

算法语言 ALGOL 60 入门

吉林大学数学系 编
计算数学教研室

内 容 简 介

计算机的结构复杂，使用时要编程序，算法语言就是编程序必须依据的规则。

算法语言的种类很多，目前我国使用的以 ALGOL 60 居多。由于 ALGOL 60 精致、合理，已为国际通用。

本书是 ALGOL 60 的入门书，内容浅显，解释清楚。全书共三章，分别是：ALGOL 60 的简单描述、过程说明和过程语句及 ALGOL 60 的进一步描述。

算法语言标准化很重要，本书介绍的是标准 ALGOL 60 语言，具有普遍的参考价值。

算法语言 ALGOL 60 入门

吉林大学数学系 编
计算数学教研室

*

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

*

1976 年 7 月第 一 版 开本：787×1092 1/32

1976 年 7 月第一次印刷 印张：4 1/2

印数：0001—35,450 字数：99,000

统一书号：13031·409

本社书号：618·13-1

定 价：0.38 元

序

目前,在科学和工程计算中,最常用的程序语言有三种,即 ALGOL 60, FORTRAN 和 BASIC. 本书是算法语言 ALGOL 60 的入门书.

本书详细介绍了算法语言 ALGOL 60 的基本内容. 对 ALGOL 60 中比较难懂而又比较重要的概念,如分程序、局部量、过程等,也做了详尽的解释.

本书可供使用电子计算机算题的人员、研制软件的工作人员和高等院校、中等专业学校有关专业师生参考.

参加本书编写工作的有: 金成植、徐立本和金淳兆同志. 书中错误不当之处,请批评指正.

吉林大学数学系

计算数学教研室

1973年8月16日

目 录

第一章 ALGOL 60 的简单描述

§ 1 基本符号、标识符、数	1
§ 2 赋值语句、简单算术表达式	4
§ 3 条件语句、转向语句、标号	7
§ 4 复合语句	13
§ 5 类型说明	15
§ 6 分程序	19
§ 7 循环语句	27
§ 8 数组说明、下标变量	35
§ 9 标号的作用域	40
§ 10 开关说明、开关命名符	41
§ 11 分程序的例子	47

第二章 过程说明和过程语句

§ 1 基本概念	57
§ 2 过程说明和过程语句的语法结构	61
§ 3 过程的调用	68
§ 4 换名的例	70
§ 5 值部分	79
§ 6 函数过程	83
§ 7 无参数过程	85
§ 8 过程说明的例	87

第三章 ALGOL 60 的进一步描述

§ 1 空语句	103
§ 2 条件算术表达式	104
§ 3 布尔表达式、布尔赋值语句	106
§ 4 多重赋值语句	111
§ 5 命名表达式和开关说明	113
§ 6 数组的动态分配	116
§ 7 行	118
§ 8 标识符的冲突	118
§ 9 代码过程	123
§ 10 递归过程、递归调用	125
§ 11 ALGOL 60 的形式描述	130
参考文献	136

第一章 ALGOL 60 的简单描述

§ 1 基本符号、标识符、数

(一) 基本符号

任何一个 ALGOL 程序, 都是由基本符号严格地按 ALGOL 语言的语法写成的。不允许在 ALGOL 程序中出现非基本符号。

基本符号共 116 个, 分四大类:

- 1) 字母(52 个),
- 2) 数字(10 个),
- 3) 逻辑值(2 个),
- 4) 定义符(52 个)。

具体列表如下:

ALGOL 60 基本符号一览表

字 母			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
数 字			0 1 2 3 4 5 6 7 8 9
逻 辑 值			true false
定 义 符	算 术	运 算 符	+ - × / + ↑
	关 系		< ≤ = ≥ > ≠
	逻 辑		∨ ∧ → ⊃ =
	顺 序		goto if then else for do

续表

定 义 符	分 隔 符	, . 10 ; ; := □ step until while comment
	括 号	() [] ' ' begin end
	说 明 符	own Boolean integer real array switch procedure
	分 类 符	string label value

(二) 标识符和标准函数

以字母开头的字母和数字的序列,称为标识符、标识符是ALGOL中很重要的概念。

标识符例:

a, eps, Y 3, Pressure, SUM, A 176.

下列各符号行不是标识符:

5 A, b 15.5, K-point, α .

标识符的作用类似于数学中的 x , y , f 等,用以代表某种量,即代表一些量的名字。究竟作为哪种量的名字(比如,是简单变量,还是向量或矩阵,还是函数),由编写ALGOL程序的人在ALGOL程序中给出说明。标识符本身没有独立的意义。

如同数学中的 $\sin$, $\cos$, $\ln$, $\arctan$ 等专用函数符号一样,在ALGOL中,下列标识符做为专用的函数符号(称为标准函数)。标准函数符号有

abs, sign, sqrt, sin,

cos, arctan, ln, exp, entier.

其意义如下:

abs(E) 表达式 E 之值的模(绝对值);

sign(E) E 之值的符号(当 $E > 0$ 时是 +1, 当 $E = 0$ 时是

0, 当 $E < 0$ 时是 -1);

$\text{sqrt}(E)$ E 之值的平方根;

$\sin(E)$ E 之值的正弦;

$\cos(E)$ E 之值的余弦;

$\arctan(E)$ E 之值的反正切的主值;

$\ln(E)$ E 之值的自然对数;

$\exp(E)$ E 之值的指数函数(e^E);

$\text{entier}(E)$ 转换函数. 将实数型表达式的值转换为整数, 其值就是不大于 E 之值的最大整数.

例如

$$\text{entier}(1.95)=1, \quad \text{entier}(-1.95)=-2.$$

(三) 数

用 N 表示任意数字串(例如, 539,0890), 则下列几种形式的数均定义为 ALGOL 60 中的一个数:

- 1) $\pm N$,
- 2) $\pm .N$,
- 3) $\pm N.N$,
- 4) $\pm N_{10} \pm N$,
- 5) $\pm .N_{10} \pm N$,
- 6) $\pm N.N_{10} \pm N$,
- 7) $\pm_{10} \pm N$,

其中 $\pm N$ 表示 N 或 $+N$ 或 $-N$, 其余情形类似.

数的最一般形式为

$$\underbrace{\pm}_{\text{符号}} \underbrace{N}_{\text{整数}} \underbrace{.N}_{\text{小数}} \underbrace{{}_{10} \pm N}_{\text{指数}}$$

简单情形是取其中的一部分.

例,

0	-200.084	-.083 ₁₀ -02
177	+07.43 ₁₀ 8	- ₁₀ +7
-.53	9.34 ₁₀ +10	₁₀ -7
+0.7300	2 ₁₀ -4	+ ₁₀ +50

其中小写₁₀是 ALGOL 60 的一个基本符号, 它只能出现在数中, 但不能单独出现。

§ 2 赋值语句、简单算术表达式

例 1.1. 在输入纸带上已穿三个数, 要求按次序把它们传送给变量 a, b, c , 并计算表达式 $\frac{a \cdot b + c}{a \cdot b - 2c}$ 的值, 最后, 打印出表达式的值。

ALGOL 程序

```
Read r (a, b, c);  
h := a × b;  
g := (h + c) / (h - 2 × c);  
Print r (g);
```

本题的 ALGOL 程序是由 4 个语句组成的。语句是 ALGOL 程序的组成单位。这里的语句严格地按其写出的顺序执行。在语句之间必须用分号“;”把语句一个个分隔开。

上面 ALGOL 程序的意义如下:

第一个语句——将纸带上的三个实数依次传送给变量 a, b, c ;

第二个语句——将表达式中分母、分子共用的值 $a \times b$ 算出,

暂存起来, 赋给变量 h ;

第三个语句——将 $\frac{a \cdot b + c}{a \cdot b - 2c}$ 的值算出, 并赋给变量 g ;

第四个语句——将变量 g 的值打印出来。

语句是按其作用分类的。在本题中,

第一个语句——输入语句;

第二、三个语句——赋值语句;

第四个语句——输出语句。

$:=$ 是一个符号, 而不是两个符号, 称为赋值号。它不是等号, 不能与等号同样使用。

赋值语句的一般形式是

$$V := E$$

其中 V 表示变量, E 是算术表达式。赋值语句的意义是 将表达式 E 的值赋给变量 V 。例如

$$x := x + 5$$

是一个赋值语句, 表示 x 的当前值加上 5 再赋给变量 x , 即变量 x 的值增加 5。

简单算术表达式是由运算分量、运算符和圆括号 按 通常的规则形成的。

运算分量包括数, 变量, 函数等。

运算符有 $+$, $-$, $\times$, $/$, $\div$, $\uparrow$,

其中 $+$ 和 $-$ 可以是单值位运算 (即单目运算, 如 $x := -a$); 运算符 $/$ 就是通常意义下的除法运算符; 运算符 $\div$ 是整数型除法运算符 (后面介绍); 运算符 $\uparrow$ 是指数运算符。

在同一括弧里, 运算的优先规则如下:

1) $\uparrow$.

2) $\times$, $/$, $\div$,

3) +, -.

对同一级的运算符,按它们在表达式中的顺序(从左向右)执行。例如

$$\begin{aligned} a+b-c & \quad \text{表示 } (a+b)-c, \\ a/b \times c & \quad \text{表示 } (a/b) \times c, \\ a \times b \uparrow c + d & \quad \text{表示 } (a \times (b \uparrow c)) + d, \\ -a \uparrow b & \quad \text{表示 } -(a \uparrow b). \end{aligned}$$

在运算符 $\uparrow$ 的后边不准许存在其他运算符。例如

$$a \uparrow -b$$

没有意义。

算术表达式举例。

常用写法	ALGOL 写法	错误写法
a^b	$a \uparrow b$	
a^{b+c}	$a \uparrow (b+c)$	$a \uparrow b+c$
a^{-2}	$a \uparrow (-2)$	$a \uparrow -2$
$10^{-4.7}$	$10 \uparrow (-4.7)$	$10 \uparrow -4.7$

对算术表达式,需要指出如下几点:

1) 对 ALGOL 来说,表达式中的乘号不能省略。例如,不能把 $a \times b$ 写成 ab 。前者表示两个变量的乘积,后者是一个标识符;

2) 在 ALGOL 中,表达式的括弧一律用圆括弧,不准用方括弧或花括弧。ALGOL 中的方括弧有另外的用法;

3) 除法用符号“/”和“÷”,不得写成 $\frac{a}{b}$ 的形式;

4) 标准函数自变量的圆括弧不能省略。比如, $\sin(x)$ 不能写成 $\sin x$ 。后者是一个标识符。

赋值语句的最一般形式是连续赋值。例如

$$x:=y:=z:=0.5$$

表示对变量 x , y , z 均赋给算术表达式 0.5 的值。这种赋值

语句叫多重赋值语句。它的一般形式为

$$V_1 := V_1 := \dots V_n := E$$

注。在 ALGOL 60 报告中，没有谈输入输出问题。但任何解题过程都要有输入和输出，因此，每个具体的编译程序都规定一套输入和输出语句，并把它做为标准过程（标准子程序）使用。各具体编译程序所规定的标准输入输出过程的名字和用法并不一样，有专门谈输入输出问题的文章，如[12]和[11]。本书将陆续引进我们自己规定的一些输入输出语句（标准过程）。本节引进了两个输入输出标准过程：Read r 和 Print r 。Read $r(a)$ 是把当前纸带上的一个实数输入（赋）给变量 a 。在（ ）中可以写多个变量的名，但必须用逗号把它们隔开，如 Read $r(a, b)$ 。它表示把当前纸带上的两个实数依次赋给变量 a 和 b 。Print $r(b+c)$ 是把（ ）里的表达式 $a+b$ 的值打印出来。在（ ）中表达式可以写几个，但必须用逗号把它们分隔开。为简单计，本书假定计算机只有一个输入设备和一个输出设备。实际情况要复杂一些。

§ 3 条件语句、转向语句、标号

例 1.2. 纸带上穿有一串数，最后一个数的标志是其值为 0。试求正数的个数和所有负数的平方和，并打印出来。

算法框图：见下页图。

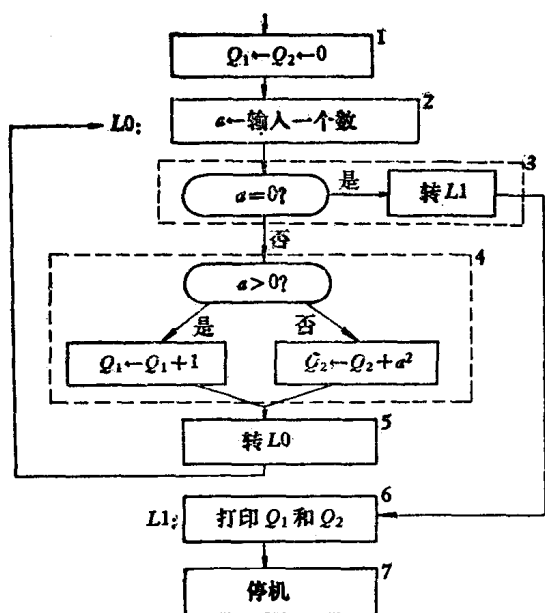
ALGOL 程序

```

      Q1:=Q2:=0;                                (1)
L0:  Read  $r(a)$ ;                                (2)
      if  $a=0$  then goto L1;                      (3)
      if  $a>0$  then Q1:=Q1+1 else Q2:=Q2+a↑2;    (4)
      goto L0;                                  (5)
L1:  Print  $r(Q1, Q2)$ ;                           (6)
      STOP                                     (7)

```

本程序由七个语句组成。每个语句对应于相应编号的框图部分。



说明:

用 $S_1, S_2, S_3, \dots$ 分别表示第一个语句, 第二个语句, 第三个语句, $\dots$, 上面 ALGOL 程序中各个语句的意义如下:

S_1 : 将变量 Q_1 和 Q_2 置 0;

S_2 : 是带标号 ($L 0$) 的语句, 它将纸带上的当前数传送给 a ;

S_3 : 如果 $a=0$ 则转向以 $L 1$ 为标号的语句 (即 S_6) 否则执行下一个语句;

S_4 : 如果 $a > 0$, 则 $Q_1 \leftarrow Q_1 + 1$, 否则 $Q_2 \leftarrow Q_2 + a^2$;

S_5 : 转向以 $L 0$ 为标号的语句 (即 S_2);

S_6 : 打印正数的个数 Q_1 和负数的平方之和 Q_2 ;

S_7 : 停止。

本程序中的 **goto** $L 0$, **goto** $L 1$ 称为转向语句, 分别转至以 $L 0, L 1$ 为标号的语句, 其中的 $L 0, L 1$ 称为标号。标

号标出语句的名字，以便从别处通过转向语句转至该语句。
STOP 是一个语句，即停止语句，它表示停止计算。

程序中的 S_3 和 S_4 ，即

if $a=0$ then goto L1

和

if $a>0$ then $Q1:=Q1+1$ else $Q2:=Q2+a \uparrow 2$ 均称为条件语句，其中的 $a=0$ 和 $a>0$ 称为关系式。关系式的一般形式是

$$E_0 < \text{关系符} > E_0$$

E_0 表示简单算术表达式，关系符有

$<, \leq, >, \geq, =, \neq$.

下面是关系式的例子：

$a \leq 0$,

$b < a + c$,

$f + g + x = y$,

$x \neq x$ (它总是假的),

$3 \neq 4$ (它总是真的)。

ALGOL 60 的语句分为三大类：

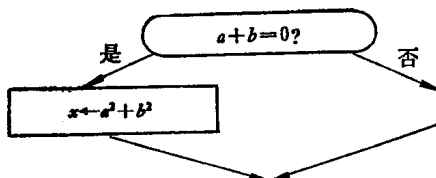
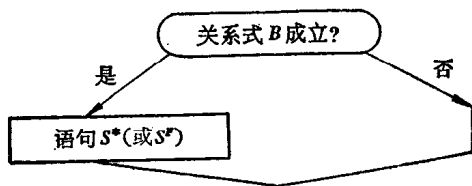
1. 条件语句，
2. 无条件语句，
3. 循环语句。

条件语句分为两类。如果用 S^* 表示无条件语句， S 表示任意语句， S^F 表示循环语句，则条件语句的一般形式为

1. if B then S^* (或 S^F);
2. if B then S^* else S ;

其中 B 表示布尔表达式，而关系式是布尔表达式的一例。我们把 B 暂时理解为关系式。

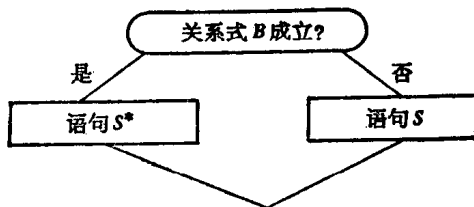
第一类条件语句,即“if—then”型的条件语句的意义是:如果其中的关系式 B 成立,则执行 **then** 后面的语句 S^* (或 S^F);如果关系式 B 不成立,则该条件语句等于空语句,即什么也不做。这种条件语句对应于下面的框图:



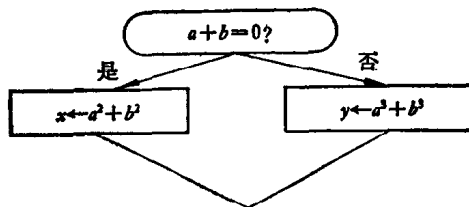
其特点是一个分支为空,即什么也不做。对应第二个框图的条件语句可以写成

if $a + b = 0$ **then** $x := a \uparrow 2 + b \uparrow 2$

第二类条件语句,即“if—then—else”型条件语句的意义是:如果关系式 B 成立,则执行 **then** 后面的语句 S^* ,然后跳过 **else** 后面的整个语句 S 执行该条件语句的后继语句;如果关系式 B 不成立,则执行 **else** 后面的语句 S 。“if—then—else”型条件语句对应于如下的框图:



其特点是两个分支都不是空的。例如，对应框图

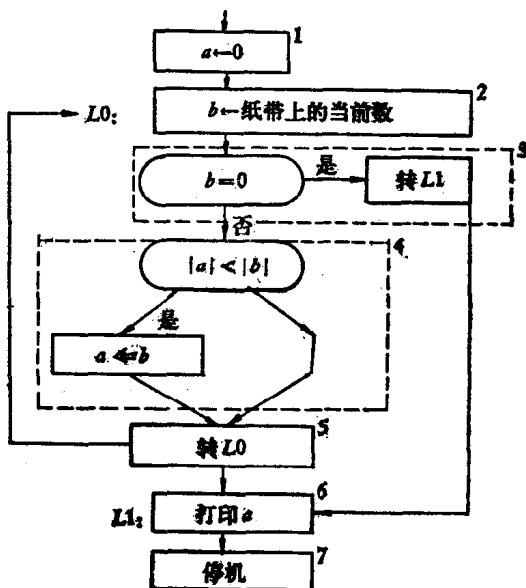


的条件语句可写成

if $a + b = 0$ **then** $x := a \uparrow 2 + b \uparrow 2$
else $y := a \uparrow 3 + b \uparrow 3$

例 1.3. 纸带上穿有一串数，最后一个数的标志是其值为 0。试求其中绝对值最大的数，并将它打印出来。

算法框图：



ALGOL 程序

```
a:=0;  
L0: Read r(b);  
    if b=0 then goto L1;  
    if abs(a) < abs(b) then a:=b;  
    goto L0;  
L1 Print r(a);  
STOP
```

应该注意的是,在条件语句中, **else** 后面的语句 S 又可以是条件语句(但 **then** 后面不允许是条件语句)。

例 1.4. 纸带上穿有一串数,最后一个数的标志是其值为 0。试对纸带上的每个数求下列函数的值。

$$f(x) = \begin{cases} x + \sin x & \text{当 } x > 0.5, \\ x^2 + e^x & \text{当 } x < -0.5, \\ x^3 + \cos x & \text{其它情形.} \end{cases}$$

ALGOL 程序

```
L: Read r(x);  
    if x=0 then STOP;  
    if x>0.5 then Print r(x+sin(x))  
    else if x<-0.5 then Print r(x↑2+exp(x))  
    else Print r(x↑3+cos(x));  
    goto L;
```

注: 1) 在 ALGOL 60 报告中 ([1] §3.5.1), 无符号整数可以做为标号, 但由于语言的二义性, 一般都不准把无符号整数做为标号。

2) 在 ALGOL 60 报告中没有停语句, 但考虑到使用方便, 一般都设置一些停语句, 本书用 STOP 表示停语句。