JIS FORTRAN の文に関する非局所的文法について*

菅　　忠　義**

Abstract

A classification method of grammatical rules for programming languages and a general standpoint of description method for reference languages are proposed. Then the contents of Section 10, JIS C 6201 are reviewed from these viewpoints, and some problems in question are pointed out in tabular form. Furthermore several problems which play an important role in order to reorganize the contents of this section are carefully examined. Finally a reorganization of the section from this standpoint is shown in a concrete form.

The principal purposes of the reorganization are:

(1) to give a consistent and complete nonlocal grammar of FORTRAN with respect to the class of statements,

(2) to realize more extensive codification and regularization of FORTRAN practice within the extent of Section 10, JIS C 6201,

(3) to achieve logical clarification of the contents of the section.

ま　え　が　き

この論文の構成は，つぎのようになっている．

I. において，言語体系についての一つの立場を示し，**II.** において，この立場から JIS FORTRAN[1] の〈**第10節**〉の特性づけと，その問題点を示し（ただし，なぜ問題になるかは詳論しないでどういう種類の問題があるかだけを表で示す），かつ，そのうちの特に重要な問題点について例示的に解決の方針を論じ，**III.** の構成の根拠の一部を示す．**III.** において，**I.** の立場から，**II.** の方針に沿って具体的に〈**第10節**〉の体系化を与える．

また，ここで問題点としてあげるものは，**I.** の立場からの体系についての問題点であり，ISO FORTRAN[2] が陽に (explicitly)，また陰に (implicitly) 規定している，あるいは規定しようと意図したと思われる範囲内での問題点である．したがって，たとえば，random file，あるいは double complex type などの機能を付加すべきだというような問題点[3] は論じない．

ここで〈**第10節**〉を特に対象として選んだのは，JIS FORTRAN 全体について論ずることは，ページ数の制限で不可能であるためと，この節は在来の FORTRAN に関するマニアルや解説書にはなかったものであり，かつ一つのまとまった内容をもっているので，ページ数をあまり超過しないで一つの体系を具体的に示すことができると考えたためである．

以下では，JIS FORTRAN の節番号は〈 〉で，この論文の節番号は，原則的には［ ］でくくって区別する．ただし［ ］の中が **III.** の節番号であるときは **III.** は一般には記入しない．

I.　プログラミング言語の文法規則の分類と，言語体系の記述法

I・1　文法規則の分類

JIS FORTRAN の文法を考察するにあたって，まずプログラミング言語の文法上の規則を分類する一つの立場を述べる．すなわち文法 (syntax と semantics) 上の規則を，その言語の構成要素の集合である一つの文法上の**類に関して**，つぎのように分類する．

- **局所的** (local) 規則
- **非局所的** (nonlocal) 規則

文法上のある類に関して局所的規則とは，その文法上の類に属するものの形式を構成するための規則と，

* On Nonlocal Grammar of JIS FORTRAN with respect to the class of statements by S. Tadayosi Kan (Faculty of Science Gakushuin University)

** 学習院大学・理学部

その類に属するものの効果（意味）を規定する規則とであり，非局所的規則とは，その類の要素相互や，その類の要素を構成要素として含むような類とその要素に，またはそれら相互に関連した文法上の規則のことである．そこで局所的，非局所的という概念は，文法上のどのような類に着目するかに依存することになる．ここではFORTRANの“文”をその着目する類とする．文を着目する類としてとったことの根拠は，この類の理論的特性づけを行なってはじめて明らかとなることであるが，ここでは，それについて論ずることはしないで，単に後に示すように，JIS FORTRANの文法体系を理解するために便利であるという理由をあげるにとどめる．

I・2 言語体系の記述法

プログラミング言語の体系の記述法については，つぎの二つに分けて考えることができる．

（1） 形式上の個々の類やそれの効果（意味）を定義する方法．

（2） 具体的なプログラミング言語の全体系を規定するとき，(1) の意味での一つの定義法の採用を決定してもなお問題として残る記述の方法．

(1) については，たとえば文献 4) に示されているように，種々論じられているので，またここではJIS FORTRANの文法体系を考察するのが目的であるので，むしろ (2) について論ずる．

(2) については，つぎのような問題がある．

（i） 全体系を記述する場合，章 (chapter) や節 (section) の構成をどのようにするかということは，言語体系というものを，単なる文法規則の集合としてだけ考えるのでなく，文法規則の相互関係を考慮して，有機的にそれらを組織したものとして考える立場をとるときに問題になる．たとえば原子の単なる集合は，それらが相互に規則的に結合してできる分子あるいは結晶とは異なるものであり，また集合として同じ原子からなるものでもその相互の結合の仕方で，分子あるいは結晶としては異なるものとなるように，文法規則の各箇条を原子とみなし，全体系を分子あるいは結晶に相当するものとみなすのである．この立場は，論理の明快性に関する精神的満足感を与えるだけでなく，その言語を理解しやすくするという実用的効用をも与えるものである．全体系は，章や節の構成で組み立てられるものであるが，それについては以下の（ii），（iii），（iv），（v），（vi），（vii），（viii）程度以上の一般的法則は，現時点では存在しないので，体系のあり方というものは，一つの言語，またはその部分に対して，実際に構成して具体的に示す以外にない．

（ii） 基準言語 (reference language) を規定する場合，特にそれが**規格**である場合，そこに記述される規則は，公理的なものだけに限り，それらの規則から論理的に推論されるもの，たとえば定理に相当するものは，本文とはしないで“解釈”として与えるのがよいであろう（**[III]** では**備考**として述べてある）．

（iii） 言語体系の本文は，（ii）に述べたように公理的なものであるという立場をとる場合，数学での公理に要求される条件，すなわち**無矛盾性**，**完全性**，**独立性**に相当する条件を満足するようにすべきである．ここで無矛盾性は，いうまでもないことであるので論じないが，完全性に相当する条件とは，規定しようと意図した事項を過不足なく，本文で規定した条項で規定しているということである．この条件は FORTRAN のように，existing practice が先行して，後から reference language を規定しようとする場合には，existing practice のどこまでを成文化するかという意味で問題になる．また独立性に相当する条件とは，文法に含まれる規則相互に重複や冗長性がないということである．

（iv） 自然言語で文法を定義する場合，日常用いられる用語に，その文法中で特殊な意味を与える場合には，それがなるべく明白になるように記述すべきである．

（v） 体系中の規則相互にバランスがとれていることが必要である．ここでバランスをとるとは，ある事項を規則として成文化するならば，それと同程度の事項は，規則として成文化すべきであるということである．（iii）で述べた完全性は，範囲についてのものであり，ここでのバランスは程度についてのものである．

（vi） 一つの章や節は，論理的に統一された内容をもつべきである．この条件は（i）に述べた立場からの必要最低限の条件である．

（vii） 章や節を分けるにあたって，**I・1** に述べた文法規則の分類の立場からすると，文に相当する効果をもつ文法上の類に関して局所的文法と，非局所的文法とを分けて記述すべきである．これも（i）に述べた立場に沿う一つの手段である．

（viii） 章や節の内容は，なるべく箇条書きにして，各箇条は互いに独立であるようにする．これは（i）に述べた原子論的立場，および（ii），（iii）の条件を実現するための一手段である．

II. JIS FORTRAN の考察

ここでは I. に述べた立場から JIS FORTRAN を考察するが，II・1 においては JIS FORTRAN の体系の中での〈第10節〉の特性づけを行ない，II・2 において〈第10節〉の問題点を指摘し，さらに II・3 において I・2 の立場に立って，III. に示す体系を導くうえで，II・2 に指摘した問題点のうち特に重要なものについて解決の方針を簡単に論ずる．

II・1 JIS FORTRAN〈第10節〉の特性

JIS FORTRAN〈第9節，第10節〉は，それより前の節が，主として“文”に関して局所的文法を規定しているのに対して，非局所的文法を規定しているという特性をもつ．実際〈第9節〉は，文を要素として含む syntax class であるプログラム部分，プログラム本体，副プログラム，主プログラム，実行可能プログラムなどの syntax が規定してあるが，これらは文に関して非局所的 syntax である．〈第10節〉では syntax class としてはなにも規定されていないが，文を構成要素として含む syntax class である実行可能プログラムやプログラム単位の中において，文や文の構成要素（たとえば変数や配列など）に対して付加される文に関して非局所的 syntax と，実行可能プログラム中の手続きおよびデータと引数の定義について，文に関して非局所的 semantics とが規定してある．

〈第9節〉，特に〈第10節〉は，I・2 に述べた意味での言語体系の整備という方向への approach を実現したものであり，それが，これらの節の最大の特性である．〈第10節〉の内容は，ASA の technical subcommitte X3・4・3 が，1962～1966にわたって作業をした成果であり，FORTRAN のプログラミング言語としての発展史上画期的業績であると思われる．

II・2 〈第10節〉の問題点

II・1 に述べたように，〈第10節〉は，ASA X3・4・3 による画期的成果ではあるが，精細な吟味をすると，もちろん多くの問題点がある．以下に〈第10節〉について，I. に述べた立場からの見解として問題があると考えられるものを第1表に示す．ただし各問題点に関して，どのような意味でそれが問題になるかということをいちいち論ずることはしないで，I・2 の立場からどういう種類の問題があるかを表で示すにとどめる．ここで問題の種類をつぎのような記号で表わすことにする．

a：論理的に欠陥がある（I・2 の（iii）参照）．

b：節の内容が雑然として体系の明快性を損なっている（I・2 の（i），（vi）参照）．

c：節の位置に問題がある（I・2 の（i）参照）．

d：完全性に相当する条件についての問題（I・2 の（iii）参照）．

e：成文化のバランスの問題（I・2 の（v）参照）．

f：独立性に相当する条件や冗長についての問題（I・2 の（iii），（vi）参照）．

g：用語の定義に関する問題（I・2 の（iv参照））．

h：用語の用い方に関する問題，および箇条の表現の仕方に関する問題．

i：規定の内容が不十分である．これは，主原因は **a, d, e** によるが，〈第10節〉中のその箇条に多少の補足的修正を加えることで解決しうる問題．

第1表において〈第10節〉の節番号のある行と，上記の記号のある列との交点の所に［　］として節番号が記してある場合は，〈　〉節にその記号に相当する問題があり，それは［　］節において取り扱われ，それが III. のものであるときはその節で解決されていることを表わし，それが II. のときはそこで問題が論じられ，かつ解決の方針が示されていることを表わす．ここで第1表は，［III］と〈第10節〉との相互関係を理解する補助手段として示したものであり，すべてをあげつくした完全なものではない．

II・3 文に関して非局所的文法の体系化の方針

II・2 に問題点とその種類を示したが，これらの問題点を解決して，〈第10節〉の内容を I・2 の立場から新しく体系化することを III. で試みる．なにゆえに III. のような体系が導かれたかという根拠は，〈第10節〉と II・2 および III. を精査することによって理解されるであろうが，それを助ける補助手段として，以下に III. の体系を導き出すために重要と考えられる問題を含む〈第10節〉中の二つの節を選んで例示的に論ずる．

（1）〈10・2〉の内容と問題点およびその解決の方針

（i）　ここでは定義の種類について述べているが，“定義”という用語について，I・2 の（iv），II・2 の **g** に関する問題がある．すなわちその種類の分け方のよりどころをもっと明白にし（[2・1] 参照），さらに I・2 の（i），（vi）の立場から**データの定義**（[2・3] 参照）と，**手続きの定義**（[2・2] 参照）とに分けて記述するほうが明析になる．

（ii）　〈10・2〉で“数値に対して**第1階**の定義と，

第 1 表　JIS C 6201〈第 10 節〉の問題点

JIS の節 ＼ 問題の種類		a	b	c	d	e	f	g	h	i
〈10·1〉	〈10·1〉		[1·1]			[1·1·2 (ii)] [1·1·3 (ii)]				
	〈10·1·1〉	[1·2·3 (i)]	[1·2]						[2·1 (3)] [1·2·3 (ii)]	[1·2·3 (i)]
	〈10·1·2〉		[1·3] [2·4·2]			[1·3]	[1·3·1]	[2·1 (3)]	[1·3·2 (2) (ii)]	[1·3] [2·1 (2)]
	〈10·1·3〉		[1·3] [2·4·2(Ⅲ)(vii)]							[1·3·2 (1) (v)]
	〈10·1·4〉		[1·3] [2·2 (3)] [2·4·2 (Ⅲ)]							
	〈10·1·7〉		[1·3]				[1·2·3]			
	〈10·1·8〉		[1·3]							
	〈10·1·9〉		[1·3]							[1·3·1 (2) (i)]
	〈10·1·10〉		[1·3]							
〈10·2〉	〈10·2〉	[Ⅱ·3 (1)]	[2·1] [2·2] [2·3]		[Ⅱ·3 (1)] [2·3·2]	[Ⅱ·3 (1)] [2·3]		[Ⅱ·3 (1)] [2·3]		[2·3·2] [2·3]
	〈10·2·1〉		[2·2 (1)] [2·2 (2)] [2·2 (3)]							[2·2 (1)] [2·2 (2)] [2·2 (3)]
	〈10·2·2〉	[Ⅱ·3 (2)] [2·4·2]	[Ⅱ·3 (2)] [2·1 (1)] [2·4]					[2·1 (2)] [2·1 (3)]		[2·1 (1)] [2·1]
	〈10·2·3〉		[2·1 (2)]					[2·1 (2)]		
	〈10·2·4〉		[2·3·2 (Ⅰ)(A)] [2·3·2 (Ⅰ)(B)] [2·3·2 (Ⅰ)(C)]							
	〈10·2·5〉		[2·3·2 (Ⅰ)(A)] [2·3·2 (Ⅰ)(B)] [2·3·2 (Ⅰ)(C)]							
	〈10·2·6〉		[2·4] [2·5]							
	〈10·2·7〉						[2·3·2 (Ⅰ)(2)(A)(c)]			
	〈10·2·8〉			[Ⅲ] から 除外						
	〈10·2·9〉		[2·3·2 (Ⅰ)(2)]							
〈10·3〉	〈10·3〉		[2·5]						[2·5·1 (i)]	[2·5·1 (iii)]

第2階の定義の2種類があるものとする．以下特に述べない限り，変数や配列要素に適用された確定または不定という術語は，第1階の定義での確定または不定を示す”と述べている．この場合，たとえば〈10・2・3の（1）〉の記述と比べると **II・2** の **a**，**i** に関する問題を生ずる．それは〈10・2・3 の（1）〉では“論理代入文”も含まれているからである．そこで“数値と論理値”に対して第1階の定義を定義しておくか，あるいは，“……確定または不定という術語は，……”という断り書きを“……確定または不定という術語は，それが数値であるときは，第1階の定義での……”とする必要がある．しかしこの問題は，さらにデータの定義全体の中で検討して解決すべき問題である（(v)，[2・3] 参照）．

(iii)　値の定義について変則的定義をあげ，また JIS C 6201 の解説〈3・2〉でそれについて「**注**」をつけているが，文字定数を値として与えることについては，なにもふれていない．ISO FORTRAN の精神が existing practice を重視し，それを成文化して，言語としての体系をととのえようというものであるとすると，“定義”に関しては，**II・2** の **d**，**e**，**i** についての問題があり，**I・2** の（i）の立場から，もっと整理して明快にする必要がある．

(iv)　FORTRAN では伝統上，定数および変数や配列と型との対応に，existing practice を含めて問題がある．ISO FORTRAN では型の種類については伝統を守るという条件の下で，semantics の成文化に積極的態度をとり，第1階，第2階の定義というものを導入したと思われるが，ここに **II・2** の **d**，**e**，**i** の問題がある．

(v)　(iii) の問題を，syntax を変えないという条件の下で，“値をもつデータの定義”を，**I・2** の（i）の立場に沿って明確にする一つの方針を以下に示す（[2・3] 参照）．ここでデータとは，〈5・1〉によれば，定数，変数，配列，配列要素や共通ブロックのことであると解すべきである（〈5・1〉には **II・2** の **g** の問題がある）．以下では〈5・1〉に“共通ブロック名以外のデータ名は値をもつものとする”という箇条を入れるものと仮定しておく．値をもつデータの定義の種類をつぎのようにする（[2・3 の（1）] 参照）．

第1種の定義：許されている値に対応する型がある場合．

第2種の定義：その値に対応する型がない場合．

第3種の定義：その値と型が，定数以外対応しない場合．

すなわち，

第1種の定義とは，数値や論理値を与える場合，

第2種の定義とは，文の番号を値として与える場合，

第3種の定義とは，文字定数を値として与える場合，

である．

さらに第1種の定義において，数値を与える場合を

第1階の定義　　**第2階**の定義

に分け，前者は値が数値そのものとしての効果をもつ場合とし，後者は多少異なった効果をもつ場合とする．ここで効果が異なるということは，コンパイラ作製の際の取り扱い方が異なりうるということを意味する．

(vi)　JIS では配列がどういうときに定義されるか，どこにも規定されていない（**II・2** の **d**，**e**，**i** の問題）ので，それを明確にする必要がある（[2・3・2] 参照）．

(vii)　〈5・1・1〉によれば“定数は，名前そのものが値を定義するデータとし，再定義できないものとする”としている．また〈10・1・2〉に DATA 文についての記述中に“**初期値**”という用語がある．ここには **II・2** の **g**，**h** の問題があるが，これは FORTRAN の定数という形を，“**初期値定義**”の一種とみることによって解消しうる（[2・1 の (2)，2・3・1] 参照）．なお〈5・1・1〉の内容として扱うべき“定数”については問題があるが，ここでは論じない．

(viii)　手続きの定義と手続き名の定義が規定されるならば，共通ブロックの定義と共通ブロック名の定義ということも **II・2** の **d**，**e**，**i** 意味で問題になるが，この規定を付加することは，他に影響を与えることなく，容易に行なえるので，ここでは省略することにする．

（2）〈10・2・2〉の内容の問題点およびその解決の方針

（i）〈10・2・2〉には結合と結合による定義について記述してあるが，ここには **II・2** の **g**，**h** の問題がある．**結合**ということは，むしろ定義の方法の一種とみるべきである（[2・1 の (2)，(3)] 参照）．

（ii）　この節は，内容が雑然としていて **II・2** の **b** の問題がある．整理統合して明快にする必要がある（[2・1 の (1)，(2)，(3)]，[2・4] 参照）．

（iii）　引数の結合ということは，引数の定義という立場から考え，その定義の方法として，結合という手段がとられるとしたほうが明確になる．また引数としては，共通ブロック名以外のデータ名が許されている

から，引数の定義をデータの定義に還元して規定しうる．しかしデータの定義と引数の定義とを，別に扱わなければならないのは，引数としては式や外部手続き名が許されているからである．そこで引数の定義という項目を別に設けて，これを明確に規定する必要がある（[2・1 の (1)，2・4] 参照）．

（iv） この節では“引数の結合は，対応する実引数と仮引数の型が同じ場合だけ行なわれる”と記述してあるが，この場合文字定数が実引数のとき，結合が行なわれないことになり，ここには II・2 の **a**，**d**，**e**，**i** の問題を生ずる（[2・4・2] 参照）．

（v） この節で“同じプログラム単位の中の COMMON 文中に現われた二つのデータの間の結合はないものとする”という記述があるが，これは〈7・3 の(2)〉へ移すほうがよいであろう．

III. FORTRAN 水準 7000 の文に関する非局所的文法

ここでは II・2 で指摘した JIS FORTRAN の〈**第10節**〉の問題点を I・2 の立場に沿って解決した一つの体系を具体的に示す．ただし，ここでは，文に関して非局所的文法のうちつぎの2項目に限る．

（1） プログラム単位内および実行可能プログラム内の英字名．

（2） 実行可能プログラム中の手続きおよびデータと引数の定義．

なお，〈**第10節**〉にはなく，III で特に付加された，あるいは修正された項目や箇条は記号“†”が，〈**第10節**〉にはないが JIS C 6201 の他の節にある箇条には記号“○”が付けてある．

1.†プログラム単位内および実行可能プログラム内の英字名と類

プログラム単位内および実行可能プログラム内の英字名と類は，以下による．

1·1† 英字名，類，および類に属するということ

英字名，類，および英字名が類に属するということは，それぞれ [1·1·1，1·1·2，1·2·3] による．

1·1·1 英字名

（i） 英字名は1～6個の英数字からなり，その最初のものは英字でなければならない．

（ii） 書式の欄記述子，あるいは GOTO，READ，FORMAT など文を識別するために使われている文字の列は，英字名ではなく英字名の初めの部分でもないものとする．

1·1·2† 類

定数以外のデータと手続きは，プログラム単位内で，それぞれつぎのような**類**に分けられるものとする．

第 I 類 配列や配列要素
第 II 類 変数
第 III 類 文関数
第 IV 類 組込み関数
第 V 類 外部関数
第 VI 類 サブルーチン
第VII類 外部手続き（そのプログラム単位内では，サブルーチンとも外部関数とも区別できない外部手続き）
第VIII類 ブロック名

1·1·3† 英字名が類に属するということ

英字名や添字のついた英字名が，プログラム単位内で一つの**類に属する**とは，それらがその類に対応するFORTRAN語の構成要素としての効果をもつことを意味するものとする．

1·2† 英字名と類の関係

1·2·1† 英字名と類に関する原則

（i） 英字名は下記 [1·2·2] の場合を除き，一つのプログラム単位において，常に一つの類にだけ属さなければならない．

（ii） 英字名と類の関係は，下記 [1·2·3] の場合を除き，一つの実行可能プログラム内の異なったプログラム単位では，制限はない．

1·2·2† プログラム単位内において，二つの類に属しうる英字名

（i） あるプログラム単位において，第Ⅷ類に属する英字名は，また第Ⅰ類，第Ⅱ類，あるいは第Ⅲ類のいずれか一つに属してもよい．

（ii） 関数副プログラム，すなわち FUNCTION 文を含むプログラム単位においては，FUNCTION 文中の FUNCTION という語につづく第Ⅴ類の英字名は，第Ⅱ類にも属さなければならない．

1·2·3† 英字名と類に関する実行可能プログラム中のプログラム単位間における制限

（i）† 実行可能プログラムを構成するあるプログラム単位内で，英字名がいったん第Ⅴ類，第Ⅵ類，または第Ⅶ類に属すると，その実行可能プログラム内の他のプログラム単位では，その英字名はこれらの類のうちの最初に属した以外の類に属するようなことがあってはならない．

（ii） 第Ⅶ類に属する英字名は，プログラム単位内で局部的にだけ存在するものとし，実行可能プログラム内では，第Ⅴ類または第Ⅵ類に属さなければならない．

備考 [1·2·1 の (ii)] により，一つの実行可能プログラム内のあるプログラム単位内で，英字名が第Ⅳ類として現われても，同一実行可能プログラムの他のプログラム単位では，その英字名が別の類に属してもよい．

1·3† 英字名と類に関する条件

プログラム単位において，文中に現われる英字名が属すべき類を判定する条件，およびプログラム単位内，および実行

可能プログラム内における各類についての，それが満たすべき条件は，以下による．

1·3·1 英字名が各類に属するための条件

この節では，各類について英字名がその類に属するための必要十分条件を示す．

(1)† 第 I 類に属するための条件

英字名が宣言子名として現われた場合に限り，その英字名は，そのプログラム単位で第Ⅰ類に属する．

備考 (i) つぎのいずれかの条件を満たす英字名は，そのプログラム単位で第Ⅰ類に属しうる．

(α) ブロック名を除き，COMMON 文中に現われた英字名

(β) EQUIVALENCE 文中に現われた英字名

(γ) 型宣言文中に現われた英字名

(δ) DATA 文中に現われた英字名

(ii) EXTERNAL 文中に現われた英字名は，そのプログラム単位で第Ⅰ類に属さない．

(2) 第 II 類に属するための条件

英字名が，つぎの3条件をすべて満たして現われる場合に，そのプログラム単位内で第Ⅱ類に属する．

(i)† 第Ⅰ類，第Ⅵ類，第Ⅷ類のいずれにも属さない

(ii) FUNCTION 文の FUNCTION という語の直後につづく場合を除いて，その英字名の後に左かっこがつくことはない．

(iii) 第Ⅷ類以外で少なくとも1度は現われる

備考 (i) つぎのいずれかの条件を満たす英字名は，そのプログラム単位で第Ⅱ類に属しうる．

(α) ブロック名を除き，COMMON 文中に現われた英字名

(β) EQUIVALENCE 文中に現われた英字名

(γ) 型宣言文中に現われた英字名

(δ) DATA 文中に現われた英字名

(ii) EXTERNAL 文中に現われた英字名は，そのプログラム単位で，第Ⅱ類に属さない．

(3)† 第 III 類に属するための条件

英字名は，つぎの3条件をすべて満たす場合に，そのプログラム単位内で第Ⅲ類に属する．

(i) EXTERNAL 文中に現われず，第Ⅰ類に属さない．

(ii) 型宣言文中の場合を除き，その名前が現われるときは常に左かっこが直後につづく．

(iii) その英字名に対して，そのプログラム単位内に文関数定義文がある．

備考 型宣言文中の英字名は，そのプログラム単位で第Ⅲ類に属しうる．

(4)† 第 IV 類に属するための条件

英字名は，つぎの4条件をすべて満たす場合に，そのプログラム単位内で，第Ⅳ類に属する．

(i) EXTERNAL 文中に現われず，第Ⅰ類にも，第Ⅲ類にも属さない．

(ii) その英字名は，〈8·2 の表3〉の中の英字名の欄に現われる．

(iii) その英字名は，〈8·2 の表3〉の中に定義されているのと異なる型として型宣言文中で宣言されていない．

(iv) 型宣言文中の場合を除き，その英字名が現われるときには常にその直後にかっこでくくられた実引数の並びがつづく．

(5)† 第 V 類に属するための条件

英字名は，つぎの条件のいずれかを満たす場合に，そのプログラム単位内で第Ⅴ類に属する．

(i) FUNCTION 文の FUNCTION という語の直後につづく．

(ii) 第Ⅰ類，第Ⅱ類，第Ⅳ類または第Ⅵ類には属さず，かつその英字名が現われるときは，型宣言文中，EXTERNAL 文中，および実引数の場合を除いて，常にその直後に左かっこがつづいている．

備考 (i) つぎのいずれかの条件を満たす英字名は，第Ⅴ類に属しうる．

(α) EXTERNAL 文中に現われた英字名

(β) 型宣言文中に現われた英字名

(ii) つぎのいずれかの条件を満たす英字名は，第Ⅴ類に属さない．

(α) COMMON 文中に現われた英字名

(β) EQUIVALENCE 文中に現われた英字名

(γ) DATA 文中に現われた英字名

(6)† 第 VI 類に属するための条件

英字名が，つぎの条件のいずれかを満たす場合，そのプログラム単位内で第Ⅵ類に属する．

(i) SUBROUTINE 文中の SUBROUTINE という語の直後につづく

(ii) CALL 文中の CALL という語の直後につづく

備考 (i) EXTERNAL 文中に現われた英字名は，第Ⅵ類に属しうる．

(ii) 型宣言文中に現われた英字名は，第Ⅵ類に属さない．

(7)† 第 VII 類に属するための条件

英字名が，つぎの3条件をともに満たす場合，そのプログラム単位内で，第Ⅶ類に属する．

(i) EXTERNAL 文中に現われる．

(ii) 外部手続きの実引数として現われる．

(iii) そのプログラム単位では，上記以外には現われない．

備考 型宣言文中に現われた英字名は，そのプログラム単位で第Ⅶ類に属さない．

(8) 第 VIII 類に属するための条件

英字名が COMMON 文中のブロック名として使用された

場合だけ，それはそのプログラム単位内で第Ⅲ類に属する．

備考 COMMON 文中の奇数番目の斜線とそのつぎの斜線の間にはさまれて現われた場合だけ，第Ⅲ類に属する[3]

1·3·2† プログラム単位内，および実行可能プログラム内における英字名の現われ方についての条件

(1)† プログラム単位内における条件

英字名は，一つのプログラム単位内において，つぎの条件を満たさなければならない．

(i) 英字名は，一つのプログラム単位内で，宣言子名として2度以上現われてはならない．

(ii) 英字名は，一つのプログラム単位内の型宣言文に2度以上現われてはならない．

(iii) 英字名は，一つのプログラム単位内の EXTERNAL 文に2度以上現われてはならない．

(iv) 英字名を第V類として使用するプログラム単位では，その英字名は，[**1·3·1** の (5) の (ii)] の形で，少なくとも1度現われなければならない．

(v) 一つのプログラム単位において，配列を識別する英字名が現われた場合には，その配列名の直後には，添字がつかなければならない．ただしつぎの場合は除く．

(α) 入出力文の並びの中

(β) 仮引数の並びの中

(γ) 外部手続きを引用する実引数の並びの中

(δ) COMMON 文の中

(ε) DATA 文の中

(ζ)† READ 文と WRITE 文の (u, f) の中

(2)† 実行可能プログラム内における条件

第Ⅱ類の英字名や，第Ⅰ類の添字のついた英字名は，一つの実行可能プログラム内で，DATA 文中に2度以上現われてはならない（[**2·5·2** の (1) の (i)] 参照）．

2.† 実行可能プログラムの中の手続きおよびデータと引数の定義

2·1† 定義の種類，および結合

FORTRAN 語における定義の種類，および結合は，以下 [(1), (2), (3)] による．

(1)† 定義の対象による分類

定義は，定義する対象によって，つぎのように分類するものとする．

- 定義
  - 手続きの定義
  - 値をもつデータの定義
    - 定数の定義
    - 変数および配列要素と配列の定義
  - 引数の定義

ここで値をもつデータとは，定数，変数，配列要素，配列のこととする．

(2)† 定義の方法による分類

定義は，定義する方法に関して，つぎのような場合があるものとする．

- 定義の方法
  - 手続きの定義の方法
    - FORTRAN 語による定義
      - 文関数定義文による定義
      - 副プログラムによる定義
    - FORTRAN 語以外による定義
  - 値をもつデータの定義の方法
    - 初期値定義
      - 定数による定義
      - DATA 文による定義
    - 間接的定義
      - COMMON文による結合
      - EQUIVALENCE 文による結合
    - 直接的定義

ここで**初期値定義**とは，実行可能プログラムの実行に際して，初期状態でのデータの定義のことをいい，定義の方法としては定数による場合と，DATA 文によって行なわれる場合とがある．**間接的定義**とは，結合による定義のこととし，**直接的定義**とは，結合によらない，実行文の実行によって行なわれる定義のこととする．

(3)† 結合

結合の定義と方法は，つぎによる．

(i)† 結合の定義：結合とは，定数や変数や配列要素または外部手続き名が，同一の記憶場所をもつこととする．

(ii) 結合の方法：結合は，つぎによるものとする．

(α) COMMON 文による結合

(β) EQUIVALENCE 文による結合

(γ) 引数による結合

上記のものを組み合わせて，いくつかのデータを結合することができるものとする．

(iii)° COMMON 文と EQUIVALENCE 文による結合：COMMON 文と EQUIVALENCE 文による結合は，つぎの条件の下に，〈**7·3** の (2), (3)〉による．

(α) そのとき結合された変数や配列要素は，それらが同じ型のときだけ同じ値になるものとする．

(β) 二つの記憶単位を占める要素が一つの記憶単位を占める要素に対応した場合には，後者は前者の記憶場所のうち最初の記憶単位を共有するものとする．

2·2 手続きの定義

手続きの定義は，つぎによる．

(1)† 手続きの定義

手続きが実行可能プログラム内で定義されているとは，その実行可能プログラム中で，その手続きが引用されるとき，実引数と仮引数の条件が満たされるならば，つぎの条件[(2)]の下に，処理系において，その手続きを実行しうる手段が与えられていることとする．この場合その手続きの英字名は，その実行可能プログラム内で定義されているとする．

(2)† 実行可能プログラム内で定義されている手続き

手続きは，FORTRAN 語により表わされている場合と，FORTRAN 語以外により表わされている場合とがあるもの

とする．それぞれの場合について，手続きが定義されているとは，以下による．

（i）† FORTRAN 語により表わされた手続きは，文関数定義文か，副プログラムによって記述されなければならない．それらの定義は，つぎによる．

（α）† 文関数定義文による場合：〈8・1・1〉の条件を満たすとき，その定義文が現われるプログラム単位を含む実行可能プログラム内で，その手続きは定義されているものとする．またその手続きは，そのプログラム単位内でも定義されているものとする．

（β）† 副プログラムによる場合：〈8・3・1, 8・4・1, 8・5〉の条件を満たすとき，この副プログラムを含む実行可能プログラム中で，その手続きは，定義されているものとする．

（ii）† FORTRAN 語以外により表わされた手続きは，つぎによる．

（α）基本外部関数，組込み関数は，それが現われる実行可能プログラム内で，常に定義されているものとする．

（β）† それ以外でも，処理系において，実行可能プログラム中で，FORTRAN 語の文法に反しない，使用者が期待する効果を生ずる実行の手段が与えられている場合，その手続きは，その実行可能プログラム内で定義されているものとする．

（3） 手続きの引用に関する条件

手続きが引用されるとき，つぎの条件が満たされていなければならない．

（i）† 手続きが定義されている場合，その引用は引数の数さえ合えば，〈8・1・2, 8・2, 8・3・2, 8・3・3, 8・4・2〉の条件の下に，実行可能プログラム中に現われることが許される．外部関数は，関数の型も一致しなければならない．

（ii） 実行可能プログラムの実行中に，ある手続き副プログラムが，その手続きの RETURN 文が実行されないうちに，自分自身が引用されてはならない．

(iii) 一つの文の中で，関数副プログラムが2回以上同じ実引数の並びで引用される場合には，その副プログラムの実行の結果は，その文の評価順序とは関係なく同一でなければならない．

(iv)° 式の中に現われる関数を評価するとき，その式やその式を含む代入文や CALL 文の他の要素の値を変更してはならない．

2・3† 値をもつデータの定義

（1）† 値をもつデータの定義と再定義，および確定と不定

実行可能プログラムにおいてプログラムの実行の進行につれて，データに値を与えることを，**データを定義**するという．

値が定義されることを**確定**，定義されていないことを**不定**という．確定あるいは不定であるデータが，改めて確定となるとき，**再定義**されたという．

以下では，データとは，値をもつデータ（[2・1 の (1)] 参照）のこととする．

（2）† 値をもつデータの種類による定義の分類

データの定義は，データの種類によって，つぎのように分けられるものとする．

データの定義 ┬ 定数の定義 [2・3・1]
　　└ 変数および配列要素と配列の定義 [2・3・2]

ここで，定数，変数，配列要素，配列が引数として用いられている場合は [2・4] による．

（3）† 値をもつデータの型と値の対応による定義の分類

変数および配列要素と配列の定義は，データの型と値の対応関係から，つぎのように分類されるものとする．

第1種の定義：値に対して対応するデータの型がある場合で，値として数値や論理値を与える場合とする．

第2種の定義：値に対して対応するデータの型がない場合で，値として文の番号を与える場合とする．

第3種の定義：値に対して対応する型のデータが定数以外ない場合で，値として文字定数を与える場合とする．

さらに第1種の定義において，値が数値の場合，**第1階の定義**と**第2階の定義**があるものとする．第2階の定義は，整数型の変数が添字式または計算形 GOTO 文の中に現われる場合に適用されるものとし，第1階の定義は，それ以外の場合に適用されるものとする．

種の異なる定義での確定は，互いに排他的とする．

2・3・1† 定数の定義

定数は，実行可能プログラムにおいて，初めから，その名前が表わす値をもつものと定義されているとする．これを定数による初期値定義という．

定数による初期値定義は，再定義されることはないものとする．

2・3・2† 変数および配列要素と配列の定義

配列は，そのすべての配列要素が定義されたとき，定義されるものとする．

（I）† 第1種の定義

（1）† 第1種の定義での確定と不定

第1種の定義の確定と不定は，初期値定義および間接的方法と直接的方法に関して，以下 [(A), (B), (C)] によるものとする．

数値に関して，第1種での確定，不定を 第1階での確定，不定とする．

(A)† 初期値定義による確定と不定

（a）† 確定となる場合

変数と配列要素は，それらの名前が DATA 文の中で，それらと同じ型の定数と対応している場合にのみ，最初から確定となるものとする．

（b）† 不定となる場合

最初から確定となっていない変数と配列要素は，すべて

主プログラムの最初の実行文を最初に実行するときには，不定となるものとする．

(B)† 実行文の実行による確定と不定

(a)† 確定となる場合

(i) 算術代入文または論理代入文の実行が完了すると，等号の左側のデータは確定となる．

(ii) 入力文の実行の際，入力装置から値を入れられたデータは，そのとき確定となる．そのデータと結合している同じ型のデータは，その入力文の実行が完了したとき，確定となる．

(iii) DO 文の実行が開始されると，その制御変数が確定となる．

(iv) DO 形並びで指定された動作が開始されると，その制御変数が確定となる．

(b)† 不定となる場合

(i) DO が満足されたとき，その制御変数は不定となる．

(ii) DO 形並びで指定された動作が完了すると，その制御変数は不定となる．

(iii) ASSIGN 文の実行が完了すると，その文中にある整数型の変数は不定となる．

(iv) 文に評価されない因子が含まれており，かつこの因子が関数の引用を含んでいる場合には，この引用の実行中に確定となるはずのすべてのデータは，この因子を含む式の評価が完了したとき，不定となる．

(C)† 結合による確定と不定

(a)† 確定となる場合

(i) ある型のデータが確定となったとき，それと結合しているデータで，それと同じ型のデータはすべて確定となるものとする．

(ii) 無名共通ブロック内のデータ：無名共通ブロック内のデータは前述の［(A)，(B)］により，1度確定となると不定となるまで，確定のままとする．

(iii) 名前付共通ブロック内のデータ：名前がプログラム単位内で，ブロック名（第Ⅷ類）として現われているとき，そのプログラム単位は**名前付共通ブロックの名前を含む**という．主プログラムまたは引用された副プログラムが，名前付共通ブロックの名前を含む場合は，その名前付共通ブロック内のデータおよびそれと結合しているデータは，1度確定となると不定になるまで確定のままとする．

(iv) 共通ブロックに入らないデータ：前述の［(A)，(B)］により，1度確定となると不定となるまで確定のままとする．

(b)† 不定となる場合

(i) あるデータが確定となったとき，それと結合していてそれと異なる型のデータは，不定となる．

(ii) あるデータが不定となったとき，それと結合していてそれと同じ型のデータは不定となる．

(iii) 名前付共通ブロック内のデータ：初期値確定となっている定数以外のデータは，再定義されることによって，あとで不定となることができるものとする．

特に副プログラムが名前付共通ブロックの名前を含み，その副プログラムを直接または間接に引用しているどのプログラム単位にも，その名前が含まれていない場合は，その副プログラムの RETURN 文を実行することによって，そのブロック内のすべてのデータは，不定となるものとする．ただし初期値確定となっていて，まだ再定義されていないものは除く．

(iv) 共通ブロックに入らないデータ：［(C) の (a)の (iv)］のデータが副プログラムにある場合には，その副プログラムの RETURN 文の実行が完了すると，そのデータおよびそれと結合しているデータは，すべて不定となる．ただし初期値確定となっていて，まだ再定義されていないものは除く．

(2)† 第2階の定義での確定と不定

第2階の定義での確定は，第1階の定義での確定を必要とし，このほかに付加条件が必要となるものとする．この付加条件において，**文節**の概念を必要とする．

(A) 文節

(a) 文節の末端文

つぎの文を文節の端末文とする．

(α) DO 文

(β) CALL 文

(γ) GOTO 文

(δ) 算術 IF 文

(ε) STOP 文

(ζ) RETURN 文

(η) GOTO 文あるいは算術 IF 文に書かれている番号を持つ文に先行する実行文があれば，その最後の実行文

(ϑ) 等号の左側に整数型の変数がある算術代入文

(ι) READ 文で，その入力並びの中に，整数型の変数があるもの

(κ) 上記のどれかを含む論理 IF 文

(b) 文節の開始文

つぎの文を文節の開始文とする．

(α) プログラム単位の最初の実行文

(β) 文節の端末文につづく実行文があれば，その最初の実行文

(c) 文節

プログラム単位内における文節は，いくつかの実行文の集りとし，つぎによる．

(i) 文節の開始文を，一つの文節の始まりとする．文節の最初の文がまた文節の端末文でもある場合には．その文節は，その一つの文からなるものとする．

(ii) 文節が一つの文からなるとき以外は，文節は，開

始文とそれにつづく最初の端末文までのすべての実行文からなるものとする.

(B)† 文節の実行による第2階での確定と不定

(a)† 確定となる場合

(i) 入力並び中にない添字式，または計算形 GOTO 文中の整数型の変数：つぎの三つの条件がすべて満たされるとき文節の開始文を実行する際に，整数型の変数は第2階で確定となるものとする.

(α) 整数型の変数が結合によって，第2階での不定となっていない ([(C)] 参照).

(β) その変数がその文節内で，添字式または計算形 GOTO 文中に現われている.

(γ) その文節の開始文を実行するとき，その変数が第1階で確定となっている.

(ii) 入力文の並びの中の整数型変数：入力文の並びの中の整数型変数は，つぎのいずれかの場合，第2階で確定となるものとする.

(α) 整数型の変数が，入力文の並びの中に現われ，その文中のそれより後で，かつ下記 [(b) の (ii)]の条件が成立しない範囲で，添字の中に現われている.

(β) DO 形並びの制御変数が，第1階で確定となった.

(b)† 不定となる場合

(i) 上記の[(a) の (i)]による第2階での確定は，つぎのいずれかが起こるとき，第2階で不定となるものとする.

(α) その文節の端末文の実行が完了した.

(β) その変数が第1階で不定となるか，あるいは第1階で再定義された.

(ii) 上記[(a) の (ii)]による第2階での確定は，つぎのいずれかが起こるとき，第2階で不定となるものとする.

(α) その入力文の実行が完了した.

(β) その変数が第1階で不定となるか，あるいは第1階で再定義された.

(γ) プログラムの実行の進行が，そのDO形並びの範囲外に移った.

(C)† 結合による不定

整数型の変数が，第1階で直接的に再定義されると，それと結合しているすべての変数は，このプログラム単位の実行中は第2階で不定となり，これらの変数が結合づけによらずに第1階で直接的に再定義されるまで，この状態を保っているものとする.

備考 第2階の定義では，直接的方法によってのみ確定となる．また第1階で確定，したがって第1種で確定でも，第2階で不定の場合がある.

(II)† 第2種の定義

第2種の定義は ASSIGN 文中に現われる整数型の変数に対してだけ適用される.

(a)† 確定となる場合

ASSIGN 文の実行によって，その中に現われる整数型の変数は，第2種で確定となるものとする.

(b)† 不定となる場合

上記の確定は，第2種以外の定義で直接的あるいは間接的に再定義されるまで保たれ，そのとき第2種で不定となるものとする.

(III)† 第3種の定義

第3種の定義は，処理系で定められた特定の一つの型の変数や配列要素に対してだけ適用されるものとする．この型を第3種の定義に対応する型という．この確定と不定は，初期値定義および直接的方法と間接的方法に関して，つぎの条件で定まるものとする.

(A)† 初期値定義による確定と不定

(a)† 確定となる場合

第3種の定義に対応する型の変数と配列要素は，それらが DATA 文中に現われ，それに対応する定数が文字型であるとき，最初から第3種で確定となるものとする.

(b)† 不定となる場合

最初から第3種で確定となっていない，第3種の定義に対応する型の変数と配列要素は，すべて主プログラムの最初の実行文を最初に実行するときには，第3種で不定となるものとする.

(B)† 実行文の実行による確定と不定

(a)† 確定となる場合

入力文の実行の際，入力装置から文字型の値を入れられた第3種の定義に対応する型のデータは，そのとき第3種で確定となる．そのデータと結合している同じ型の他のデータは，その入力文の実行が完了したとき，第3種で確定となる.

(b)† 不定となる場合

実行文の実行により，第3種で確定となっているデータが，第3種以外で確定となったとき，それは第3種で不定となるものとする.

(C)† 結合による確定と不定

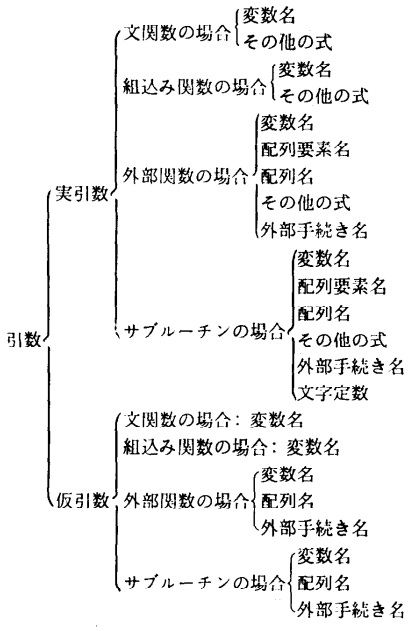
第1種の定義 ([(I) の (1) の (C)] 参照）の場合と同様とする．ただし各条件においての確定と不定は，第3種での定義についてのものとする．また [(I) の (1) の (C) の (a) の (i)] の“ある型”とは，第3種の定義に対応する型のこととする.

2·4† 引数の定義

2·4·1† 引数と引数の定義の種類

(1)† 引数の種類

引数は，実引数と仮引数とし，それぞれつぎのとおりとする.



(2)† 引数の定義の種類

引数の定義は，引数の種類によって，つぎのように分けられるものとする．

引数の定義 { 実引数の定義 / 仮引数の定義

2・4・2 引数の定義と結合

実行可能プログラムにおいて，プログラムの実行の進行につれて，引数に値を与えることを，**引数を定義する**という．引数の値が定義されることを**引数の確定**，また定義されていないことを**引数の不定**という．

(I)† 実引数の定義

実引数の定義は，つぎによる．

(A)† 手続きの実行の開始時における確定と不定

実行可能プログラムで，手続きの引用を実行する直前における，その実引数の確定または不定は，つぎによる．

(a)† 確定となる場合

(i)† 実引数が定数のとき，確定とする．

(ii)† 実引数が変数，配列要素，配列である場合，それが第1種あるいは第3種で確定のとき，確定とする．

(iii)† 実引数が上記以外の式である場合，その式を構成するデータが第1種で確定となっていて，かつその式の評価が可能であるとき，確定とする．

(iv)† 実引数が外部手続き名である場合，その外部手続きが，その実引数が現われる実行可能プログラムで定義されているとき，確定とする．

(b)† 不定となる場合

上記のいずれの条件も満たされないとき，不定とする．

(B)† 手続きの実行の進行に伴う確定と不定

実引数と結合している仮引数が，手続きの実行中，確定となったり不定となったりすると，その実引数は，それぞれ確定となったり不定となったりするものとする．

(II)† 仮引数の定義

仮引数の定義はつぎによる．

(A)† 手続きの実行開始時における確定と不定

(a)† 確定となる場合

仮引数は，それに対応する実引数が開始時に確定であり，かつそれと結合されるとき，確定となるものとする．

(b)† 不定となる場合

仮引数は，対応する実引数が開始時に不定であるとき，あるいは引数の結合が行なわれないとき，不定となるものとする．

(B)† 手続きの実行の進行に伴う確定と不定

仮引数が変数名あるいは配列名であるとき，それを含む実行可能プログラム中で，それがデータとして，第1種あるいは第3種で確定または不定となるとき，その仮引数は，それぞれ確定または不定となるものとする．

(III)† 引数の結合

外部手続きの引用が実行されるとき，実引数と仮引数の結合を引き起こすものとする．この結合は，この外部手続きを定義する副プログラム内で，仮引数が実行文や文関数定義文の中に現われたり，整合寸法として現われるすべての場所で，下記の条件[(i), (ii), (iii), (iv), (v), (vi), (vii), (viii)]のもとに行なわれるものとする．

この結合ののち，副プログラムの最初の実行文の実行につうるものとする．

またこの結合は，外部手続きの最初の実行文を実行するときから，その手続きの RETURN 文の実行が開始されるまで保たれ，RETURN 文の実行が開始されるとき解消されるものとする．

(i)† 実引数と仮引数の結合は，実引数が文字定数あるいは外部手続き名であるときを除き，対応する引数の型が同じ場合だけ行なわれるものとする．なお実引数が定数，変数名，配列要素名以外の式であるとき，下記 [iv] で定まる定数の型が実引数の型となるものとする．

(ii)† 実引数が文字定数のときは，対応する仮引数が，第3種の定義に対応する型の場合だけ，それを含む手続きを実行するとき結合が行なわれるものとする．

(iii)† 実引数が外部手続き名であるとき，対応する仮引数が外部手続き名である場合だけ，それを含む手続きを実行するとき，結合が行なわれるものとする．ただし関数名には関数名が，サブルーチン名にはサブルーチン名が対応するものとする．

(iv)† 実引数が定数，変数名，配列要素名以外の式のとき，仮引数と結合させられる直前に式を評価して，得られた値に対応する定数が，実引数となるものとする(〈8・3・2, 8・4・2〉参照)．

(v)† 実引数が配列要素名で，その添字式の中に変数を含む場合は，その実引数は，仮引数と結合させられる直前に添字式を評価して得られた値に対応する定数の添字式をもつ配列要素名が，実引数となるものとする．

(vi) 実引数と仮引数が配列名の場合，両方の配列の先頭の配列要素どうしが結合され，また実引数が配列要素名で，仮引数が配列名の場合，実引数の配列要素と仮引数の配列の先頭の配列要素とが結合されるものとする．以下配列要素関数で決められた順序にしたがって実引数の配列の配列要素と仮引数の配列の配列要素とが結合されるものとする．

(vii) 実引数が配列名か配列要素名である場合にかぎり，対応する仮引数は，つぎの条件の下に，配列名であってもよい．

(α) 実引数が配列名の場合は，仮引数の配列の添字の最大値は，実引数の添字の最大値をこえてはならない．

(β) 実引数が配列要素名の場合は，仮引数の配列の添字の最大値は，実引数の配列の最大値に1を加え，その実引数の添字の値を減じて得られた値をこえてはならない．

備考 (vii) の場合，実引数が手続きの実行開始時に確定で，かつ引数の型が同じであると，[(II) の (A)] と [2・3・2] によって仮引数である配列も確定となる．

(viii)° 仮引数が外部手続き中で，定義されたり再定義されたりする場合は，対応する実引数は，変数名，配列要素名，あるいは配列名でなければならない。

2・5† 定義についての条件

手続きおよびデータや引数の定義は，実行可能プログラムが与えられ，それが処理系において，実行中である場合に適用されるものとする．これらの定義は，以下の条件に従うものとする．

2・5・1 手続きおよび引数の定義についての条件

(i) 1次子として引用された関数，および CALL 文で引用されたサブルーチンは，その引用を含む実行可能プログラム内で定義されていなければならない．

(ii)† 外部手続きの引用の中の実引数が定数，変数名，配列要素名以外の式である場合，その引用を含む文が実行されるとき，それは確定となっていなければならない．

(iii) 外部手続きの引用の中の実引数が，変数名，配列要素名，配列名である場合は，その手続きが引用されるとき，確定となっていなくてもよい．

(iv) 外部手続きの引用の中の実引数が外部手続き名である場合には，それはその実行可能プログラム内で定義されていなければならない．

(v)† FORTRAN 語以外で定義されている手続きにおいて，その引数となっている変数名や配列名を，この手続きの引用を含む実行可能プログラムの実行に際して，第1種または第3種で確定とすることができるものとする．ただし第3種で確定となるためには，その引数の型が第3種の定義に対応するものでなければならない．

2・5・2† データの定義についての条件

(1)† 第1種の定義についての条件

(i) 再定義の条件：特定の整数型の変数 (〈7・1・2 の (8)，7・1・3，7・2・3〉) や副プログラムの特定のデータ(〈6・4，8・3・2，8・4・2〉) を除いて，1度確定となったデータは，その値が変えられて再び確定となってよい．

(ii) 無名共通ブロック内のデータ：無名共通ブロック内のデータおよびそれと結合しているデータは，最初から確定となることはできないものとする．

(iii) 名前付共通ブロック内のデータ：名前付共通ブロック内のデータまたはそれと結合しているデータは，最初から確定となってもよい．

(iv) 共通ブロックに入らないデータ：共通ブロックに入らないデータで関数の値と仮引数以外のものは，最初から確定となってもよい．

(v) 1次子として引用された変数や配列要素は，その引用を含む文が実行されるとき，確定となっていなければならない．

(vi) DATA 文による初期値定義は，実行可能プログラムにおいて，一つの記憶単位に対しては，ただ1度だけ行なうことができるものとする．

(vii) 関数副プログラムの RETURN 文が実行されるときには，その関数の英字名は確定となっていなければならない．

(viii) DO 文または DO 形並びの初期値パラメタ，終値パラメタ，または増分パラメタとして引用されている変数は，その引用を含む文が実行されるとき，確定となっていなければならない．

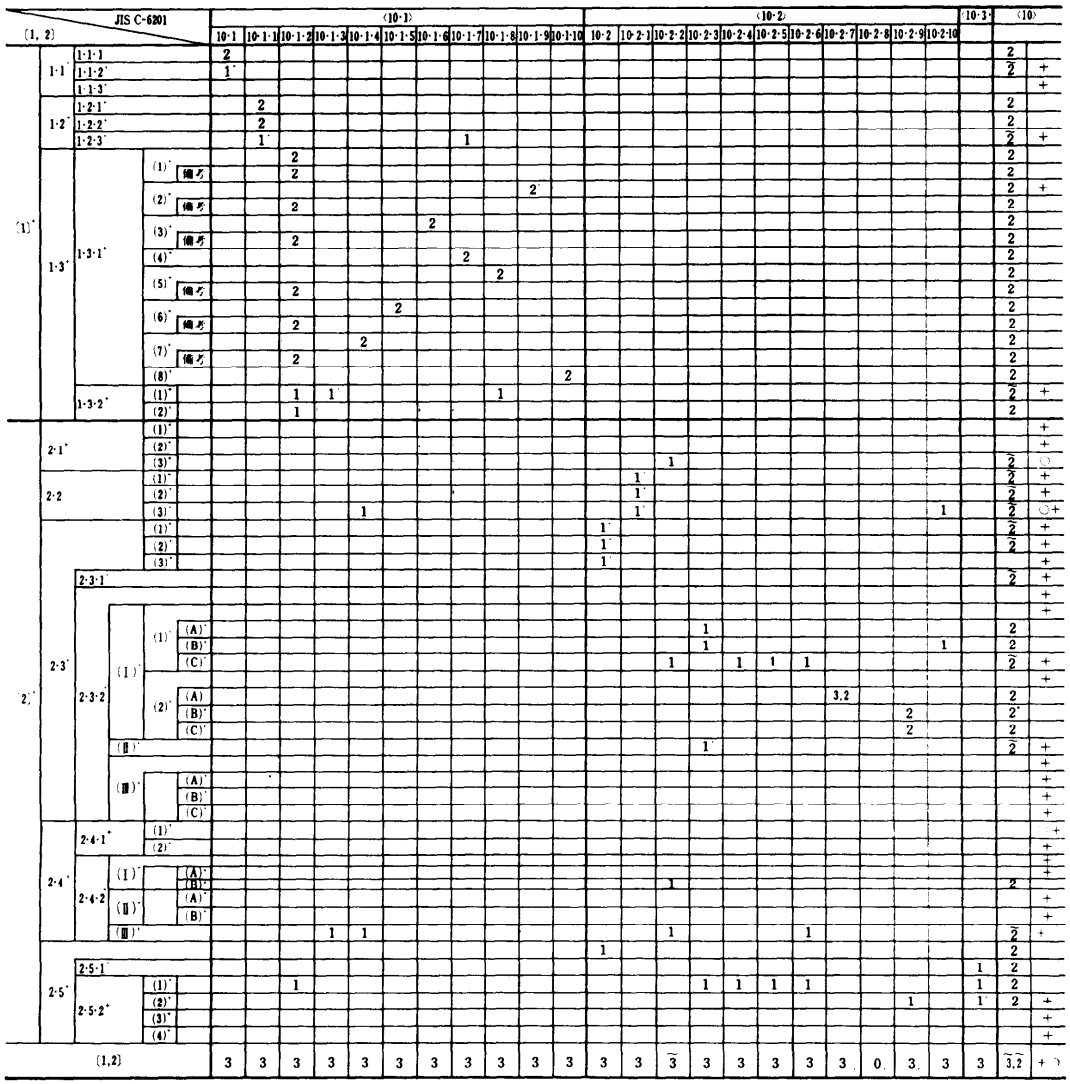
(ix) 入出力装置を識別するために引用されている変数は，その引用を含む文が実行されるとき，確定となっていなければならない．

(x) 出力文が実行されるとき，出力装置に転送される値をもっているデータは，すべて確定となっていなければならない．ただし出力が書式仕様で制御され，かつ対応する変換記号がAのものは除く．出力が変換記号A以外で，書式仕様で制御されている場合には，変換記号とデータの型はつぎのように関連づけられていなければならない．

(α) Iと整数型
(β) Dと倍精度実数型
(γ) E，F，G と実数型および複素数型
(δ) Lと論理型

(2)† 第2階の定義についての条件

(i) 入力並び中以外に現われる添字式，または計算形 GOTO 文の中で，引用された整数型の変数は，その引用を含む文が実行されるとき，第2階で確定となっていなければならない．



〔表の記号の意味〕

(i) a行，b列の要素を（a，b）と書くとき，（a，b）＝0は〔a〕と〈b〉の箇条は素であること，（a，b）＝1は〔a〕と〈b〉の箇条は共通部分があるが，それぞれ共通でない部分もあること，（a，b）＝2は〔a〕の箇条は〈b〉の箇条に含まれること，（a，b）＝3は〈b〉の箇条は〔a〕に含まれること，を表わす。ただし0は記入してない。

(ii) ＋は JIS C 6201 にない項目であること，（a，b）に記されているときは，JIS C 6201 になり，あるいは修正された箇条が付加されていること，○は〈10〉にはないが JIS C 6201 の他の節番号についているときは，箇条が付加されていることを表わす。

(iii) $\tilde{3}$ は，〈10·2·2〉のある部分，〈10·2·8〉を除いて，3であることを表わす。

(iv) $\tilde{2}$ は，＋や○の部分を除いて2であることを表わす。

(v) 右下の角が $\tilde{3}$，$\tilde{2}$ となっていることは，〔1，2〕と〈10〉とが，上記除外部分を除いて，同等であることを表わす。

（ii） 入力並び中に現われる添字式の中で引用される整数型の変数は，その入力文が実行されるときに第2階で確定とならなければならない．

（3）† 第2種の定義についての条件

割当て形 GOTO 文の中で引用された整数型の変数は，この GOTO 文が実行されるとき，第2種で確定となっていなければならない．

（4）† 第3種の定義についての条件

（i） 再定義の条件：第3種の定義に対応する型のデータは，1度第3種で確定となったのちに，サブルーチン手続きの引用によって，その値が変えられて再び第3種で確定となってもよい．

（ii） 無名共通ブロック内のデータ：無名共通ブロック内のデータおよびそれと結合しているデータは，最初から第3種で確定となることはできないものとする．

（iii） 名前付共通ブロック内のデータ：名前付共通ブロック内のデータまたはそれと結合しているデータは，最初から第3種で確定となってもよい，

（iv） 共通ブロックに入らないデータ：共通ブロックに入らないデータは，最初から第3種で確定となってもよい．

（v） DATA 文による初期値定義は，実行可能プログラムにおいて，一つの記憶単位に対しては，ただ1度だけ行なうことができるものとする．

（vi） 出力文が実行されるとき，書式仕様で制御され，かつ変換記号がAであるものに対応する出力装置に転送される値をもっているデータは，第3種で確定となっていなければならない．

（vii）† 書式つき入出力文の（u, f）において，f が配列名である場合，この配列は，この文が実行されるとき第3種で確定となっていなければならない．

備考 配列の値として与えられる書式仕様は，文字定数として扱われることになる．

むすび

以上 JIS FORTRAN の〈**第10節**〉の特性づけと，問題点の指摘，および **I.** の立場からのこの節の体系化を示した．それによって FORTRAN の文に関して非局所的文法についてつぎのような成果が得られたと思われる．

（1） この部分の体系が論理的に明快になった．

（2） 〈**第10節**〉中にある **II・2** に指摘した問題点が解決された．

（3） FORTRAN practice の一層広い文法化が行なわれた．しかし ISO FORTRAN が規定している，あるいは規定しようと意図した範囲の外には出ていない．

ここでは，まえがきに述べた理由から〈**第10節**〉に限定して問題点を論じたが，この節を含む JIS FORTRAN 全体にわたって解説したり，問題点を指摘しているものとしては文献 5），6）がある．またわが国のメーカが問題点をまとめたものとして文献 3）が，USASI が問題点と解釈を与えたものとして文献 7）がある．

なお**第2表**は，**III** の内容が，〈**第10節**〉の内容をなす箇条と集合的にほとんど同じであり，またどういう事項が新たに付加されたかを具体的に示すものである．

参考文献

1） 電子計算機プログラム用言語FORTRAN（水準 7000） JIS C 6201（1967-5） 日本規格協会．
電子計算機プログラム用言語FORTRAN（水準 5000） JIS C 6202.
電子計算機プログラム用言語FORTRAN（水準 3000） JIS C 6203.

2） DRAFT ISO RECOMENDATION, (1965) ISO/TC 97/SC 5

3） 日本工業規格「電子計算機プログラム用言語FORTRAN」に関する検討，（1969-3） 電子工業振興協会．

4） T. B. Steel 編：Formal Language Description Languages for Computer Programming, (1966) North Holland.

5） 菅忠義：JIS FORTRAN の特徴，数理科学，**8**, 2 (1968).

6） 森口繁一：JIS FORTRAN 入門，上巻（1968），下巻（1969） 東大出版会．

7） Clarification of Fortran Standards-Initial Progress, CACM, **12**, **5** (1969).

（昭和44年9月1日受付）