电子计算机介绍

程序自动化基础

清华大学
电子工程系 计算数学教研组 编

科学出版社

程序自动化基础

清华大学
电子工程系计算数学教研组 编

科学出版社
1975

内 容 简 介

本书主要介绍 DJS-5 机上的一个编译系统，包括算法语言、编译程序的结构等方面的内容。对于掌握其他程序语言和开展编译程序工作，也有一定的参考价值。

程序自动化基础

清华大学
电子工程系计算数学教研组 编

*

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

*

1975年7月第一版　开本：787×1092　1/32
1975年7月第一次印刷　印张：9 11/16
印数：0001—39,450　字数：218,000

统一书号：13031·348
本社书号：532·13—1

定价：0.78 元

毛主席语录

中国人民有志气，有能力，一定要在不远的将来，赶上和超过世界先进水平。

在生产斗争和科学实验范围内，人类总是不断发展的，自然界也总是不断发展的，永远不会停止在一个水平上。因此，人类总得不断地总结经验，有所发现，有所发明，有所创造，有所前进。

前　言

在伟大领袖毛主席关于“**独立自主，自力更生**”的方针和社会主义建设总路线的指引下，在无产阶级文化大革命取得伟大胜利的大好形势下，我国电子计算机技术及其应用日益发展。随着计算机在国防军事、国民经济各部门日益广泛的应用，广大工农兵和工程科技人员迫切需要了解计算机的功能，并学会使用，因此开展程序自动化的研究和推广就显得很重要了。

本书主要介绍DJS-5机上的一个编译系统，包括算法语言、编译程序的结构等方面的内容。对于掌握其它的程序语言和开展编译程序的工作，也有一定的参考价值。书中备有附录供读者参考。

由于我们学习马列主义、毛主席著作不够，实际经验和理论研究都很少，书中必然会有许多错误之处，欢迎读者批评指正。

目　录

第一部分　算法语言介绍

第一章　DJS-5算法语言一般介绍

§1　算法语言、机器语言

用电子计算机解数学问题时，首先要探讨合适的计算方法，然后把计算方法编成机器指令组成的程序。这种机器指令通常又称机器语言。事实上，机器语言和一般的数学语言有很大的不同，初学程序设计的人感到很不习惯。机器语言和数学语言各有其优缺点，列表表示如下：

数学语言		机器语言
对机器不适用。	→	适合于机器计算。
难于对计算过程做细致的描述。	→	可以描述计算过程的细节。
易读，易接受。	←	难读，不易接受。

算法语言

算法语言（亦称程序语言）是吸取前面二种语言各自的优点的程序设计语言，它比较接近于数学语言，但用它又能描述计算过程。它具有以下的特点：直观性（阅读和书写起来比较方便易懂），精确性（含意确定，不会产生混淆，机器能够识别）和通用性（不象机器语言只局限在特定的机器上，它可以适用于一般的机器，因此，可以用它来交流科学计算的知识）。

用算法语言写成的程序通称源程序。源程序可以输入到机器中去，但机器不能直接按照这种语言执行操作。解决的办法不外乎两种：

（一）由一个编译程序把源程序翻译成意思一样的机器语

言的程序，称目标程序．这犹如把用英文写成的文章翻译成中文一样．

(二)通过解释程序对源程序进行解释执行．

以上两者要由专门的人员为算法语言配备相应的编译程序或解释程序．这是电子计算机上应该配备的主要程序系统(或叫软设备)．

在六十年代初期，设计了一种算法语言称ALGOL60，它在国际上得到广泛的应用．DJS-5算法语言是在ALGOL 60语言的基础上，进行缩减和改进设计而成的．

本章先对DJS-5算法语言作一般的介绍．

§ 2 基本符号、名字、变量

在科学计算中，往往会遇到各种形式的计算公式．例如要计算$e^{\frac{1}{4}x}$，可以把它展开为台劳级数，取前五项得到下面的近似公式：

$$e^{\frac{1}{4}x} = 1 + \frac{1}{4}x + \frac{1}{32}x^2 + \frac{1}{384}x^3 + \frac{1}{6144}x^4 .$$

为了便于程序设计，可变换为

$$e^{\frac{1}{4}x} = 1 + x\left(\frac{1}{4} + x\left(\frac{1}{32} + x\left(\frac{1}{384} + \frac{1}{6144}x\right)\right)\right).$$

如果我们希望计算的结果放在Y这个单元中，用DJS-5算法语言写出的语句为

y:=1+X*(1/4+X*(1/32+X*(1/384+1/6144*X))).

上面“:=”右端部分和数学公式非常相似．这里用到X，Y二个汉语拼音字母，用到阿拉伯数字，用到算术运算符“+”，“*”，“/”，用到“(”和“)”．其中“*”代表乘号，“/”代表除号，“:=”表示把右边的计算结果放到左边的单元Y中去．

在DJS-5算法语言中，有七十七个基本符号，包括

A，B，…，Z等二十六个汉语拼音字母，“0”，“1”，…，“9”十个阿拉伯数字，“+”，“-”，“×”，“/”…等十五个符号，另外还有二十六个拼写起来的符号，它们由两个拼音字母或两个符号拼写起来，当作一个基本符号。其中大部分是用来为了使写出的程序比较直观易懂，一小部分是用

DJS-5算法语言拼写定义符表

拼写定义符	意　　义
KS	“开始”的汉语拼音缩写
JS	“结束”的汉语拼音缩写
RU	“如”的汉语拼音缩写
ZE	“则”的汉语拼音缩写
FO	“否”的汉语拼音缩写
DY	“对于”的汉语拼音缩写
ZO	“做”的汉语拼音缩写
ZX	“转向”的汉语拼音缩写
GR	“光（电输）入”的汉语拼音缩写
DR	“电（传输）入”的汉语拼音缩写
DC	“电（传输）出”的汉语拼音缩写
HC	“火（花输）出”的汉语拼音缩写
DD	“读带”的汉语拼音缩写
JD	“记带”的汉语拼音缩写
TJ	“停机”的汉语拼音缩写
FJ	“符（合）停(机)”的汉语拼音缩写
JB	“简（单）变(量)”的汉语拼音缩写
XL	“向量”的汉语拼音缩写
JZ	“矩阵”的汉语拼音缩写
BL	“变量”的汉语拼音缩写
GC	“过程”的汉语拼音缩写
FZ	“赋值”的汉语拼音缩写
SA	“删”　的汉语拼音缩写
:=	赋值号
*/	乘幂运算符，其右运算项的值为实数
**	乘方运算符，其右运算项的值为整数

来弥补输入设备的符号不足．这些拼写起来的符号我们称为拼写定义符．

用DJS-5算法语言写出的程序就是由基本符号表中七十七个符号组成的．

名字 名字用以表示某一个对象，例如在上面的例子中，我们用X代表某一变量，X就是代表这一变量的名字．DJS-5算法语言规定名字都是以汉语拼音字母开头后跟字母或数字组成的，构成一个名字的符号的总数不能超过三个．例如X、L1、ETA、FY1、A1E等都是名字．

变量 前面例中的变量X称为简单变量，在数学公式中还常常碰到矩阵和向量的运算，它们的元素通常可表示为：A_i，$B_{m,n}$等，在DJS-5算法语言中把它们称做下标变量，写成：A〔I〕，B〔m，n〕，其中A和B后面的一对方括号把下标括了起来．由于机器无法用位置的高低来辨认下标，因此方括号的作用就是用来表示括起来的符号是下标．我们称A为一向量（它具有若干个元素），而A〔I〕为向量A的一个元素．同样称B为一矩阵（设它具有L列，K行，故有K×L个元素），而B〔m，n〕为其中的一个元素（是第m行第n列位置上的一个元素）．向量和矩阵合称数组．

§3 算术表达式、赋值语句

前面所举的例中，“:=”的右边是一个算术表达式，算术表达式是由一些变量和无符号数通过“+”、“-”、“*”、“/”等运算符连接起来并用括弧适当地括起来而组成．在算术表达式中还可以出现一些标准函数，如SIN（X），LN（Y）（自然对数），SQR（W）（开平方根），ABS（Z）（求绝对值）等．表达式的计算次序与一般的代数式的计算次序一样．在表达式中“*”是不能省掉的，例如：

A＊3—B＊4不能写成A3—B4，因为这样写就表示两个变量A3和B4做减法。下面所写的都是算术表达式。

1 +0.01＊P/Q + COS(ETA)

R＊＊ 2 ＊3.14159

A + B＊(C + D＊(E + F))

由于机器无法辨认符号位置的高低，故在表达式中若分子或分母多于一项，应加括号以示区别。如$A-\frac{B+C}{E\times D}\times F$应写成A -(B +C)/(E＊D)＊ F其中B + C 和 E＊D，分别用括号括起来表示分子和分母。显然，$A\times\frac{B}{C}+D$可写成A＊B/C + D，无须再加括号。

符号“:=”称赋值号，上面§ 2 所写将$e^{\frac{1}{4}x}$的值送到Y中去的语句称作赋值语句。“:=”的左边是变量，“:=”的意思是将其右端的算术表达式的值赋给左边的变量。例如A:= 3 意即将 3 赋给变量A。“:=”与一般数学公式中的“=”意义不一样。它具有动态的意义，例如A:=A + 3；即表示将变量A原来的值与 3 相加后赋给变量A，左边的变量在赋值后就具有新的值即右边表达式的值，而不是象“=”那样只是建立“=”两边的相等的关系。下面的写法：

3.14＊Y:=COS(X)＊ 6 + X

是错误的，因为左边是一个表达式，它计算出一个值来，而不是一个变量。

§ 4　转向语句、条件语句

在手编程序中常常用到无条件转移指令，在DJS-5算法语言中也具有这样功能的语句，称做转向语句。手编程序以机器指令为单位，写出的是一条条指令，转移指令应指明所

要转向的那条指令在内存的地址．算法语言中规定以语句为单位，写出的程序是一系列的语句，因此转向语句应指明所要转向的语句的名字．语句的名字叫做标号，例如L：Y:=A+B；其中L即称标号，它用“：”与后面的语句分隔开来．如果在程序的某处需要转向这句语句，就可以写：ZX L；并非每句语句都要加标号，只有在需要时才加标号．

注意，每句独立的语句都要用“；”分开，否则就容易产生错误．语句按写出的次序顺次执行，当执行转向语句后就中断了这个次序，程序从所转向的语句再开始顺次执行．

与手编程序中的条件转移指令一样，算法语言中有条件语句以实现条件转向．下面举一个例子说明．

例 设有复数x+yi，其中$x=\frac{5}{13}$，$y=\frac{12}{13}$，用复数0.6+0.8i与它相乘，所得结果再用0.6+0.8i相乘，如此共乘100次，试求最后结果的实部和虚部．

```
（1）     X:=5/13;   Y:=12/13;
（2）     K:=0;
（3）  L1: U:=0.6*X-0.8*Y;
（4）      Y:=0.8*X+0.6*Y;
（5）      X:=U;   K:=K+1;
（6）      RU   K<100  ZE  ZX  L1;
（7）      DC(X, Y);
```

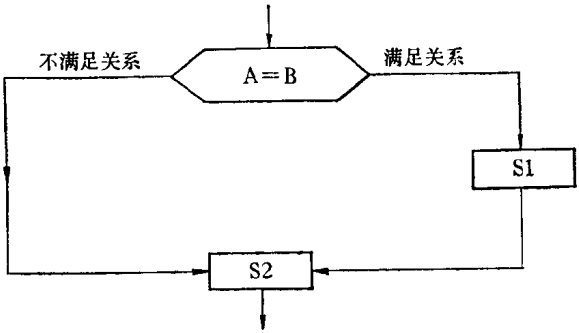
其中（6）为条件语句，当K＜100时即转向语句L1，重复执行，否则就执行下一语句（7）（即整个条件语句后面的语句）．（7）为电传输出打印X和$\overline{Y}$．

这种条件语句的一般形式为

RU A＜B ZE S1；或

RU A=B ZE S1；

其中A及B代表算术表达式，A = B，（A＜B）构成一个关系(即条件)，如满足条件则执行语句S1，不然就执行条件语句后面的语句，例如：**RU** A = B **ZE** S1；S2；用框图示意如下：



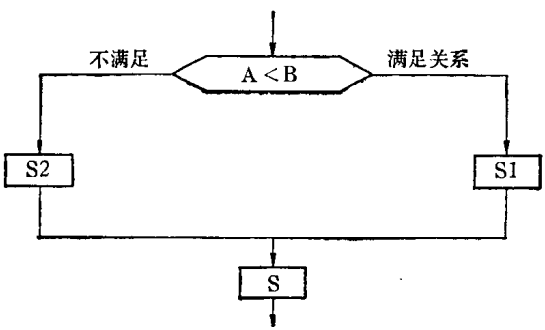
还有另一种条件语句，其一般形式为

RU A = B **ZE** S1 **FO** S2；或

RU A＜B **ZE** S1 **FO** S2；

意即如满足条件则执行语句S1，然后再跳过**FO**后面的语句，执行整个条件语句后面的语句；如条件不满足则执行语句S2，然后执行整个条件语句后面的语句。例如

RU A＜B **ZE** S1 **FO** S2；S；其意义亦可用框图示意如下：



在DJS-5算法语言中，只有两种关系，即：A＝B或A＜B，其它的一些关系符如≤、≥、≠等都可以适当地通过改写程序表示出来。例如，设有条件X≠0，当满足此条件时执行S1，不满足时执行S2，用DJS-5语言改写如下：

RU X＝0 ZE S2 FO S1；

由于S1或S2还可以是条件语句，所以通过这一句条件语句可以判断在不同的情况下执行不同的语句。

例 从三个变量A，B，C中选其最大者送到名为MAX的单元中去。

```
RU A<B ZE
        RU B<C ZE MAX:=C
           FO MAX:=B
    FO
        RU A<C ZE MAX:=C
           FO MAX:=A
```

其中虚线框中的语句又为一条件语句。

§5 循环语句

在程序设计中，循环是很重要的。算法语言中有循环语句以实现多次重复迭代。

例 $M=\sum_{k=1}^{n} k^2$

用DJS-5算法语言写出的程序如下：

```
(1)      M:=0;
(2)   DYK:=1 (1) N ZOM:=K*K+M;
```

其中(2)即为循环语句. K为循环的控制变量，1为控制变量的初值，N为终值. 括号中的1为控制变量每次增加的步长. 循环控制变量开始赋以初值，此时若不超过N，即执行ZO以后的语句，此语句称为循环体. 循环体执行完后，控制变量增加一个步长，若控制变量的值不超过终值(小于或等于终值)，循环体将重复执行，如此重复迭代，直至最后控制变量的值超过终值时，循环语句即告结束，转向下一个语句.

当步长为负时，显然控制变量的值小于终值时循环才告结束. 一般而言，循环语句的形式为

DY K:=A (B)C **ZO** S;

读作“对于K初值为A步长B终值为C做S”，其中K为循环的控制变量，S为重复执行的语句即循环体，A及C分别代表初值及终值，它们可以是一个算术表达式，此时初值和终值都是根据当时表达式算出来的值. B为步长，可以是一个数（不一定是整数），也可以是一个简单变量，即此时步长为该变量的值. 语句S本身又可以是循环语句，这样，可以实现多重循环.

例 用霍纳公式计算 $F(x)=\sum_{k=1}^{n} a_k x^{k-1}$

F:=0;

DY K:=N(-1)1 **ZO** F:=A〔K〕**×××**(K-1)+F

上例中步长为负. 控制变量出现在下标变量中作为下标，当K取值从N直至1时，下标变量分别为A〔N〕，A〔N-1〕，…，A〔1〕.

§6 复合语句、说明和分程序

复合语句 在程序设计中，有一些语句往往构成一个整