

算法语言文集

F. L. 包厄尔 K. 薩梅耳桑 等著

科学出版社

算 法 語 言 文 集

F. L. 包厄尔 K. 薩梅耳桑等著

許 孔 时 編 譯

科 学 出 版 社

1 9 6 4

內 容 簡 介

为實現程序設計自动化和开展信息加工的研究，我們需大力开展算法語言的研究。

本文集选譯了国外有关算法語言(ALGOL)的主要文献，其中包括：产生 ALGOL 60 的历史背景及經過；ALGOL 58 及有关的介紹；ALGOL 60 及有关的介紹；SMALGOL 61 及扩充 ALGOL 60 的一个方案；苏联的自动程序設計系統的輸入語言。

算 法 語 言 文 集

F. L. 包厄尔 等著
K. 德梅耳 等

許 孔 时 編 譯

*

科 学 出 版 社 出 版 (北京朝陽門大街 117 号)

北京市书刊出版业营业許可証出字第 061 号

中国科学院印刷厂印刷 新华书店总經售

*

1964 年 4 月第 一 版

书号：2895 字数：268,000

1964 年 4 月第一次印刷

开本：850×1168 1/32

(京) 0001—3,700

印张：10 3/16 插頁：1

定价：[科七] 1.80 元

序

編譯这本文集的目的是向我国讀者初步地介紹算法語言 ALGOL 及其发展情况。

近十余年来,随着电子計算机的广泛应用,产生了若干新的科学分支,其中之一就是程序設計自动化,通常簡称为程序自动化。近五、六年来,程序自动化有了很大的发展,同时逐漸形成了一个范围更广的科学分支——信息加工。随着这些发展以及为了更有效地發揮电子計算机的作用,也逐漸形成了另一个科学分支——算法語言。

关于算法語言 ALGOL 的发展簡史在(一)至(五)这五篇文章中都有所論述,这里不再重复。

各篇文章的主要內容簡單地介紹如下:

(一)所討論的是“公共語言”的一般問題;(二)是“国际代数語言”的方案,它是 1958 年討論后公布的,通常叫作 ALGOL 58;(三)是讲解 ALGOL 58 的語法和語义的;(四)的內容是 1960 年討論后公布的算法語言方案,它是从 ALGOL 58 发展来的;通常叫作 ALGOL 60;(五)是对 ALGOL 60 的介紹;(六)是 ALGOL 60 的一个扩充方案;(七)是 ALGOL 60 的一个縮減方案;(八)是苏联學者們建立的一个算法語言方案,也是对 ALGOL 60 的一种扩充。

由上述內容可見,本文集的主要內容是 ALGOL 60,它是一个較完整的目前已被广泛使用的算法語言方案,很多人在发表程序自动化方面的研究成果时都使用它。虽然 ALGOL 60 主要是用来描述数值計算过程的,但它的描写能力却不限于此。目前看来,它是研究算法語言的一个合适的模型。通过对 ALGOL 60 的分析研究,可涉及算法語言的各个方面和与編譯算法有关的許多問題。

看来,ALGOL 60 的作者們企图建立一种描述計算过程的形

式化的語言，但作为形式化的語言，ALGOL 60 是有缺陷的。首先，它的語法虽是用一些形式規則給出的，但分解用 ALGOL 60 描述的計算过程时結果却不是唯一的，这就产生了二义性問題；其次，語义不是用形式規則給出的，并且語义有不确切之处、未規定之处和互相矛盾之处。关于 ALGOL 60 在語法和語义两方面的問題，国外已有不少研究和討論，限于篇幅，本文集內未收进这方面的文章，讀者有兴趣研究这些問題时，請参看总附录 4。

讀者如果想較快地了解本文集的主要部分，可以先閱讀(四)，若觉得还不够，可以再讀(五)，这两篇都是关于 ALGOL 60 本身的。讀过(四)和(五)以后，再讀(六)或(七)时就比較容易；否則因(六)和(七)都用到(四)，故讀起来有些困难。(八)也是 ALGOL 60 的扩充，当然与(四)有关，但(八)本身是一个完整的算法語言方案，可以单独閱讀。对 ALGOL 60 的前身及其发展經過沒有兴趣的讀者，可以不讀(二)和(三)；对专门研究算法語言的讀者，(二)和(三)还是應該讀一讀的。

文集后面有四个总附录，这对研究算法語言和从事程序自动化工作的讀者可能是有帮助的，前三个总附录的內容都是关于 ALGOL 60 的語法結構的，在研究 ALGOL 60 本身和建立 ALGOL 60 的編譯算法时是自然要用到的。

本文集有不少譯名是第一次使用，当然不完全妥当。此外，原文中也有前后不一致的情形，还有些錯誤或缺漏，凡极明显之处譯者都作了改正，名詞也力求統一；不便改动者一仍其旧，讀者自己将会发现这些，譯者未一一注明。

譯者非常感謝朱三元同志，他曾仔細地校閱了全部譯稿，提了許多宝貴的意見；重編了(四)和(八)的索引，又編了本文集的索引，并且繪制了总附录 2 的“ALGOL 60 語法图”。譯者也很感謝卢慧琼同志，她曾仔細校閱了部分譯稿并提了許多宝貴的意見。

限于譯者水平，譯文中的錯誤将是难免的，敬請讀者多加指正。

許 孔 时

1963年1月

目 录

序.....	iii
(一)公共語言問題.....	F. L. 包厄尔 K. 薩梅耳桑 (1)
(二)初步报告——国际代数語言.....	A. J. 珀利士 K. 薩梅耳桑 (11)
(三)苏黎世 ACM-GAMM 會議提出的国际代数語言的語 法和語义.....	J. W. 巴科斯 (35)
(四)关于算法語言 ALGOL 60 的报告.....	P. 璫尔等 (54)
(五) ALGOL 介紹	H. R. 施瓦茲 (98)
(六) ABS 12 ALGOL: 为工业問題用的 ALGOL 60 的一 个扩充方案.....	R. W. 霍克奈 (136)
(七) SMALGOL 61	G. A. 巴彻洛尔 J. R. H. 德姆斯特 D. E. 克努斯 J. 斯皮罗尼 (158)
(八)程序自动化系統的輸入語言.....	...A. П. 叶尔紹夫 Г. И. 科茹欣 Ю. М. 沃洛申 (168)
总附录 1. ALGOL 60 的元語言公式总表	(262)
总附录 2. ALGOL 60 的語法图解	(268)
总附录 3. ALGOL 60 的語法分析	(270)
总附录 4. ALGOL 的文献目录.....	(298)
后記.....	(305)
ALGOL 60 报告的附件	M. 伍哲 (307)
中英名詞对照表.....	(314)
中俄人名对照表.....	(318)
中英人名对照表.....	(318)
英中人名对照表.....	(319)

(一) 公共語言問題*

特別是科學數值工作的公共語言問題

(蘇黎世 ALGOL 會議的動機、限制、目的和結果)

包厄爾 (F. L. Bauer) 薩梅耳桑 (K. Samelson)

為了通訊方便和準確，長期以來就感到需要一種公共語言。用戶協會(歇爾,尤斯)¹⁾的存在就說明了此點，但這些協會只是在它們特殊的計算機語言的基礎上解決了通訊問題。由於實際上的考慮，選擇這種公共語言時顯然不能只適用於一種或多種現有計算機的指令代碼而不顧其他使用不同代碼的計算機。無疑地，對一些通訊的技術問題（包括把一個公共語言譯為每一種計算機的特殊語言）而言，面向計算機的通用語言 (UNCOL²⁾) 是重要的；但蘇黎世會議認為數值分析和科學計算用的公共語言應尽可能地接近通常的數學表示法，這種表示法是已經普遍使用了的。公共語言應象 FORTRAN³⁾ 那樣，是可構造運算的語言（這就是說它應當依時間先後可構造地定義出運算的序列），使計算機或者能執行已給的指令，或者能機械地把它們譯為計算機語言。這樣就去掉了所有隱式的數學定義；此外，會議特別注意到用自由構造的單用數值過程的定義代替子程序和圖書館程序的可能性。於是所要的

* 本文譯自 *INFORMATION PROCESSING, Proceedings of the International Conference on Information Processing, UNESCO, Paris, 15—20 June (1959), 120—124.*

- 1) 用戶系指電子計算機的用戶，歇爾 (SHARE) 和尤斯 (USE) 都是用戶協會的名字——譯者注。
- 2) UNCOL 是 universal computer-oriented language (面向計算機的通用語言) 的縮寫——譯者注。
- 3) FORTRAN 是 formula translation (公式翻譯) 的縮寫，這是美國一個早期程序自動化系統的名字——譯者注。

語言就好象是魯提施豪塞爾 (H. Rutishauser) 早先想法中的那種算法語言 (ALGOL¹⁾) 了, 因此這是面向問題的語言。會議的工作是把數學表示法標準化並把它擴充為完全可運算的語言。

1. 建立 ALGOL 的動機

1958 年 5 月 ACM²⁾ 的語言特別委員會和 GAMM³⁾ 的程序設計委員會雙方的代表在瑞士蘇黎世舉行了一次會議。這次會議成功地建立了一種為科學數值工作用的算法語言 (ALGOL), 打算把它作為數值分析的公共語言並在科學計算中使用它。會議報告已同時在“數值數學”(Numerische Mathematik) 和“ACM 通訊”(Communications of the ACM) 兩種雜誌上發表。它是 ALGOL 的初步報告, 並準備在形式和內容兩方面作可能的修正和改進。它的基礎是幾個小組獨立的周密的準備工作的結果, 這些小組既代表了計算機的設計師、製造商和用戶, 也代表了從事數值分析工作的數學家。

觀點和方法很不相同的這些小組能得到折衷方案這件事是一個顯著的進步。這樣每個小組都發展自己的語言的真實危險就消失了, 而是更好地朝着公共目標進行競賽。既然在計算機領域內為了通訊方便和準確早已感到需要一種公共語言 (特別是關於數值分析和科學計算方面的需要), 所以希望 ALGOL 能被廣泛地接受。

2. 可構造運算的語言

因為在計算機領域內公共語言不僅是通訊的手段, 也是給計算機指令的手段, 所以公共語言必須是可構造運算的語言。為了 ALGOL 怎樣適合這一要求, 有必要先分析一下這個概念。可構造運算的語言 (OPCOL⁴⁾) 所處理的是可構造的運算及其運算

1) ALGOL 是 algorithmic language (算法語言) 的縮寫——譯者注。

2) 美國的計算機協會的縮寫——譯者注。

3) 德意志聯邦共和國的应用數學和力學協會的縮寫——譯者注。

4) OPCOL 是 operational-constructive language (可構造運算的語言) 的縮寫——譯者注。

对象，它們二者都用某些符号标識出来。OPCOL 是一个形式系統，它具有形成可允許的符号模式(用以表示在給定对象上进行已知运算的可构造的方案)的語法規則。

任何計算机語言都是可构造运算的語言，对象是存儲单元和寄存器中已編碼的内容，运算是这些数据的标准变换。現有的可构造运算的語言在对象和运算的范围方面是不同的。在最簡單的情况下，我們有数值对象(用数字表示、或用变量名字的符号表示)和算术运算；或者邏輯对象(真假值)和命題邏輯的运算。在更进一步的情况下，对象可以是代数的、集合論的或形式邏輯的形式表达式，而运算是推导、检查是否相等或展开。也有可构造运算的語言的例子，其对象是函数表达式，而运算是形式微分或形式积分。

給 OPCOL 定义語法規則的最簡單的和最自然的方法是由它的运算結構来定义。但相同的运算結構可用不同的表示法来表示，我們知道在数学和形式邏輯方面就是如此。举一个例，我們考虑在通常算术表示法中的表达式

$$((a + b) \times (a - b) + c) \times d,$$

在函数表示法中以

$\Sigma(x, y)$ 表示 $y + x$ 的和,

$\Delta(x, y)$ 表示 $y - x$ 的差,

$\Pi(x, y)$ 表示 $y \times x$ 的积。

于是前式就变为

$$\Pi(d, \Sigma(c, \Pi(\Delta(b, a), \Sigma(b, a)))).$$

在算子表示法中，如羣論中用到的那样，定义

A_a 为加 a ,

S_a 为減 a ,

M_a 为乘 a ,

于是我們得到

$$M_d A_c M_{S_b a} A_b a.$$

在形式邏輯中用华沙无括号表示法就有

$$\Pi d \Sigma c \Pi \Delta b a \Sigma b a,$$

它叫做簡單語言^[1]。

函数表示法和算子表示法显然是同构的，并且可立即縮減为一种簡單語言。另一方面，通常的算术表示法有优先規則，这使得它的运算結構复杂起来。和大多数現有的表示法相同，不顾二維表示法中复杂的結構变得更清楚这一事实，这些規則都是順序的¹⁾。不幸，由于技术上的原因更高維的表示法通常都被排斥掉了。在所有的情况下用递归的方法能最有效地描述語法規則。在函数表示法的情况，形式的构成規則是容易用插入法来表示的，在华沙表示法中則用把符号行聚在一起的方法。二者都表明在发展形式語言时可能有若干不明确的地方(非本質的)。

3. 可构造运算的語言的限制考虑

由

$$q_1' = q_1 + e_1,$$

$$q_1'e_1' = e_1q_2,$$

$$e_1' + q_2' = q_2 + e_2,$$

$$q_2'e_2' = e_2q_3,$$

$$e_2' + q_3' = q_3$$

表示的靜态的相等。这个方程組沒有运算意义。可以把有'的量解出来，这就給出結構

$$q_1' := q_1 + e_1,$$

$$e_1' := (e_1q_2)/q_1',$$

$$q_2' := q_2 + e_2 - e_1',$$

$$e_2' := (e_2q_3)/q_2',$$

$$q_3' := q_3 - e_2',$$

这里記号:= 的意义是“被定义为”或“是…的結果”，它有动态的定向的意义。但也可把无'的量解出来而得到

$$q_3 := e_2' + q_3',$$

$$e_2 := (q_2'e_2')/q_3,$$

$$q_2 := e_1' + q_2' - e_2,$$

1) 意即写成一行的，一維的——譯者注。

$$e_1 := (q_1' e_1') / q_2,$$

$$q_1 := q_1' - e_1.$$

在有些情況下(例如這裡的情況),我們可以从上下文成功地判定明顯的計算方法的含義。但一般象

$$“x \text{ 使得 } x^3 + a_1 x^2 + a_2 x + a_3 = 0”$$

這樣的隱式定義,在給定的數值對象和算術運算的範圍內就不能確定可構造的過程。事實上,首先要解決存在問題和唯一性問題,並且甚至是唯一存在的結果也可能用不同的逼近方法得出來。因此我們要从 OPCOL 中把甚至是很平常的隱式定義都除掉。在數值運算的範圍內,等號“=”只在已構造好的對象間敘述一個條件時才有定義。

4. 可構造運算的語言的目的

OPCOL 可包括自由的定義,這種定義指明用某些新的特殊的字母序列作為在 OPCOL 內已定義好的其它序列的縮寫。為了實際上避免龐雜重複,可以使用這種定義。然而從理論觀點看來,有興趣的是去確定 OPCOL 中哪些複雜的指令能由較簡單的運算用自由的定義表示出來。因此可以考慮求相應的原始 OPCOL 的問題,在這個 OPCOL 中任何運算都不能用此 OPCOL 中的其它運算表示出來。這對於研究利用學習裝置把一種 OPCOL 譯為另一種 OPCOL (例如計算機語言)的理論是重要的。但我們暫時還不應當指望這種學習裝置去模擬直觀。看來基於學習裝置的翻譯程序是否聰明是和建立翻譯程序的人的指導有關。

上述的最後一點說明使我們必須考慮 OPCOL 的目的。為理論研究用的 OPCOL 可以完全獨立地發展;而實際應用的 OPCOL 是在一定的應用範圍內通訊和表達指令的手段,它被認為是這個範圍內的公共語言。因此它的結構、符號和表示方法都要受這些情況的影響。

機器語言 (ML¹⁾) 從其本義來看是可構造運算的語言,其中運

1) ML 是 machine language (機器語言)的縮寫——譯者注。

算对象是存储单元和寄存器中已编码的内容，算子是这些数据的标准变换。现有的计算机有多少差异，这些机器语言就有多少差异。因此在一定程度上接近计算机而又使通讯成为可能时，面向计算机的通用语言（UNCOL）可能是有实际价值的。最近关于以假设的 UNCOL 为基本语言的研究^[2]表明，甚至在处理把 OPCOL 翻译到特殊的 ML 时也有种种有效的应用。这表明 UNCOL 的概念和其它更面向问题的语言并不是对抗的。

这些面向问题的公共语言（POL¹⁾）的通用性是有限制的，因为在不同的领域内被陈述的问题的结构、表示法和习惯方法的符号化是很不同的，它们受着内部结构上的特点和长期习惯的支配。在与数学处理接近的所有领域中这种情况是很牢固的，通用性在这里甚至要涉及两个方面。特别在导致数值计算的科学领域和其他领域中有这样一大类问题，它们通常的叙述方法很受数学结构、表示法和符号化的影响。因此为科学数值计算用的 POL 除了缺少标准化之外，在很大程度上已经事先确定了。在经济问题的数据处理中，在数学结构的处理中，在陈述自然语言的翻译过程中或在陈述 OPCOL 彼此之间的翻译过程中，其它的 POL 可能都是有用的。显然不能希望有面向问题的通用语言存在。

5. 对公共数学语言的一般要求

因为完全通用的语言可能是得不到的，所以苏黎世会议把它的目的严格地限制在为科学数值计算用的面向问题的公共语言方面，并以下列几点为主要目标：

- 1) “这个新语言应尽可能地接近标准的数学表示法，并且稍加解释就可理解”。
- 2) “在出版物中应当能用它描述计算过程”。
- 3) “这个新语言应当能机械地译成机器程序”。

1) POL 是 problem-oriented common language（面向问题的公共语言）的缩写——译者注。

在以前的嘗試中,例如 FORTRAN 和 MATH-MATIC¹⁾ 中,这些要求中的大部分已被滿足了,至少是部分地被滿足了。把此点与苏联从事程序自动化的小組所列出的目标^[3]相比是很有兴趣的,它們的目标是:

- a) “初始信息的表示必須接近于問題的数学形式的表示”。
- b) “与数学形式的表示无关的輔助工作和技术工作的規模必須減至极少”。
- c) “初始信息应給出关于結果程序的結構的全部信息”。
- d) “初始信息必須尽可能地紧凑和容易看”。

从这种相似的情况可以期望莫斯科苏联科学院在“編制程序的程序”方面的下一步工作将很接近于 ALGOL。事实上,这两組要求在本質上是一致的。莫斯科苏联科学院的 PPS²⁾ 在結構上相应地等价于一个基本的 ALGOL,只是在一定的程度上包含着类似于通常編譯程序技巧的更面向机器的語言。在这方面苏黎世會議并未探討,因为在討論面向計算机的通用語言 (UNCOL) 时有适当的地方会考虑这种問題的。

6. ALGOL 应滿足的具体要求

6.1. 結構上的要求

对于面向問題的計算語言來說,主要对象显然是算术运算所联接的数。因为依赖于二元关系的判定是不可避免的运算,所以要加上以真假值为对象的布尔代数的运算,虽然这并不是严格必須的。因此需要若干不同的运算和若干类包含各种符号表示的对象(例如变量、关系式、布尔变量)。这些运算的构成規則用到熟知的递归方法,例如在数論基础中就是如此。

自然語言中句子的概念在 ALGOL 中对应于象算术公式那样

1) 这是一个自动編碼系統的名字——譯者注。

2) 这是苏联科学院計算中心箭牌計算机上的編譯程序的名字英文縮写——譯者注。

的自封結構,叫作語句。对应于段、节、章、卷等則可用不相交的括号結構形成复合語句,这为标出 ALGOL 程序的某些部分提供了唯一的方法。

用自由的定义給这个語言加上新成分的可能性被認為是重要的。这些定义可以作为 ALGOL 中或其它 OPCOL 中表示的某些运算序列的縮写。只当附有定义語句时这些定义才有意义,因此,实际上它們和这語言的基本符号是不等价的。會議并未企图为此語言中引入新的基本符号提供便利而創立元語言,但这并不是說會議不贊成研究这种更高級的語言。

6.2. 形式上的要求

面向問題的公共語言不仅应当在构造上而且也应当在形式上遵守普遍接受的習慣。这就是說,在 ALGOL 中用来代表这語言中的抽象符号的字符应当根据需求和可能,尽量是数学中共同使用的那些字符,假定这样作不导致含混的話。其次,在沒有标准的習慣表示法和現有語言的符号相平行的情况下,只要可能,这个語言就应当是能自己解释自己的。这就是說,这种符号应当是自然語言中指明这些符号的意义的那些字。

6.3. 技术上的要求

最后,对于象 ALGOL 这样的語言,有某些很重要的技术上的特点是要考慮到的。原因之一(事实上是主要的原因)就是,对于发展 ALGOL,我們要的是一种容易了解的并能机械地翻譯的程序設計語言。只有在用这个語言写程序的同时就能机械地譯碼而无需人工的中間記錄,才能真正达到上述的目的。因此,表示这个語言时用到的字符应当是大量生产的譯碼装置(如穿孔帶或穿孔卡片)上所所有的字符,这就是說字符的总数是受到严格限制的。

另一方面,可用的字符之总数是随着机器的类型而变化的,并且即将出現的新机器中,这个总数可能有相当的增加。为此,記載 ALGOL 的文献中用到的一組字符实际上是譯碼机上現在用到

的和想要用到的(印书时也用它)两者之間的一个折衷方案。这个語言主要是供参考目的之用,而且不因在机器上的使用而对字符的选择有偏見,为了明确地指出此点,这个語言就叫作“参考語言”。

但参考語言的字符是这样选择的,它們中間那些最重要的字符是現有机器上已有的字符,或者是能用这些已有字符构造出来的字符。因此,为了实用,只要行得通,就建議尽量如此。同时这包含着建立“机器小組”的愿望;即建立一些使用同样的譯碼机器的小組,它們使用的算法語言在形式上直到所用的一組字符都是相同的。这样,在小組的全体成員之間就可无需人的干預而自由地交換程序。欧洲使用电传打字譯碼装置 CCIT2 的 ZMMD¹⁾ 小組的那些成員已經开始这样做了。其次,在可能要分行的情况下,現有的机器要求选择一种严格地綫性的表示法,因为有了这个限制,处理出版物中的下标和指数时就不实用,所以要把参考語言的字符选得使之能直接譯成出版語言中的下标和上标表示。

7. ALGOL 的其它特点

算术运算加上依赖于布尔表达式的条件以及用綫性形式描述进程图的多种連接关系的其它运算,对 ALGOL 的直接目的而言已經形成了一个足够充分的原始系統。此外,ALGOL 包含着叫做“說明”的元素,它只給出关于对象的性質的附加信息,例如数的特性(整数、双倍精确度、复数、布尔量)或数組的維数。也还有“函数”的定义和叫作“单用过程”的更复杂的过程的定义,它們相应于通常的程序設計中的图书馆程序。

从实用中考虑,感到还需要給这个語言加上某些多余的运算元素,它們是能在这原始語言中表达的过程的縮写。这些元素是:

- 1) 在已知变量中描述递归的符号,

1) ZMMD 是欧洲一个小組的名字,它是由 Zürich (苏黎世), Mainz (美因茲), München (慕尼黑)和 Darmstadt (达姆施塔得)四个城市的一些人組成的——譯者注。

2) 描述多次选择和传递根据算术计算得到的标号的符号。

3) 中断这语言中句子的线性进程的符号,它有效地导致嵌入已单独写好的某个句子序列,同时允许代入。

这些就总结了我們关于 ALGOL 的意见,关于它的语法结构的具体叙述請参看 J. W. 巴科斯的文章。

参 考 文 献

- [1] Rosenbloom, P. C., The elements of mathematical logic, New York, 1950.
- [2] Strong, J., et al., The problem of programming communication with changing machines, *Comm. ACM*, 1 (1958).
- [3] Ershov, A. P., The work of the computing centre of the Academy of Sciences of the USSR in the field of automatic programming, Lecture, Symposium on the Mechanization of Thought Processes, Teddington (England), November 1958.

(二) 初步报告——国际代数語言*

珀利斯(A. J. Perlis) 薩梅耳桑(K. Samelson)

注。为了及时传播 ACM-GAMM 委员会关于代数程序设计語言的工作成果,乃提出本报告。所描述的語言很自然地是一个折衷产物,但这个折衷产物对不同的爱好考虑得多一些,而对内容或基本概念考虑得较少。虽然如此,它却为表示一大类算法提供了一个自然而简单的媒介。本报告的錯誤和矛盾之处尚未彻底检查。預計委员会将准备一份此語言的更完整的描述以便出版。

ACM-GAMM 委员会的

A. J. 珀利斯

K. 薩梅耳桑

第一部分 序 論

1955 年,作为达姆施塔得电子计算机会议的结果,德意志联邦共和国的 GAMM (应用数学和力学协会) 成立了一个程序设计委员会。其后一个小组委员会在公式翻译和建立翻译程序方面开始工作,在这方面作了大量的工作。

为检验便于交换所有类型计算知识的方法和手段,于 1957 年 5 月 9 日和 10 日在洛山磯召开了一次由 USE, SHARE 和 DUO 这些组织及计算机协会 (ACM) 的代表参加的会议。这些会议参加者特别感到,一个单一通用的计算机语言是非常需要的。事实上,

* 本文译自 *Communications of the Association for Computing Machinery*, 1 (1958), No.12, 8—22.