

## A6-2 プログラム言語標準化の最近の動向(FORTRAN)

菅 忠 義 (学習院大学)

### 0. まえがき

ここでは、1章、2章でFORTRAN標準化までの厂史と標準化に際してとられた基準に関して述べる。ついで3章には、72年3月制定の改正されたJIS FORTRANについて、その経過と改正の方針と改正英とを述べ、また改正作業中に集められた将来への提案を示す。さらに4章では、先ず米国でのFORTRAN標準化の時英(1966年)以後今日(1972年12月)までの動向を述べ、ついで1972年11月に開かれたISO/TC97/SC5の総会での動向について述べる。また5章に米国からISOに送らされている改正と拡張案の内容を示す。

### 1. FORTRAN標準化の厂史

ここでは、1.1に、业界で最初に標準化の作業に着手した米国における経過を述べ、ついで1.2に、ISOにおけるFORTRAN標準化の経過を述べる。また1.3では、日本において初めてJIS FORTRANが制定されるまでの経過を述べる。

#### 1.1. 米国における標準化の厂史

米国では、1960年に、BEMA (Business Equipment Manufacturers Association) の後継のFに、ASA (American Standards Association) に Sectional Committee 又は組織され、この委員会が、コンピュータと情報処理の問題を扱うことになった。ASA 又は、~~又は~~ 又は、4 という Sectional Subcommittee をつくり、この委員会がプログラミング言語の規格の問題を扱うことを決め、さらに1962年5月17日に、作業委員会又は、4.3-FORTRAN と組織し、米国の標準言語FORTRANの原案作成に着手することと決議した。この又は、4.3には、Policy Committee と Technical Committee とから成り、Policy Committee がW4の作業の完遂に関して又は、4に責任を持ち、基本的方針を決めたり、Technical Committee へ指示したりする役割りを果たし、Technical Committee が実際の調査研究と言語仕様の原案作成および Policy Committee への報告書の作成をする役割りを果たすことになった。又は、4.3の第1回の会合が、1962年8月13、14日に New York で開かれ、実際の作業が始まった。この時期に米国では、いわゆるFORTRAN IV と FORTRAN II とが普及していたので、二つの Technical Committee 又は、4.3-IV と 又は、4.3-II とがつけられ、それぞれがそれぞれ原案を作成することになった。又は、4.3-II は、1963年5月に原案を完成し、これを承認した。また、又は、4.3-IV は、1964年4月に、原案を完成した。1964年6月に又は、4.3が二つの原案を受け付け、比較し、両者の内容の矛盾を解決し、また表現形式と術語の相異と解決した。これは、又は、4.3-II の原案を、その本来の内容を保存して、又は、4.3-IV の枠の中に埋め込むようにつくりなおすことにより行なわれた。又は、4.3は、混乱をおこさないように、又は、4.3-II の原案に対応する言語を Basic FORTRAN<sup>3)</sup> と呼び、又は、4.3-IV の原案に対応する言語を FORTRAN<sup>2)</sup> と呼ぶことに決めた。1966年3月に、この両者は USA Standard として正式に承認された。

#### 1.2. ISOにおける標準化の厂史

1960年6月に、ISOの中に電子計算機と情報処理の国際的な標準化と専門的に行なう

Technical Committee 97 (TC97と略称) が設定され、1961年には、この委員会の下にワーキンググループの標準化に担当する作業グループが設置され、1963年にこの作業グループが Subcommittee 5 (SC5と略称) に格上げされた。1965年3月に、このSC5に対して、ASA (1966年の改組でUSASIとなり、さらに1969年9月の改組でANSIとなる) およびECMA (欧州電子計算機工業会) からFORTRANの規格原案が提出された。ISO/TC97/SC5は、この規格原案を参加各国に配布して、意見を求め、多々の修正を行なって1965年10月東京で開かれたISO/TC97の総会でISO推薦規格案 (Draft ISO Recommendation) となった。ISO FORTRANには三つの水準があり、大きい方から順に、FORTRAN, Intermediate FORTRAN, Basic FORTRANと呼ばれ、それぞれASAのFORTRAN, ECMAのFORTRAN, ASAのBasic FORTRANに対応する。日本ではISO/TC97/SC5からASAおよびECMAの提出したFORTRANのISO規格原案に対する意見を求められた際に、情報処理学会の規格委員会のSC5委員会 (当時 森口直直) でこのISO規格原案を検討する作業委員会をつくった。この作業委員会は、1965年春から4回の委員会を開き、検討作業を行なって、意見をまとめISO/TC97/SC5へ提出した。日本から出された意見のかなりの部分がISO推薦規格案に盛り込まれている。なお、1968年4月にISO推薦規格案の文書<sup>14)</sup> (ISO DR 1539 Programming Language FORTRAN) が送付され、ISO推薦規格とするための投票が行われ、日本は賛成投票を行った。

### 1.3. 日本における標準化の歴史

1965年10月にFORTRANがISO推薦規格案となった時期に、日本電子工業振興協会のALGOL/FORTRAN分科会 (森口直直) で、FORTRANの日本工業規格を作成する意見が出たので、1965年11月にALGOL/FORTRAN分科会にFORTRAN作業委員会をつくり、JIS FORTRANの原案作成に着手した。1965年中に3回、1966年中に約30回の作業委員会が開かれた。その間に工業技術院から日本電子工業振興協会にJIS FORTRANの原案作成の依頼があり、協会はその作業をこの作業委員会に一任した。

1966年春には、最初の原案が完成し、同時に完成したALGOLの原案とともに5月9日にこれらに関する公開の説明会が開かれた。その後広く一般および識者から寄せられた意見を参考にして数度の改訂を行なったのに、JIS原案として完成したものが、工業技術院に提出され、そこでさらに規格の体裁をよびのえるための修正が行なわれて、1967年5月1日に日本工業規格<sup>5)</sup>として制定された。

## 2. FORTRAN標準化の基準

ここでは、2.1で米国で初めてFORTRANが標準化される際にとられた基準を示し、2.2でISOでのFORTRAN標準化の際にとられた基準を述べる。さらに2.3でJIS FORTRANの原案作成の際の基準を述べる。

### 2.1. 米国における基準

CACMにのせうに Heising の論文<sup>1)</sup>によると、ASA の又 3.4.3 の基準はつぎのようであった。(ここの順序は必ずしも重要さの順になつてゐるわけではない。)

- A. Ease of use by humans
  - B. Compatibility with past FORTRAN use
  - C. Scope of application
  - D. Potential for extension
  - E. Facility of implementation, i.e. compilation and execution efficiency
- さらに FORTRAN 規格の目的としてつぎのことが述べられている。

"The FORTRAN standard will facilitate machine-to-machine transfer of programs written in ASA Standard FORTRAN. The standard will serve as a reference document both for users who wish to achieve this objective and for manufacturers whose programming products will make it possible. The content and method of presentation of the standard will recognize this purpose."

## 2. 2. ISO における基準

ISO における基準は、JIS C 6201<sup>6)</sup> の解説中に述べられているが、これは ISO FORTRAN の APPENDIX A より抜粋したものである。それによるとつぎのようである。  
"FORTRAN を標準化するにあつて用いた基準は (ほぼ重要さの順位に従つて)、つぎのとおりである。

- (1) 異種処理系間での FORTRAN プログラムの互換性
- (2) 現存のものと矛盾しないこと
- (3) FORTRAN 使用者にとって明確かつ簡単であること
- (4) 本範囲の計算機の能率的な動向への適合性
- (5) 言語に将来の発展の余地を残すこと

この FORTRAN の規格を制定するにあつて、実用されていない言語機能は新たに付け加えず、もっぱら現行の FORTRAN の成文化にとどめた。FORTRAN の過去および現在の広範囲な使用状況からみて、規格の制定にあつて言語の設計よりもむしろ言語の定義にもっぱらつとめ、

- (1) 規格の内容が実用場で価値を証明されたもののみを反映するようにし、また
- (2) この規格を受け入れ易いようにした。"

この ISO での基準と 2.1 に示した米国での基準とから FORTRAN の言語特性を知ることはできる。すなわち、これらの基準において互換性の重視は規格というものの本来の意味から当然として、

- ・容易性
- ・効率性

が重視されていることが明かであり、これらのことから FORTRAN の言語構造の特徴を理解

することが出来る。

### 2.3. 日本における基準

日本でのFORTRANの標準化の作業が始められたのは、前述したように1965年11月であり、この時点ではISOで規格原案とISOの推薦規格案とすることが承認されていた。そこで日本ではつぎのことが標準化の方針としてとられた。

- (1) ISO FORTRANに準拠すること
- (2) なるべく良い日本語にすること

ここで(1)は、必ずしもISO FORTRANとして公表された推薦規格案をそのまま受け取るということではなく、むしろISO FORTRANの本意を思っていると思われるものを推察して、その立場から日本の実情を考慮して標準化するということである。また(2)は、当時はFORTRANに関する術語はすべて英語のままを用いられていたが、この際なるべく良い日本語としての術語をつくること、およびISO FORTRANの条文を単に直訳するだけでなく規格書の様式(JIS Z 8301)の制限内で日本語として良い文章になるようにすることの意味がある。

以上で明かなようにJIS FORTRANはISO FORTRANに準拠しているもので、その内容上の基準は、ISOの基準と同じである。これは、本来JISは、ISOに準拠するという原則によるためである。しかしこれはISOで、すでに推薦規格案が承認されてしまった時点での原則であり、ISO規格原案が検討されている段階では、日本として独自の意見や提案を提出することは可能なものであり、むしろ日本としてはそのような努力をすべきであろう。

### 3. 改正JIS FORTRANについて

ここでは、1972年3月に改正されたJIS FORTRAN<sup>6)</sup>について、3.1で改正の経緯を、3.2で改正の方針を、3.3に改正案と述べ、3.4で改正の作業中に集められた将来の改正への提案を示すことにする。

#### 3.1. 改正の経緯 (JIS C 6201<sup>6)</sup>の解説 2.2.1 参照)

1970年11月4日に日本工業標準調査会のプログラミング言語専門委員会が招集され、FORTRANとALGOLのJISの改正問題を審議し、FORTRANとALGOLについてそれぞれ分科会をつくり、1971年3月1日改正を目標として、1971年1月17日に予定された第2回プログラミング言語専門委員会までに改正原案を作成することに決められた。この直後より改正のための活動が始められ、FORTRAN分科会は、1970年11月18日に第1回の委員会を開いた。それ以後、7回の委員会を開き1971年1月中旬に改正原案の作成を終了した。しかし1月17日に予定されていたプログラミング言語専門委員会は延期され、4月28日に開かれた。この専門委員会が改正原案を審議し、それに基づいて修正された原案を5月中旬に作成し、これについては書面審議を行ない1月末までに意見を求めることが決められた。これ以後、寄せられた意見の検討と最終原案の作成、解説の内容の検討と作成の作業が進められ、9月末に最終原案が工業技術院に提出された。このあといくつかの手続きを経て、1972年3月1日に正式にこの最

終原案が改正 JIS FORTRAN<sup>6)</sup>として制定された。

### 3.2. 改正の方針 (JIS C 6201<sup>6)</sup>解説 2.2.2 参照)

1971年3月を目標に改正するという日程を考慮して、第1回プログラミング言語専門委員会が改正の方針としてつぎのことを決められた。

- (1) 最小限の改正にとどめること
- (2) 新しい機能を加えるとか、文法体系を構成しなおすということはないので、現行のものとそのなかの中であるべく完全なものにすること
- (3) 改正についての意見は、上記の条件と関係なく広く求めること。

このように今回の改正は、極く限られた範囲で行なわれたが、(3)の方針から、同時に将来への改正提案も集められ、今後の改正の際の資料をつくることになった。

### 3.3. 改正案 (JIS C 6201<sup>6)</sup>解説 2.2.3 参照)

ここでは、今回の改正のおもな案を列挙する。

- (1) 備考と参考
- (2) 基本的な術語
- (3) 一般の定数
- (4) 手続き
- (5) 評価されない部分
- (6) 単純 GO TO 文
- (7) 拡張範囲
- (8) ファイル
- (9) 抑記述群
- (10) ATAN 2

これらについては、JIS C 6201<sup>6)</sup>の解説 2.2.3 に、その内容と改正の理由が示されている。その他にも細かい案で多くの修正が行なわれているが、それについては、文献<sup>8)</sup>を参照された。

### 3.4 将来への改正意見

今回の改正に際して集められた意見を整理して、作業用資料 10-1~4 を作成し、これによって作業を進めたが、この中には、将来への改正意見も含まれるようにしたので、この資料の中から将来への改正意見を取り出して、表 I に列挙することにする。ここで、1000 という桁の番号は、JIS C 6201<sup>6)</sup>の節番号で、改正意見の関係するものである。ref. という桁の番号は、その意見が提案されている資料番号 (FORTRAN 分科委員会での資料番号) である。

## 4. 最近の米国と ISO における動向

ここでは、4.1 で米国における ASA FORTRAN 制定以後の動向を述べ、4.2 で 1972年 11月に開かれた ISO/TC 97/SC5 の plenary meeting における FORTRAN に関する事項を述べる。

### 4.1 米国での制定以後最近までの動向

1966年に ANS (当時 ASA) FORTRAN<sup>2)</sup> と Basic FORTRAN<sup>3)</sup> が制定されて以来、米国では、

二つの方向の作業が行なわれている。すなわち、一つはこの規格に関する clarification の作業であり、他一つは revision と extension のための作業である。そこでここでは先ず 4.1.1 において前者のいきさつと文献 9), 10) によって述べ、ついで後者については、4.1.2 において文献 11), 12), 13) によって紹介する。

なお、現在 X3.4.3 は、X3J3 と改組されている。

#### 4.1.1. Clarification について

CACM に 1969 年に発表された Clarification of Fortran Standards - Initial Progress<sup>9)</sup> と、1971 年に発表された Clarification of Fortran Standards - Second Report<sup>10)</sup> によると、clarification のいきさつはつぎのようである。

“文献 1) が公表されるに、コメントや疑問や批判が起り始めたが、それらを分類すると、つぎの三つの種類に分けることができる。

- (1) ある言語機能を含めたいあるいは除外したいという希望
- (2) 言語を minor extension したいという要求
- (3) 規格の適用範囲、目的、内容および解釈についての問題

多くの(1)に関する事項は、また(3)についての誤解に關係して起りつつある。

承認された規格について 1 年間に意見を求めた後に、規格の適用に關して難解な点や誤りを見つけ出し、それらについて適当な補修をすべきであるということが決められた。そこで 1967 年 12 月に X3.4.3 (現在は X3J3) が招集され、ANS FORTRAN と Basic FORTRAN について、解釈上の問題、誤り、脱落を取り扱う working subcommittee X3.4.3 B をつづた。この作業は、initial clarification document を 1969 年に公表することによって完了し、ついでこの X3.4.3 B は解散された。しかし clarification の作業は、X3J3 自身によって続けられ、1971 年に second report が公表された。”

clarification のいきさつは、以上のようであるが、その内容についてはここでは触れないことにする。それについては文献 9), 10) を参照されたい。

#### 4.1.2. Revision と Extension について

ANSI では、ANSI を 5 年毎に見直しし、承認するかの改正を可るかを決められることになっている。実際、X3J3 は、見直しを行ない、拡張を含む改正を行なうことを決め、その作業が進められ、その資料が ISO に提出されている。1970 年 4 月に、ISO/TC 97/SC 5 に提出された CRITERIA FOR 1971 DRAFT PROPOSAL<sup>11)</sup> によると、改正の基準はつぎのようである。

1. Interchangeability of FORTRAN programs between processors
2. Compatibility with X3.9-1966, allied standards, and existing practice
3. Consistency and simplicity to user
4. Suitability for efficient processor operation for a wide range of computing equipment of varying structure and power
5. Allowance for future growth in the language
6. Achievement of capabilities not currently available, but needed for processes appropriately expressed in FORTRAN

7. Acceptability by a significant portion of users
8. Availability of Preliminary Draft Proposed Standard (PDPS) in 1971
9. Improved ability to use FORTRAN programs and data in conjunction with other languages and environments

これをみると、制定時の基準とその趣旨は、依然として残されている。また新たに付加された基準として、2.に改正が又3.9-1966の趣旨と compatibility を保つて行われるべきこと、また6.に拡張と考慮すること、および9.に other languages and environments と関連して FORTRAN のプログラムやデータを利用しうらうにすることか述べられている英を注目すべきであろう。

改正と拡張に關する内容は、1972年8月に、COLLECTION OF X3J3 APPROVED FORTRAN REVISIONS AND EXTENTIONS<sup>12)</sup> という膨大な資料が ISO/TC97/SC5 に提出されている。また、X3J3 による提案を考慮されているすべての拡張案と簡条書きにした Appendix E of document X3J3/34のメモが、1972年8月に ISO/TC97/SC5 に資料<sup>13)</sup> として提出されている(これを5章に表として示す)。これらにより現在の米國における動向を知るべきである。

#### 4.2. 1972年11月の ISO/TC97/SC5 の Plenary Meeting

この総会での FORTRAN の AD HOC WORKING GROUP の会合の議題は、つぎのようであった。

- a. Presentation of mechanical flaws in R 1539<sup>11)</sup>
- b. Presentation of SC5 N 283
- c. Differences between the Japanese Industrial Standard and R 1539
- d. Relationship of ANSI clarifications and ISO R 1539
- e. Presentation SC5 N 280<sup>11)</sup>, 281<sup>12)</sup> and 282<sup>13)</sup>
- f. Recommendation to SC5
- g. Review SC5 N 270 for compatibility with R 1539
- h. Mechanical flaws.

このうち改正と拡張に關する議題は、e, f, g, であるのでこれについての討論内容と簡単に述べる。

e. は、ANSI の X3J3 による改正と拡張案の提出と、その説明であるが、これに対して United Kingdom は、subarrays に関する拡張案に強く反対し、日本のこれを支持した。投票の結果、subarrays proposal は、FORTRAN として不適当であるとみとめられることになった。

f. は、四つの recommendation であるが、このうち #1 について紹介する。

#1: ANSI は、できるだけ早く、ISO/TC97/SC5 へ、その Draft Proposed FORTRAN Standard を送付すること。また、この文書は現在の ISO FORTRAN R 1539 に対する ISO proposed revision のための一つの基礎として考えるべきこと。

この recommendation は、<sup>FORTRAN 上の今後のこと</sup> 議決事項として採用されている。

つぎに、総会での FORTRAN に関する resolution を示す。

RESOLUTION 8 <sup>that</sup> ANSI be requested to forward to ISO/TC97/SC5 its Draft proposed FORTRAN Standard at the earliest possible time. This document should be considered

by ISO/TC 97/SC 5 as a possible basis for any ISO Proposed Revision to the current ISO FORTRAN R 1539

RESOLUTION 9 - that, if the decision is made to revise the current ISO recommendation on FORTRAN (R 1539) greatest care be taken to avoid incompatibility with the present standard. However, should incompatibilities arise, they must be documented in appendix to a new standard.

RESOLUTION 10 - that SC 5 take no action at the present time towards forming a FORTRAN maintenance (working) group.

RESOLUTION 11 - that the list of mechanical flaws be forwarded to the SC 5 Secretariat for verification. Upon verification by the Secretariat, the mechanical flaws shall be corrected in subsequent printings of ISO/R 1539. See Document N 314.

#### 5. 米国での改正と修正案

ここでは、ISO/TC 97/SC 5 (USA-41) 282<sup>13)</sup>の内容を表 2 に示す。

#### 6. おわりに

以上、FORTRANの標準化について、制定時から最近までの動向を述べたが、米国での作業の進行状況からみれば、ISO FORTRANの内容は、日本の積極的な意見と反映させるためには、なるべく早い時期に原案作成の作業に着手すべきであろう。

#### 参 考 文 献

- 1) Neising, W.P.: History and Summary of FORTRAN Standardization Development for the ASA; CACM Vol. 7 (1964), No. 10
- 2) ASA X3.4.3: American Standard FORTRAN. (X3.9-1966); ASA
- 3) ASA X3.4.4: American Standard Basic FORTRAN (X3.10-1966); ASA
- 4) ISO/TC 97/SC 5: DRAFT ISO RECOMMENDATION FORTRAN, (1965); ISO
- 5) FORTRAN起草委員会: 電子計算機用語 FORTRAN (水準 7000) JIS C 6201, (1967)
- 6) FORTRAN分科委員会: 電子計算機用語 FORTRAN (水準 7000) JIS C 6201, (1972)
- 7) FORTRAN分科委員会: FORTRAN分科委員会資料, (1971)
- 8) 菅忠義: JIS FORTRANとALGOLの改正; bit Vol. 3 (1972)
- 9) USASI X3.43B: Clarification of Fortran Standards - Initial Progress; CACM Vol. 12 (1969), No. 5
- 10) ANSI X3J3: Clarification of Fortran Standards - Second Report; CACM Vol. 14 (1971) No. 10
- 11) ANSI X3J3: CRITERIA FOR 1971 DRAFT PROPOSAL; ISO/TC 97/SC 5 (USA-39) 280
- 12) ANSI X3J3: COLLECTION OF X3J3 APPROVED FORTRAN REVISIONS AND EXTENSIONS. ISO/TC 97/SC 5 (USA-40) 281; (1972/8)
- 13) ANSI X3J3: ISO/TC 97/SC 5 (USA-41) 282; (1972/8)
- 14) ISO/TC 97/SC 5: ISO RECOMMENDATION FORTRAN (R 1539);



表 1

7000	改正意見	7000	改正意見
1.1	水準設定	7.1.5	新しい巻頭の句題
3.1	\$ &	7.3	整合手法の共通プロシキの受け渡し
3.1.1	8進数, 16進数		動的な EQUIVALENCE 宣言
3.2.1	行かけの数		EQUIVALENCE文の配列名
	注釈行の文字		EXTERNAL文の組込み関数
4.1	文字型の変数名		文節数
4.2	倍精度複素数型		F <sup>n</sup> -F <sup>m</sup> の長さの byes <sup>n</sup> を指定
	4倍精度型		型宣言文での初期値設定
	倍精度整数型		IMPLICIT文
	ラベル型	7.4	DATA文の括弧, DO型並び, 共通プロシキの初期値設定
4.3	論理型と1記憶単位との違い		
5.1.1	16進数に於ける型指定	7.5.2	nH 3F17 nX17 空とありとあり
	引用符にFを文字定数	7.5.6	数値入力の桁内空白
5.1.3	配列の次元		出力でwとdを不足しと処理
6.1	論理型以外の混合		変換記号とF <sup>n</sup> の型の混合
7.1.1	算術代入文の左辺と右辺の型	7.5.10	配列内の書式仕様
	多重代入文	8.2	AND, OR, NOTの組込み関数
	マスキング代入文	8.3.3	関数名の命名法
7.1.2	割当てのGO TO文のコンマと文番号の省略		Automatic Function Typing
	計算的GO TO文の変数の値	8.4.2	文字定数の実数(長さ)
	2命令IF文		Reference by name or value
	低い水準での論理IF文とF <sup>n</sup>		ENTRY文
	DO文の10 <sup>n</sup> ×F		RETURN文
	DOの端末文	9.	名前と子との機能
	括弧範囲のDO文		
7.1.3	単数並びのコンマのrest		
7.1.4	7.1.4 入力出力文		
	ENCODE文, DECODE文		
	NAME LIST文		
	入力出力の括弧		
	END, ERROR option		
	書式なし標準 WRITE文		

I. Minor revisions and certification of common practice:

1. Define effect of REWIND when unit at initial position.
2. Require ASSIGN statement to be in same program unit as assigned GO TO or FORMAT statement to which it refers.
3. Require consecutive blanks in format specification to cause printing of blank lines.
4. Allow an optional comma in assigned & computed GO TO statements and preceding the control variable in DO statement.
5. Allow in a comment line any characters capable of representation by the processor.
6. Include the apostrophe ' in the FORTRAN character set.
7. Allow Hollerith character strings to be delimited by apostrophes.
8. Allow quoted character strings in PAUSE statements.
9. Allow decimal digits PAUSE & STOP statements.
10. Require character in STOP statement be accessible.
11. Allow unlimited number of digits in real or double precision constant.
12. Require that the number of digits in an input datum under D, E, F or G type conversion is limited only by the field width W.

II. Minor extensions to the language:

1. Make optional the statement label list in the assigned GO TO statement.
2. Allow the IMPLICIT statement.
3. Allow as terminal statement of a DO-loop, any executable statement for which, upon completion of its execution, execution of the next following statement may take place.
4. Allow READ, PRINT & PUNCH statements, with respective units processor defined.
5. Allow many levels of parentheses in a format specification.
6. Require printing of asterisks in entire field when value exceeds field width in type I,m or F,w,d conversion.
7. Restrict scale factor nP in E & D output fields, such that  $-d < n < d+2$ .
8. Allow optional sign in scale factor in FORMAT statement.
9. Allow ASSIGN statement to set the value of variable format references in a formatted I/O statement.
10. Specify units of arguments and results for the Basic External Functions.
11. Allow a program statement in main program unit.
12. Allow naming of BLOCK DATA subprograms.
13. Allow more than one BLOCK DATA subprogram in an executable program.
14. Require END line to act as a STOP statement in a main program unit, and as a RETURN statement in a subprogram.
15. Remove concept of second level definition.

III. General extensions to the language:

1. Allow functions with no arguments to be defined and reference. e.g. F0
2. Extend the rules for assignment of e to v, Table I, to allow complex expressions with real, double precision or integer entities, and vice versa.
3. Require a complex entity to be equivalent to two type real entities, the first being the real part and the second being the imaginary part of the complex number.
4. Allow multiple assignment statements.
5. Specify right to left grouping of successive exponentiations, removing ambiguity  $A**B**C**D$  means  $A**(B**(C**D))$ .
6. Require left to right expression evaluation for operations of same hierarchy level.
7. Allow mixed mode arithmetic expressions.
8. Allow array to have maximum of seven dimensions.
9. Allow implied-DO in DATA statement list.
10. Allow array name (sans subscript) in DATA statement list.
11. Allow array name in EQUIVALENCE statement.
12. Allow an array element name in a statement function expression.
13. Allow generalized subscript expression.
14. Allow adjustable dimensions (i.e. variable names used as) in COMMON statement.
15. Allow redefinition and/or undefinition of adjustable dimensions within a subprogram unit without affecting the array dimensions.

16. Allow integer expressions as adjustable dimensions.
17. Require additional Basic External & Intrinsic Functions:  
ASIN, DASIN, ACOS, DACOS, SINH, DSINH, COSH, DCOSH, DTANH, TAN, DTAN, DINT  
DFLOAT, DDIM, DPROD

18. Allow ERR= in READ/WRITE statements.
19. Allow END= in READ statement.
20. Allow more than one file to appear on one unit (sequentially).
21. Allow list directed I/O
22. Allow expressions in output lists.
23. Allow quoted character constant in output list of formatted WRITE statement, with a corresponding Aw format field descriptor required.
24. Allow a designated character to stop scan of a format specification at the end of an I/O list.
25. Allow -nX and +nX format field descriptors for relative tabbing, with implied initialization of record to blank on output.
26. Allow T format field descriptor for absolute tabbing, with implied initialization.
27. Require processor to print + sign, positive or zero exponent in E,w,d field.
28. Allow control of exponent sign on E,w,d,e & D,w,d,e conversion output.
29. Allow BACKFILE & SKIPFILE statement to position sequential unit.
30. Allow integer expressions in computed GO TO statement.
31. Require the execution of the next following statement, if the computed GO TO expression is out of range.
32. Allow real or double precision type control variable in DO statement.
33. Allow any arithmetic expression in DO-parameter with appropriate conversion.
34. Allow negative values as DO-parameter.
35. Require determination of incrementation value and number of range executions at start of execution of DO statement.
36. Require minimum number of DO iterations to be zero.
37. Allow redefinition and/or undefinition of DO parameter variables within range without affecting number of iterations.
38. Allow branching into the range of any active DO.
39. Allow a CONSTANT(PARAMETER) statement.
40. Allow ENTRY statement in subprogram to provide additional entries & names.

IV. Array extensions to the language:

1. Allow non-unity lower bounds for an array dimension.
2. Allow block transfer list item an I/O list.
3. Allow an array cross section reference and (-\*) for a reverse cross-section.
4. Allow a SUBARRAY statement.
5. Allow array assignment statement with any arithmetic expression. An array assignment statement is an assignment statement in which v is an array cross-section. The expression e may or may not contain array cross-section references.

V. Direct Access I/O extensions:

1. Allow OPEN FILE and CLOSE FILE statements.
2. Allow direct access READ & WRITE statement, both formatted and unformatted.
3. Allow direct access files to contain fixed length or variable length records.

VI. Character Data type extensions:

1. Allow CHARACTER type specification statement.
2. Allow character type constants.
3. Allow character string reference in both variable and array elements.
4. Allow character type expressions in assignment statement, argument of subprogram, and I/O lists. A concatenation operation is defined for character data.
5. Allow character expressions as I/O unit, format reference in formatted READ/WRITE statements.

VII. Binary Data type extensions\*:

VIII. Subsets of the language:\*

\* NOT APPROVED BY FULL 73J3 COMMITTEE.



本 PDF ファイルは 1965 年発行の「第 6 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 サイトの [https://www.ipsj.or.jp/topics/Past\\_reports.html](https://www.ipsj.or.jp/topics/Past_reports.html) 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」を掲載して、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。

#### 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](mailto: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>