

国际标准化组织推荐文本

# 程序设计语言 ALGOL

黄硕风 译

段 祥 校

科 学 出 版 社

## 内 容 简 介

本书译自国际标准化组织(ISO)推荐的《程序设计语言 ALGOL》标准文本一九七二年第一版。

绪论对 ALGOL 的来源、分级作了简要说明,还解释了基准语言、出版语言和硬件表示的概念以及它的目的范围。第 I 部分为语言说明书,主要介绍了关于算法语言 ALGOL 60 的修正报告,关于 1 级和 2 级子集(ECMA 子集)的报告,以及关于 3 级子集(IFIP 子集)的报告。第 II 部分为输入-输出过程说明书,主要介绍关于 ALGOL 60 初等输入-输出过程和关于 ALGOL 60 的一般输入-输出过程。

本书可供从事计算机程序设计人员、计算数学工作者、工程技术人员、高等学校计算机软件专业的师生参考。

### ISO Recommendation R 1538 PROGRAMMING LANGUAGE ALGOL

Printed in Switzerland, 1972

### 国际标准化组织推荐文本 程序设计语言 ALGOL

黄硕风 译

段 祥 校

\*

科 学 出 版 社 出 版

北京朝阳门内大街 137 号

中国科学院印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

\*

1979 年 10 月第 一 版 开本: 787×1092 1/32

1979 年 10 月第一次印刷 印张: 4 1/2

印数: 0001—45,500 字数: 95,000

统一书号: 13031·1067

本社书号: 1498·13-1

定 价: 0.48 元

# 目 录

|         |   |
|---------|---|
| 绪论..... | 1 |
| 范围..... | 5 |

## 第 I 部分 语言说明书

### 第 I 部分 A: 关于算法语言 ALGOL 60 的修正报告

|                            |    |
|----------------------------|----|
| 1. 语言的结构.....              | 6  |
| 1.1. 语法描述的形式体系 .....       | 7  |
| 2. 基本符号、标识符、数和行——基本概念..... | 8  |
| 2.1. 字母 .....              | 8  |
| 2.2. 数字和逻辑值 .....          | 9  |
| 2.3. 界限符 .....             | 10 |
| 2.4. 标识符 .....             | 11 |
| 2.5. 数 .....               | 12 |
| 2.6. 行 .....               | 13 |
| 2.7. 量、种类和作用域 .....        | 14 |
| 2.8. 值和类型 .....            | 14 |
| 3. 表达式.....                | 15 |
| 3.1. 变量 .....              | 15 |
| 3.2. 函数命名符 .....           | 16 |
| 3.3. 算术表达式 .....           | 18 |
| 3.4. 布尔表达式 .....           | 24 |
| 3.5. 命名表达式 .....           | 26 |
| 4. 语句.....                 | 28 |
| 4.1. 复合语句和分程序 .....        | 28 |

|                 |    |
|-----------------|----|
| 4.2. 赋值语句 ..... | 31 |
| 4.3. 转向语句 ..... | 32 |
| 4.4. 空语句 .....  | 33 |
| 4.5. 条件语句 ..... | 34 |
| 4.6. 循环语句 ..... | 36 |
| 4.7. 过程语句 ..... | 38 |
| 5. 说明 .....     | 42 |
| 5.1. 类型说明 ..... | 44 |
| 5.2. 数组说明 ..... | 45 |
| 5.3. 开关说明 ..... | 47 |
| 5.4. 过程说明 ..... | 49 |
| 过程说明的例子 .....   | 53 |

|                    |    |
|--------------------|----|
| 概念定义和语法单位的索引 ..... | 57 |
|--------------------|----|

## **第 I 部分 B: 关于 1 级和 2 级子集 (ECMA 子集) 的报告**

|             |    |
|-------------|----|
| 引言 .....    | 64 |
| 定义和注意 ..... | 64 |

## **第 I 部分 C: 关于 3 级子集 (IFIP 子集) 的报告**

|             |    |
|-------------|----|
| 引言 .....    | 69 |
| 定义和解释 ..... | 69 |

# **第 II 部分 输入-输出过程说明书**

## **第 II 部分 A: 关于 ALGOL 60 的初等输入-输出过程**

|                 |    |
|-----------------|----|
| 引言 .....        | 77 |
| 1. 定义 .....     | 78 |
| 2. 过程体的作用 ..... | 79 |
| 3. 例子 .....     | 81 |
| 4. 结束语 .....    | 83 |

## **第 II 部分 B: 关于 ALGOL 60 的一般输入-输出过程**

|             |    |
|-------------|----|
| 1. 格式 ..... | 84 |
|-------------|----|

|                       |     |
|-----------------------|-----|
| 1.1. 数格式 .....        | 84  |
| 1.2. 其它格式 .....       | 89  |
| 1.3. 格式行 .....        | 93  |
| 1.4. 格式代码一览表 .....    | 94  |
| 2. 关于输入和输出的标准过程 ..... | 95  |
| 2.1. 一般特征 .....       | 95  |
| 2.2. 水平控制和垂直控制 .....  | 96  |
| 2.3. 布局过程 .....       | 99  |
| 2.4. 表过程 .....        | 103 |
| 2.5. 输入-输出调用 .....    | 104 |
| 3. 例子 .....           | 120 |
| 参考文献 .....            | 122 |
| 英中名词对照表 .....         | 123 |

## 绪 论

ISO 本推荐标准**程序设计语言 ALGOL**，由几种原始文本编纂而成。在这些文本中最基本的是关于算法语言 ALGOL 60 的修正报告。该报告已在许多地方发表，见参考文献[1]。那个报告中定义的语言构成关于 ISO 本推荐标准的其它几部分的基本素材，这几部分是用以叙述语言的各子集以及输入-输出的规定。然而，所有这些附加部分都限于 ALGOL 60 体制之内。

ISO 本推荐标准对原来报告的内容不作任何改变，只是由于编辑上的原因以更合乎逻辑的方式对一些不同部分重新作了安排。

除了基准语言之外，还要建立许多语言子集，建立的理由出于准备原始文本的若干组织的一些实际考虑。旨在正确地使用这一语言的这些考虑是如此重要，所以已经写进了序言之中。

语言说明书各部分的安排是以下面四个不同的实际级别为基础的。

- 0 级。ALGOL 60 全语言，给出完全的定义。
- 1 级。这一实现级是历史上众所周知的 ECMA 子集，但增加了可递归使用过程的能力。
- 2 级。这是低一级的实现级别，即没有任何增加的 ECMA 子集。
- 3 级。这是最低一级可实用的实现级别，即历史上所谓的 IFIP 子集。

一个 ALGOL 60 的子集的意思是指这样的一种语言,使用这个子集语言写成的每个程序自然地也是较高一级子集和全语言写成的程序,并具有相同的意义. 因此,所有的各个子集都是真正嵌套的.

在全语言的描述时还把那些对于各个不同子集的限制,都已放到全语言 ALGOL 60 的报告中. 要把 ALGOL 60 报告转换为 1 级、2 级或 3 级的报告,就必须把方框中的文字去替换那个方框左边的文字. 这些方框都标有它们所属级别的相应数码. 真正的嵌套意味着属于 1 级的方框也可适用于 2 级和 3 级,标有 2 的方框也可适用于 3 级.

除了这种方框形式外,这些子集还用其原始形式给出. 因为这些报告的某些部分不仅包括了定义,而且包括了关于它们的应用的资料.

和 ALGOL 的初步报告一样,这里承认三种不同的语言,即基准语言、出版语言和若干种硬件表示.

## 基 准 语 言

1. 它是委员会的工作语言.
2. 它是定义的语言.
3. 字符取决于容易互相了解,并不取决于计算机的限制、编码者的表示法或纯粹的数学表示法.
4. 它是编译程序设计人员的基本参考材料和指南.
5. 它是一切硬件表示的指南.
6. 它是从出版语言译成任何局部适用的硬件表示的指南.
7. ALGOL 语言本身的主要出版物将使用基准语言.

## 出版语言

1. 出版语言是基准语言的变型，只是考虑了印刷和书写的习惯(例如：下标、空白、指数、希腊字母)。

2. 用它叙述和交流计算过程。

3. 使用的字符在不同的国家中可以是不同的，但应保证和基准语言有唯一的对应。

## 硬件表示

1. 每种硬件表示都是基准语言的一种紧缩，因为在标准输入设备上字符的数目是有限的。

2. 每种硬件表示使用特定的计算机的一组字符，并且是该计算机的翻译程序所接受的语言。

3. 每种硬件表示应附有一组特定的规则，以便把出版语言或基准语言翻译成硬件表示。

为了在基准语言和适合于出版的语言之间进行翻译，建议了下列规则：

### 基准语言：

下标括号 [ ]

乘方  $\uparrow$

圆括号 ( )

底十 <sub>10</sub>

### 出版语言：

把括号中的行降下来并去掉括号。

将指数升上去。

任何形状的圆括号、方括号、花括号。

把十和它后面的指数升起来，并加上所需的乘号。

在 ISO 本推荐标准中，没有把硬件表示提出来作为一个

标准。由于各种不同的输入-输出设备和代码的广泛使用,因此,目前还不可能达到国际上的一致,请参考描述硬件表示的文献中有关部分以及一些国家的标准<sup>[7],[8],[9]</sup>。

ISO 本推荐标准的第 II 部分是介绍关于 ALGOL 60 输入-输出过程的两个报告。这两个报告中的第一个讨论旨在建立较大的输入-输出过程所必需的较少数量的初等过程,这些较大的过程能够很好地用初等过程来描述。虽然用这个方法真正去实现它们可能是不实际的,但是,在特定情形下,即使这些更完全的过程体已用代码表示,通过初等过程的描述保证了它们能够在任何仅实现了初等过程的其它环境中运行。

第二个报告介绍一个完备的输入-输出系统。它是以一组少量的初等过程为基础,通过增加一些对该系统所必需的进一步的初等过程而成。这份报告是由 ACM 程序设计语言委员会的 ALGOL 分会提出的<sup>[7]</sup>。但这个报告并不意味着全输入-输出系统的所有特征都能适用于语言的各级子集。

一般地说,应该承认——尽管 ALGOL 60 在定义的方法上,特别是在语义非形式化方面还存在着某些缺陷——但是,ALGOL 60 已经为语言的目的服务了并且将完全有希望为语言的目的做出非常有益的贡献。任何企图在次要点上修改 ALGOL 语言,只会影响它作为程序设计语言的标准使用。

ISO 本推荐标准的产生途径与普通的 ISO 推荐标准完全不同。普通的推荐标准是由某些国家标准化组织提出的,而这里的基本材料之所以起作用,一方面是由于一些个别的作者公布了这些材料并且散发很广;另一方面也是一些国际性专家组织的努力。这就是说,本语言是不能轻易地改动的,也不能在某一国家标准化组织内改动,而应该通过国际性专家组织。无论如何,ALGOL 60 语言的使用者决不能把后来改

进了的 ALGOL 60 与原来的 ALGOL 60 的名称混同起来。

## 范 围

ISO 本推荐标准定义一种程序设计的算法语言及其子集,以便用来描述这样一大类的数值处理问题,这些问题将以足够简明的形式直接地自动翻译为自动计算机的程序设计语言。

第 I 部分是语言说明书。第 II 部分是输入-输出过程说明书。

# 第 I 部分 语言说明书

---

## 第 I 部分 A:

### 关于算法语言 ALGOL60 的修正报告

#### 1. 语言的结构

如绪论中所述, 算法语言有三种不同的表示——基准语言、硬件表示和出版语言——并且以下的描述用基准语言。这就是说, 在本语言内定义的所有对象都用给定的一组符号来表示, 只是在符号的选择上和其它两种表示可以不同。对所有三种表示而言, 结构和内容应当是一样的。

算法语言的目的是描述计算过程。描述计算规则时用到的基本概念是众所周知的算术表达式(它包含数、变量和函数为其组成部分), 从这些表达式出发, 运用算术运算的规则, 组成语言中自封闭的单位, 叫做赋值语句。

为了表明计算过程的流向, 要加上某些非算术语句和语句句, 它们可以描述比如计算语句的选择或迭代重复等。因为就这些语句的作用而言, 一个语句涉及其它语句是必要的, 所以, 语句可带有标号。一系列语句可被括在语句括号 **begin** 和 **end** 之间从而构成一个复合语句。

语句伴有说明, 说明本身不是计算指令, 但它告诉翻译程序, 语句中出现哪些对象以及对象的某些性质, 例如, 作为变量值的数属于哪一类, 数组的维数, 或者甚至是定义一个函数

的一组规则。把一系列说明及其后面的一系列语句都括在 **begin** 和 **end** 之间就构成一个分程序。每个说明均以此种方式出现于分程序中,并且,只对该分程序有效。

程序是一个分程序或一个复合语句,该分程序或复合语句不包含在其它语句之中,并且不使用它所不包含的其它语句。

下面给出基准语言的语法和语义<sup>1)</sup>。

### 1.1. 语法描述的形式体系

语法将借助于元语言公式<sup>2)</sup>来描述。

对它们的解释最好用一个例子来说明:

$$\langle ab \rangle ::= ( [ | \langle ab \rangle ( | \langle ab \rangle \langle d \rangle$$

括在尖括号  $\langle \rangle$  内的字符序列代表元语言变量,它的值是符号的序列。记号  $::=$  和  $|$  (后者的意义是“或”)是元语言连接词。公式中的任何记号不是变量或连接词时,就表示它自己(或一组与它相类似的记号)。

公式中记号和/或变量的并列,都意味着它们所表示的序列的并列。因此,上述公式给出形成变量  $\langle ab \rangle$  的值的一个递归规则。它指出:  $\langle ab \rangle$  可以有值“(”或 “[”或者给定  $\langle ab \rangle$  的某个合法值后,  $\langle ab \rangle$  另外的值可由在它后面跟以符号“(”或跟以变量  $\langle d \rangle$  的某个值而做出。假如  $\langle d \rangle$  的值是十进制数

- 1) 每当提到计算的精确度,一般说来尚未详细确定或某个过程的结果有待定义或没有定义时,要按下述意义加以解释: 如果附加信息详细说明了所要的精确度和所作的计算种类,并且,对在完成计算时所有可能发生的情况都详细说明了要采取的操作顺序,只在此时才说程序完全确定了一个计算过程。
- 2) 参看 J. W. Backus, The syntax and semantics of the proposed international algebraic Language of the Zürich, ACM-GAMM conference, ICIP, Paris, June, 1959.

1107457

字,则 $\langle ab \rangle$ 的某些值是:

[(((1(37(

(12345(

((

[86

为了便于研究,选定一些近似地描述了相应变量的性质的字作为区别元语言变量(即出现在尖括号 $\langle \rangle$ 中的字符序列,如上例中的 $ab$ )的符号。在文中其它地方用到以这种方式出现的字时都要参照对应的语法定义。此外,有的公式不只在一处给出。

定义:

$\langle \text{空} \rangle ::=$

(即没有符号的空行)。

## 2. 基本符号、标识符、数和行——基本概念

基准语言是用下列基本符号构造起来的:

$\langle \text{基本符号} \rangle ::= \langle \text{字母} \rangle | \langle \text{数字} \rangle | \langle \text{逻辑值} \rangle | \langle \text{界限符} \rangle$

### 2.1. 字母

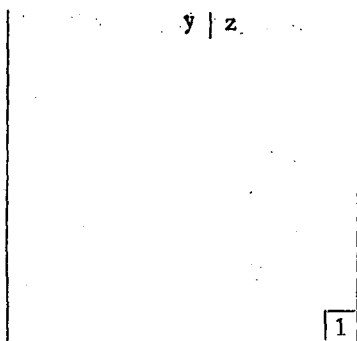
$\langle \text{字母} \rangle ::=$

|   |   |   |   |
|---|---|---|---|
| a | b | c | d |
| e | f | g | h |
| i | j | k | l |
| m | n | o | p |
| q | r | s | t |
| u | v | w | x |

$\langle \text{字母} \rangle ::=$

|   |   |   |   |
|---|---|---|---|
| a | b | c | d |
| e | f | g | h |
| i | j | k | l |
| m | n | o | p |
| q | r | s | t |
| u | v | w | x |

y | z |  
 A | B | C | D |  
 E | F | G | H |  
 I | J | K | L |  
 M | N | O | P |  
 Q | R | S | T |  
 U | V | W | X |  
 Y | Z



这个字母表，可以任意地缩减，或者随意添加任何其它不同的字符（即不与任何数字、逻辑值或界限符相同的字符）而加以扩充。

这个字母表，可以任意地缩减。

注：如果具体实现要求大写字母，我们应当认为它们是小写字母的硬件表示。

3

字母没有单独的意义。它们是用来组成标识符和行的（参看第 2.4 节的第 2.6 节）<sup>1)</sup>。

## 2.2. 数字和逻辑值

### 2.2.1. 数字

$\langle \text{数字} \rangle :: = 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9$

数字是用来组成数、标识符和行的。

### 2.2.2. 逻辑值

$\langle \text{逻辑值} \rangle :: = \text{true} | \text{false}$

1) 应当特别注意，在整个基准语言中粗黑体字用于定义独立的基本符号（参看第 2.2.2 节和第 2.3 节）。这些粗黑体字被认为与组成它们的单独字母没有关系。在本报告中，粗黑体字将不另作它用。

逻辑值有确定而明显的意义。

## 2.3. 界 限 符

〈界限符〉:: = 〈运算符〉|〈分隔符〉|〈括号〉|〈说明符〉|  
〈分类符〉

〈运算符〉:: = 〈算术运算符〉|〈关系运算符〉|〈逻辑运算符〉|〈顺序运算符〉

〈算术运算符〉:: =  
+|-|×|/|÷|↑

〈算术运算符〉:: = +|-|  
×|/|↑

3

〈关系运算符〉:: = <|≤|=|>|≥|≠

〈逻辑运算符〉:: = ≡|⊃|∨|∧|¬

〈顺序运算符〉:: = goto|if|then|else|for|do<sup>1)</sup>

〈分隔符〉:: = ,|.|:|;|:=|□|step|until|while|  
comment

〈括号〉:: = (|)|[|]|'|begin|end

〈说明符〉:: = own|  
Boolean|  
integer|  
real|  
array|  
switch|  
procedure

〈说明符〉:: = Boolean|  
integer|real|  
array|switch|  
procedure

1

〈分类符〉:: = string|label|value

1) do 是用在循环语句中，它与初步报告中的 do 没有任何关系，那个 do 不包含在 ALGOL 60 中。

界限符有确定的意义,大部分界限符的意义是明显的;否则,其意义将在后面适当的地方给出。

在基准语言中,象空白或另起一行等印刷上的特点是什么含义的。但是,为了容易阅读起见,可以随意地使用它们。

为了在程序的符号间加进一段话,下面关于“注解”的约定成立:

| 基本符号的序列:                                                | 是等价于:        |
|---------------------------------------------------------|--------------|
| <b>;</b> <b>comment</b> <任何序列,<br>但不包括“;”>;             | ;            |
| <b>begin comment</b> <任何序列,<br>但不包括“;”>;                | <b>begin</b> |
| <b>end</b> <任何序列,但不包括<br><b>end</b> 或“;”或 <b>else</b> > | <b>end</b>   |

所谓等价这里是指:左列所示三种结构中的任何一种在行外出现时,换成右列同一排所示的符号后,对程序的执行没有任何影响。更可进一步理解为:从左向右读在文中首先遇到的注解结构比该序列中包含的以后的注解结构要优先替换。

## 2.4. 标识符

### 2.4.1. 语法

<标识符> :: = <字母> | <标识符> <字母> | <标识符> <数字>

### 2.4.2. 例

*q*

*soup*

V-17 a

a 34 KTMN s

MARILYN

### 2.4.3. 语义

标识符没有固定的意义,只用于标识简单变量、数组、标号、开关和过程。

它们可以随意地选用  
(参看第 3.2.4 节)。

标识符可以随意地选用;但是由前六个基本符号相同的两个不同标识符引起的效果没有意义的(参看第 3.2.4 节)<sup>1)</sup>。

1

不能用同一标识符表示两个不同的量,除非程序的说明确定了这两个量有不相交的作用域(参看第 2.7 节和第 5 节)。

## 2.5. 数

### 2.5.1. 语法

$\langle \text{无正负号整数} \rangle ::= \langle \text{数字} \rangle | \langle \text{无正负号整数} \rangle \langle \text{数字} \rangle$

$\langle \text{整数} \rangle ::= \langle \text{无正负号整数} \rangle | + \langle \text{无正负号整数} \rangle | - \langle \text{无正负号整数} \rangle$

$\langle \text{十进制小数} \rangle ::= . \langle \text{无正负号整数} \rangle$

$\langle \text{指数部分} \rangle ::= {}_{10} \langle \text{整数} \rangle$

$\langle \text{十进制数} \rangle ::= \langle \text{无正负号整数} \rangle | \langle \text{十进制小数} \rangle | \langle \text{无正负号整数} \rangle \langle \text{十进制小数} \rangle$

1) “(参看第 3.2.4 节)” 原文遗漏。——校者注