 3.

プログラムの形式をテストするもので, FORTRAN用の文字, 空白の使用, 行, 文の番号について調べる。

03 英字名 3.5

FORTRANの reserved words が変数名, 配列名, 関数名として使えるかどうか調べる。

04 定数 1 5.1.1, 7.5.1

実数型, 倍精度実数型について, 代入文, データ文, READ文によって, 同じ形式の数値を変数に与え, この3つが等しくなることを調べる。

05 定数 2 5.1.1, 7.5.1

実数型, 倍精度実数型, 整数型, 論理型, 複素数型, 文字型について代入文, データ文, 入力によって, 同じ形式の数値(符号つき)を変数に与え, この3つが等しくなることを調べる。

06 添字式 5.1.6

1次元, 2次元, 3次元の配列についての添字式をテストするもので, ある配列要素を異なる2つの添字式であらわし, これらが等しくなることを調べる。

07 算術式(実数型, 倍精度実数型) 6.1

実数型, 倍精度実数型の算術式をテストするもので, 算術式を実数型および倍精度実数型で計算し, 2つの値を比較する。

08 算術式(整数型) 6.1

整数型の算術式をテストするもので; 算術式を整数型および実数型で計算し, 2つの値を比較する。

09 算術式(複素数型) 6.1

複素数型の算術式をテストするもので, 複素数型の算術要素を含む算術式を複素数型の変数をもちいて計算したものと, 実数部分, 虚数部分にわけて実数型で計算したものとを比較する。

10 算術式 (実数型) 6.1

実数型の算術式をテストするもので、実数型、倍精度実数型の算術要素を含む算術式を、実数型、倍精度実数型で計算したものと、倍精度実数型で計算したものとを比較する。

11 関係式 6.2

2つの算術式を異なるいくつかの関係演算子でつなぎ計算結果を調べる。

12 論理式 6.3

いろいろな論理式について計算結果を調べる。

13 制御文 7.1.2

GO TO文、IF文、CALL文、RETURN文、DO文をテストするもので、JIS規格7.1.2の中でSTOP文とPAUSE文はテストから除外している。GO TO文、IF文では正しい位置にコントロールが移るかどうかを調べ、さらに論理IF文では真のときに実行する文の種類を変えてテストする。CALL文、RETURN文については、結果を1として、もとのプログラムへもどってくることを調べる。

DO文については、文末の種類、パラメータの選択、入れ子をなしたDO文、DO文からの脱出、DOの範囲の拡張について調べる。

14 入出力並び1 7.1.3

実数型、倍精度実数型、複素数型の変数および配列について入出力並びをテストするもので、単純並び、DO形並びを調べる。

15 入出力並び2 7.1.3

整数型、論理型の変数および配列について入出力並びをテストするもので、単純並び、DO形並びを調べる。

16 入出力文 7.1.4

READ文、WRITE文、REWIND文、BACK SPACE文、ENDFILE文をテストするもので、磁気テープへ、バイナリー、BCDモードで出力した後に、ふたたびそれを入力して、正しく入出力することを調べる。

17 行送り、ページがえ 7.1.5, 7.5.4

／および書式づけられた記録の第1字目による行送りをテストする。

18 配列宣言子、添字の値、整合寸法 7.2, 7.3.1

DIMENSION文、型宣言文、DATA文 7.3.5, 7.4

配列宣言子、添字の値、DIMENSION文については、1次元の配列を2次元、3次元の配列とEQUIVALENCE文でむすび、添字の値を計算し、それぞれの

配列要素の値が等しくなることを調べる。整合寸法については、整合配列を宣言し、正しく実引数と結合されることを調べる。型宣言文については仮引数を型宣言し、対応する欄記述子で印刷する。DATA文については、同じ数値をDATA文と代入文とで代入し、それらが等しくなることを調べる。

- 19 **COMMON** 文, **EQUIVALENCE** 文 7.3.2, 7.3.3
COMMON 文については、**COMMON** 文で宣言された変数、配列に正しく数値が代入されるか、又共通ブロック間の対応が正しいか、調べる。
EQUIVALENCE 文については、**EQUIVALENCE** 文で宣言された変数、配列が正しい記憶場所に割り当てられるか調べる。
- 20 **EXTERNAL** 文 7.3.4
基本外部関数名、関数副プログラム名、サブルーチン副プログラム名を**EXTERNAL** 文で宣言して、副プログラムの引数として間接的に引用した場合と直接引用した場合の結果を比較する。
- 21 **FORMAT** 文 欄記述子 7.5
欄記述子について調べる。外部の欄の幅、小数部分のけた数、けた移動子をいろいろ変えてテストする。
- 22 **FORMAT** 文 欄記述子の組合せ、反復 7.5
欄記述子のいろいろな組合せと反復について調べる。
- 23, 24 文関数 1, 2 8.1
文関数と外部関数で同じ計算を行ない、その結果を比較する。
- 25 組込み関数 8.2
組込み関数について、同様の計算をする外部関数を作り、その結果を比較する。
- 26 基本外部関数 8.3.3
実数型と倍精度実数型の基本外部関数の値を比較する。複素数型の基本外部関数については、倍精度で計算する外部関数を作り、その結果と比較する。
- 27 外部関数 8.3
主プログラムと関数副プログラムで同じ計算を行ない結果を比較する。
- 28 サブルーチン 8.4
主プログラムとサブルーチン副プログラムで同じ計算を行ない結果を比較する。
- 29 初期値設定副プログラム 8.5
初期値設定副プログラムのDATA文と主プログラムおよびサブルーチン副プログラムのDATA文で同じ数値を代入し結果を比較する。

(1) 式のテストで問題になるのは、式の中の演算子の組合わせと、結果の評価である。たとえば算術式の場合の組合わせの個数は単純に考えても次のようになる。

1 次子には定数、変数の引用、配列要素の引用、関数の引用、かっこでくくられた算術式の 5 種類がある。

因子は 1 次子または 1 次子 * * 1 次子であり、後者が $5 \times 5 = 25$ 種類となる。

項は因子、項 / 因子、項 * 項であるが、かりに因子 / 因子と因子 * 因子だけを考えても $30 \times 30 \times 2 = 1800$

となる。

符号付項は項に正負の符号をつけたもので項の個数の倍の 3600。

算術式は 1 つの項、1 つの符号付項、あるいはそれらを + や - でつないだもので、かりに 2 つの項をつないだものも考えても 3600^2 以上になる。

このようにまともなすべての組合わせを計算することは実際上不可能であるから、組合わせの一部をランダムに拾ってテストしている。

次に算術式の結果が正しいかどうかの制定は、数値の精度が処理系にまかされている以上規定することはできない。ここでは実数演算は倍精度演算と、整数演算は実数演算と比較して、両者の相対的な差を求めて印刷しておくことにした。倍精度演算の結果はチェックできないことになるが、他の機種の結果と比較してみたらよからう。(どちらが誤った結果を得ているかはわからない)

- (2) 入出力関係のテストには、プログラム内部で自動的にチェックしてメッセージを印刷することが困難なものが多い。行送り、ページ送りなどは目で見て制定するよりない。書式仕様による数値の変換の場合にも、いったん結果を磁気テープにでも書き出して置いて、正解とくらべることは可能であろうが、いまは結果を読みとる方式にした。
- (3) 機器ごとに変更しなければならないものは、機器番号であり、その番号をプログラム内の 1 ケ所で変数に代入できるようにした。その他 1 つの変数に記憶できる文字数が異なる点については、原則を 6 文字とし、4 文字であるバイト・マシンでは REAL * 8 の変数として宣言することにした。
- (4) COMMON 文、EQUIVALENCE 文などについては、文法上規定されている場所に値が正しく格納されているかどうかを調べている。
- (5) 副プログラム関係のテストでは、副プログラムを利用した場合と利用しない場合の両方を計算して比較している。組みこみ関数に対しては関数を別に作成したが、基本外部関数に対しては、実数計算と倍精度計算とをくらべるのにとどめた。

4. テストの結果

プログラムは TOSBAC3400, TOPS3 の FORTRAN を用いて作成したが、一応完成したのち次の各機種にかけてテストを行なった。

TOSBAC3400, TOPS11

TOSBAC3400, TOPS14

IBM360/75 (Gレベル)

HITAC5020 (DOS 203 version 01)

NEAC2200-500 (mode 3 reliese 8 level K)

FACOM230-60 (monitor V SC7)

CDC3300 (USASI FORTRAN)

その結果の一部を次に示そう。

4.1 文法上の問題点

29本のテストプログラムに対して、各機種とも数本についてはコンパイル時または実行時にエラーを生ずる。それは文法的にあいまいな箇所の解釈のちがいでよるもの、規定より慣習が優先されているもの（逆にいえば規定の作り方がまずい？）、機器のちがいでよるもの、完全に虫であると思われるもの、などがある。いくつか例をあげておく。

(a) ENDFILE を実行した後 BACKSPACE をしたときどうなるかという処理系によってまちまちであり、次の3つの場合があった。

- (i) ファイル終了記録の直前の論理記録の前に装置を位置づける。
- (ii) ファイル終了記録の前に装置を位置づける。
- (iii) 禁止されている。

また一般的に BACKSPACE や ENDFILE に関してはバッファのとり方などによって制限がついている場合が多い。マニュアルにはいずれの場合にも簡単な説明があるだけで、そのような取扱上の注意にふれたものは少ない。互換性の点からもっときちんと定義すべきではないだろうか。

(b) EXTERNAL文

次の2つの例で問題が生じた。

(i) EXTERNAL 文で倍精度型や複素型の基本外部関数を定義し、そのプログラム内でその関数を使うと、1記憶単位分の結果しかもたらさない。このことはいくつか処理系であらわれているが、虫ではないだろうか。

(ii) 主プログラム

EXTERNAL SIN

CALL SUB1(SIN)

副プログラム1

SUBROUTINE SUB1(X)

CALL SUB2(X)

副プログラム2

SUBROUTINE SUB2(F)

$$Y = F(A)$$

のような例で、SUB2の中では

EXTERNAL X

が必要だと思われるが、この文があるとエラーになる処理系があった。これも明確に規定した方がよからう。

- (c) FORMAT 文中のかっこの深さについて、JIS規格では

欄記述群のかっこの中には——別の欄記述群が含まれてもよい。

とあるので、かっこは何重になってもかまわないことになる。しかし実際にテストをしてみると、大多数の処理系で、かっこは2重まで(FORMAT文の直後のカッコは別にして)に制限されている。ISOの規格では2重までの制限がついているように読めるけれども、あきらかにJISはみたしていないことになる。

- (d) 基本外部関数 ATAN2(A1, A2)は、たとえばJIS規格で

$$\arctan(a_1/a_2)$$

となっており、ATAN(A)の方は

$$\arctan(a)$$

とされている。関数値をどの範囲に定義すべきかは指定されていないが、

$$\text{ATAN2}(A1, A2) \text{ と } \text{ATAN}(A1/A2)$$

は同じと考えざるをえない。しかし実際には大部分の処理系で

$$\text{ATAN} \text{ の方は } \pi/2 \text{ と } -\pi/2 \text{ の間に定義され}$$

$$\text{ATAN2} \text{ は } \pi \text{ と } -\pi \text{ の間に定義されている。}$$

実際使用する上でもこの方が便利である。このような違いがなければATAN2をわざわざ用意する意味もない。規格にも、マニュアルにもこれを明記すべきであろう。

- (e) FORTRANの規定にはreserved wordはないので、DOとかIFとかいう名前の変数を用いてかまわないはずであるが、いくつかの処理系でエラーが出ている。
- (f) バイト・マシン(IBM360など)でCOMMON文に並べる変数の順序に制約があるものが多い。このためにCOMMON文のテストプログラムが通らない。
- (g) 代入文、データ文、READ文で同じ形式のデータを変数に与えたとき、内部表現が同じにならないものがいくつかあった。

その他、EQUIVALENCE配列内の書式仕様、倍精度演算などに虫があることが多かった。その他、処理系ごとに数値の桁数が異なるために、計算の途中でオーバーフローがおこって、計算が先へ進まない場合があった。テストプログラム中では、どんな機種にも安全な程度に小さい数を取扱かっておくべきであった。しかし、オーバーフローのおこった時に、すぐに計算を打切ってしまうもの、何回かそのような現象がおこったときに打切るもの、メッセージだけ印刷して計算を続けるものなど、処理は異なっている。

4.2 計算の精度について

FORTRAN で計算された数値の誤差は、もちろんその機種における数値の最大ケタ数、計算の内容によって変るもので、評価の対象となるかどうか分からない。参考のため2つの例をあげておく。

- (a) 実数計算の精度を、倍精度計算と比較して求めている。(プログラム07)、表2、表3はその結果を示すものである。面白いのは因子や項の計算で、相対誤差の符号を調べると処理系によってかなり偏った分布を示すことである。

表2 算術式の精度の桁数による分類

機 種		A	B	C	D	E	F	
仮数部のビット数		23	23	27	38	38	38	
種 類	精度				精度			
	ケタ				ケタ			
因 子 (サンプル数24)	05	0	6	0	09	0	0	0
	06	12	13	1	10	16	21	21
	07	3	0	16	11	2	0	0
	以上	9	5	7	以上	6	3	3
項 (サンプル数35)	05	13	11	0	09	0	2	5
	06	16	19	0	10	26	28	23
	07	3	4	25	11	7	4	6
	以上	3	1	10	以上	2	1	1
符号つき 項 (サンプル数30)	05	4	5	0	09	1	1	2
	06	17	17	0	10	14	20	21
	07	3	3	15	11	10	6	4
	以上	6	5	15	以上	5	3	3
算術式 (サンプル数18)	05	3	2	0	09	2	4	3
	06	12	13	1	10	11	10	11
	07	2	3	10	11	5	4	4
	以上	1	0	7	以上	0	0	0

表3 算術式の誤差の符号による分類

機種		A	B	C	D	E	F
仮数部のビット数		23	23	27	38	38	38
種類	符号						
因子 (サンプル数24)	-	10	20	0	18	19	19
	0	3	3	5	0	3	3
	+	11	1	19	6	2	2
項 (サンプル数35)	-	28	32	18	22	22	26
	0	2	3	0	0	0	0
	+	5	0	17	13	13	9
符号つき 項 (サンプル数30)	-	17	22	7	13	12	18
	0	4	1	2	2	2	2
	+	9	7	21	15	16	10
算術式 (サンプル数18)	-	12	16	12	7	8	12
	0	1	0	0	0	0	0
	+	5	2	6	11	10	6

- (b) 基本外部関数EXPに対して引数を変えて計算し、精度のケタ数を求めたものが表3である。(プログラム26)このような関数は、最も誤差の大きいところで2桁以上精度が落ちていることがわかる。(a)の方の計算結果でも同じような事情があらわれている。

表4 基本外部関係EXPの相対精度

		A	C	G	F
仮数部のビット数		23	27	36	38
引数	関数值				
-775921	0.20E-33	5	4	9	9
112487	0.71E-49	6	5	8	9
-443835	0.53E-19	6	6	8	9
-141421	0.72E-06	6	7	10	10
-638522	0.18E-27	5	6	9	9
-47925	0.15E-20	5	6	10	9
-605036	0.52E-26	5	5	9	9
-118857	0.68E-05	5	7	9	9
834815	0.18E-37	5	5	9	9
-90927	0.11E-03	6	7	9	9
-59212	0.26E-02	6	5	9	11
-42683	0.29E-18	6	6	9	9
-1967	0.13E-00	6	8	10	11
-09365	0.39E-00	7	7	11	11
12341	0.22E-06	6	7	10	10
0062	0.10E-01	6	9	10	11
296901	0.78E-13	4	6	9	9
35425	0.24E-16	5	7	8	9

5. まとめ

今回作成したFORTRAN のテストプログラムは、前に説明した若干の問題点を除けば、各機種に通すことができ、各処理系ごとの文法の解釈の差や制限事項など処理系に対する互換性を制限する事項がかなりはつきりと実験できた。JIS 規格や、各社の説明書の中のあいまいさや書きおとしについては今後検討を加えることがのぞまれる。このテストプログラムは処理系製作者が完成テスト用プログラムに使うこともできるし、また利用者の側での検査や、互換性のくわしい検討に使用することができる。

今回の報告はいちおう文法上のことがらに重点をおいたが、元来この研究はもっと一般的な評価の問題の手掛かりとして行なったものである。プログラムないし計算機システムの総合評価ということを考える場合、関係するファクタは数多いが、今回の経験から得られたいくつかの感想を記しておく。

(a) 処理時間

評価のファクタとしてもっとも大きいものの1つは処理時間であろう。各機種でのテストの際の処理時間は大体出力されているが、それらを直接比較するわけには行かない。処理の方法およびI/O機器の能力をどのように評価し、計数化して行くのであろうか。部分的にプログラムの最適化とか、翻訳時間の多少などというところを相対的に評価しても、全体としてどのように評価すべきだろうか。

(b) エラーの処理

ソース・プログラムのミスがあるのに気づかず、別の機種にかけてみたところメッセージが出てわかった例もある。また実行時のエラーについても、メッセージが出たり、出なかったり、計算が打ち切られたりする。使用者の側に立てばエラーの処理はプログラム評価の大きなファクタになる。

(c) 計算の誤差

計算した結果に生ずる誤差は数値計算的に求めることはできるけれども、基本外部関数の誤差などは処理系にまかせているだけでよいのだろうか。系統的に誤差が出る場合には問題になるかもしれない。

本 PDF ファイルは 1970 年発行の「第 11 回プログラミング・シンポジウム報告集」をスキャンし、項目ごとに整理して、情報処理学会電子図書館「情報学広場」に掲載するものです。

この出版物は情報処理学会への著作権譲渡がなされていませんが、情報処理学会公式 Web サイトに、下記「過去のプログラミング・シンポジウム報告集の利用許諾について」を掲載し、権利者の検索をおこないました。そのうえで同意をいただいたもの、お申し出のなかったものを掲載しています。

https://www.ipsj.or.jp/topics/Past_reports.html

過去のプログラミング・シンポジウム報告集の利用許諾について

情報処理学会発行の出版物著作権は平成 12 年から情報処理学会著作権規程に従い、学会に帰属することになっています。

プログラミング・シンポジウムの報告集は、情報処理学会と設立の事情が異なるため、この改訂がシンポジウム内部で徹底しておらず、情報処理学会の他の出版物が情報学広場 (=情報処理学会電子図書館) で公開されているにも拘らず、古い報告集には公開されていないものが少からずありました。

プログラミング・シンポジウムは昭和 59 年に情報処理学会の一部門になりましたが、それ以前の報告集も含め、この度学会の他の出版物と同様の扱いにしたいと考えます。過去のすべての報告集の論文について、著作権者 (論文を執筆された故人の相続人) を探し出して利用許諾に関する同意を頂くことは困難ですので、一定期間の権利者搜索の努力をしたうえで、著作権者が見つからない場合も論文を情報学広場に掲載させていただきたいと思えます。その後、著作権者が発見され、情報学広場への掲載の継続に同意が得られなかった場合には、当該論文については、掲載を停止致します。

この措置にご意見のある方は、プログラミング・シンポジウムの辻尚史運営委員長 (tsuji@math.s.chiba-u.ac.jp) までお申し出ください。

加えて、著作権者について情報をお持ちの方は事務局まで情報をお寄せくださいますようお願い申し上げます。

期間：2020 年 12 月 18 日 ~ 2021 年 3 月 19 日

掲載日：2020 年 12 月 18 日

プログラミング・シンポジウム委員会

情報処理学会著作権規程

<https://www.ipsj.or.jp/copyright/ronbun/copyright.html>