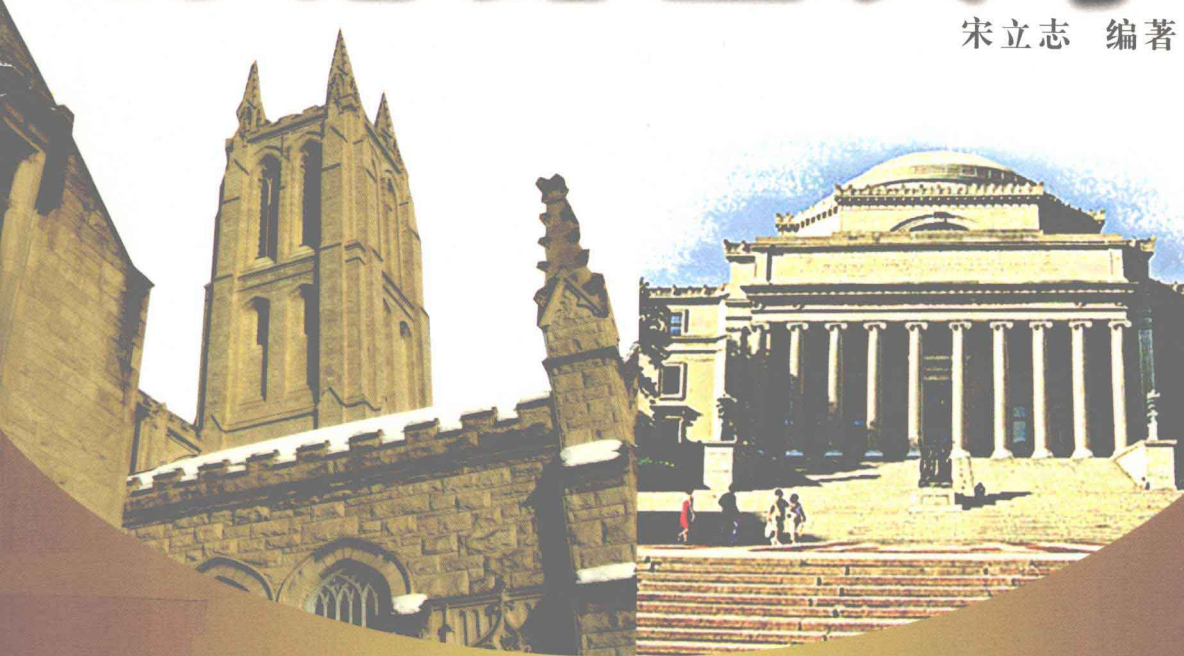


名校精英

MINGXIAO JINGYING

芝加哥大学 哥伦比亚大学

宋立志 编著



本书介绍他们凭借自己的勤奋刻苦、聪明智慧和坚持不懈的努力赢得了举世瞩目的非凡成就，为发展祖国的科学教育事业，为推进世界科学技术进程作出卓越的贡献。

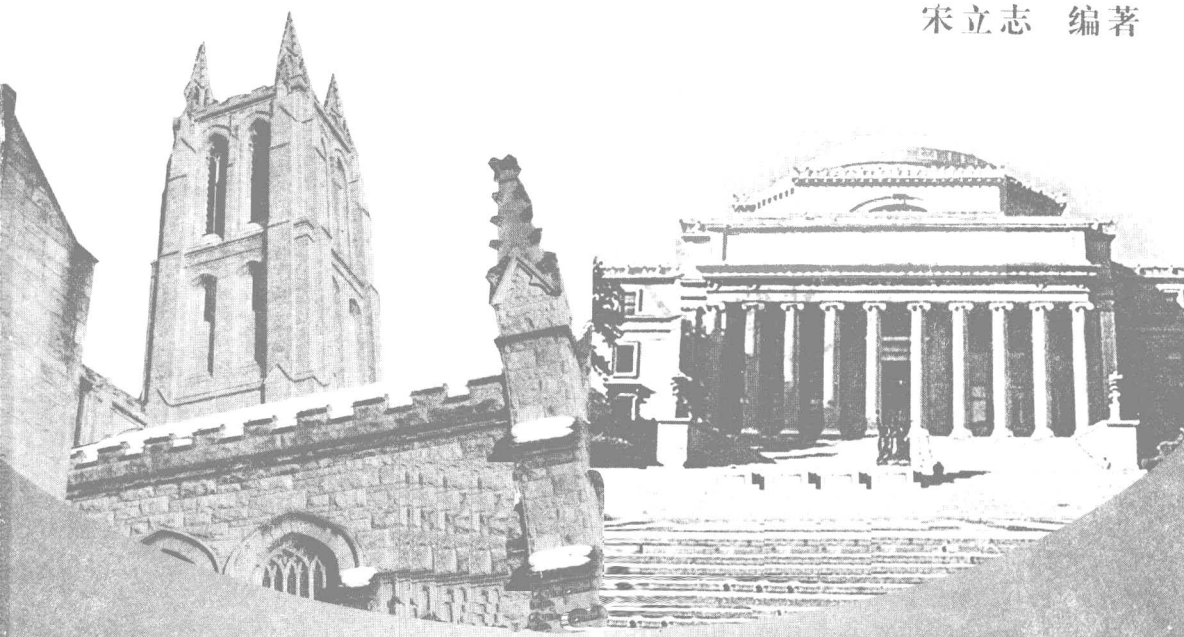
京华出版社

名校精英

MINGXIAO JINGYING

芝加哥大学 哥伦比亚大学

宋立志 编著



本书介绍他们凭借自己的勤奋刻苦、聪明智慧和坚持不懈的努力赢得了举世瞩目的非凡成就，为发展祖国的科学教育事业，为推进世界科学技术进程作出卓越的贡献。

京华出版社

图书在版编目 (CIP) 数据

芝加哥大学、哥伦比亚大学 / 宋立志编著. —北京: 京华出版社, 2010. 7
(名校精英)

ISBN 978-7-80724-934-4

I. ①芝… II. ①宋… III. ①芝加哥大学—概况②哥伦比亚大学—概况
IV. ①G649.712.8

中国版本图书馆 CIP 数据核字 (2010) 第 108667 号

名校精英：芝加哥大学、哥伦比亚大学

编 著 宋立志

出版发行 京华出版社

(北京市朝阳区安华西里一区 13 楼 2 层 100011)

(010) 64258473 64255036 84241642 (发行部)

(010) 64259577 (邮购、零售)

(010) 64251790 64258472 64255606 (编辑部)

E-mail: jinghuafaxing@sina.com

印 刷 北京昌平新兴胶印厂

开 本 787mm×960mm 1/16

字 数 230 千字

印 张 15 印张

印 数 1—5000

出版日期 2010 年 7 月第 1 版 第 1 次印刷

书 号 ISBN 978-7-80724-934-4

定 价 298.00 元 (全 10 册)

京华版图书, 若有质量问题, 请与本社联系



目 录

芝加哥大学

芝加哥大学简介	(3)
理查德·哈明	(5)
罗伯特·弗洛伊德	(9)
周 培 源	(13)
庄 长 恭	(23)
李 景 晟	(30)
顾 翼 东	(33)
蔡 镛 生	(37)
阿瑟·荷里·康普顿	(41)
克林顿·约瑟夫·戴维森	(45)
罗伯特·安德罗·米利肯	(48)
詹姆斯·W·克罗宁	(52)
乔治·约瑟夫·斯蒂格勒	(55)
蒂奥多·威廉·舒尔茨	(59)
赫尔伯特·A·西蒙	(63)
米尔顿·弗里德曼	(66)
索尔·贝娄	(71)
查尔斯·布伦顿·哈金斯	(75)
赫伯特·查尔斯·布朗	(77)
罗伯特·桑德逊·马利肯	(81)
约翰·格里格·汤普逊	(83)
亚历克西斯·卡雷尔	(86)
阿尔伯特·A·迈克尔孙	(89)
威拉德·弗兰克·利比	(93)
杨 振 宁	(96)
李 政 道	(102)



谢 玉 铭	(108)
林 巧 稚	(112)

哥伦比亚大学

哥伦比亚大学简介	(123)
约翰·杜威	(125)
威廉·赫德·克伯屈	(147)
王 天 眷	(162)
约翰·巴克斯	(166)
任 鸿 隽	(172)
罗尔德·霍夫曼	(176)
陈 楨	(178)
王 守 竞	(181)
R. 朗兰兹	(183)
巴茹·贝纳塞拉夫	(186)
塞缪尔·巴鲁克·布卢姆伯格	(189)
康拉德·萨查理雅斯·洛伦兹	(192)
康拉德·E·布洛赫	(195)
阿瑟·肖洛	(200)
詹姆斯·雷恩沃特	(204)
里昂·N·库柏	(206)
哈罗德·克莱顿·尤里	(208)
小狄金森·武德茹夫·里查兹	(210)
安德烈·弗雷德里克·库南德	(213)
爱德华·卡尔文·肯德尔	(215)
波利卡普·库什	(219)
小威利斯·E·拉姆	(221)
伊西多·依塞克·拉比	(223)
吴 健 雄	(227)

芝加哥大学



芝加哥大学简介

芝加哥大学 (University of Chicago) 是美国中西部最富盛名的私立大学之一, 由石油大王 J. D. 洛克菲勒捐资于 1981 年创办, 位于美国伊利诺州的芝加哥市的密歇根湖湖畔。校园占地约 200 英亩, 其哥特式建筑古香古色。芝大提出的旨在防止学术课程和职业课程过分专门化的“芝加哥计划”, 对其他大学的本科教育计划产生巨大影响。



芝加哥大学

芝加哥大学的校训是“提升知识, 以便充实人生。”(拉丁文原文: *Crescat scientia vita excolatur*。英译: *Let knowledge grow from more to more; and so be human life enriched*)。

芝大的学术单位包括大学学院部、4 所研究生院, 7 所专业研究学院、大学进修部、芝大出版社、图书馆及各学术中心、委员会和研究所等。

芝大学院部是小型的私立寄宿学校, 用于培养本科生, 学制 4 年, 授文学士和理学士学位。该部由 5 个学院组成, 它们是生物科学、人文学科、自然科学、社会科学和新大学学院; 除上述研究生院外, 芝大还有以下专业研究学院。神学院、法学院、商学研究生院、普利策医学院、欧文·哈里斯公共政策研究生院、社会服务管理学院。

自从创建以来, 芝大在许多领域都作出了杰出贡献, 为美国和全世界培养了许多杰出人才。芝加哥大学有 75 位校友曾获诺贝尔奖, 为美国大学中



最多。2 位校友曾获图灵奖,12 位教授荣获过国家科学奖章,教授中有超过 60 位国家科学院、国家工程院和国家医学科学院院士。我国著名的科学家学者也曾在此求学或任教,其中有周培源、庄长恭、顾翼东。

芝大对教育观念的“宏观”与实验精神,奠定了它在美国教育史上的重要地位;而它在学术研究上的地位与贡献,也同样值得称道。芝加哥大学的人类学、地球科学、经济学、地理学、历史学、语言学、统计学、社会学等学科专业在美国具有较强的学术实力。经济学、社会学实力尤其超强,“芝加哥学派”赫赫有名。在经济学方面,芝加哥大学的经济系产生了 22 位诺贝尔奖得主,弗里德曼,戴维德等学者坚持自由主义而被誉为“芝加哥学派”,对经济学的研究有着深远的影响。一九八二年该校的经济学家,同时也教授 MBA 课程的 George J. Stigler 荣获诺贝尔经济奖,是全球商学院教授获此殊荣的第一人。



理查德·哈明

一提起“哈明码”，恐怕很少有人不知道的。这种能找出并纠正数据块在传输过程中出现的错误的编码方法，对于计算机技术和通信技术来说真是太重要了。发明这种编码技术的理查德·哈明（Richard Wesley Hamming, 1915—1998）因此而获得了第三届即1968年度的图灵奖。

哈明1915年2月11日生于芝加哥。1937年在芝加哥大学获得数学学士学位，1939年在内布拉斯加大学获得硕士学位，接着又于1942年在伊利诺伊大学获得博士学位，成为一名数学专家。学成以后，他留校工作两年，然后转入肯塔基州位于俄亥俄河畔的路易斯维尔大学任教，两年后来到洛杉矶阿拉莫斯国家实验室，参与了著名的曼哈顿计划。但在那里哈明也只呆了两年，就又转到贝尔实验室工作。正是在这里，哈明遇到了他感兴趣和能发挥他特长的课题，也有一个适宜的工作环境，因此一干就是30年（1946—1976）。这期间，他曾长期担任贝尔实验室计算机科学部的主任。1976年他离开贝尔，到美国海军研究生院（Naval Postgraduate School，在加利福尼亚州的蒙特雷）工作，直到1997年82岁高龄时才退休，第二年1月7日去世，享年83岁。

哈明到贝尔实验室后接受的第一个任务就是解决通信中令人头痛的误码问题。通信时发送方发出的信息在传输过程中由于信号的衰减和外界的电磁干扰，到接收方产生了畸变和失真，获得的是错误的信息。这在商业、军事等应用中都会产生严重的后果，有时简直会祸国殃民，因此迫切需要加以解决。但在相当一段时间里，这成了摆在许许多多科学家和工程师面前的一大难题，谁也找不出解决的好办法。哈明接受这个任务以后，意识到通信线路质量的改善是有限度的，外界干扰是客观存在也无法绝对避免，因此这个问题不可能通过让发送的代码不出错这条途径去解决，而只能通过一旦出错如何发现、如何纠正才能解决。这使哈明的研究沿着正确的路线进行。经过深入探讨，1947年哈明终于发明了一种能纠错的编码，这种码就叫“纠错码”（error-correcting-code）或“哈明码”（Hamming code）。哈明码是一种冗余码，即在有效信息代码中要加入校验位，这是为纠错而必须付出的代价。其基本原理是使每一信息位参与多个不同的奇偶校验（parity check）。所谓奇偶校验是在代码中设置一个校验位，通常置于代码的最左边。若整个代码中



“1”的个数为奇数认为代码正确,称为奇校验(odd check);反之,若整个代码中“1”的个数为偶数认为正确,则称为偶校验(even check)。哈明码就是有多个奇偶校验位的一种代码。在适当安排下,通过这多个奇偶校验位就可以检查出代码传送中的错误并自动纠正。一般而言,对于长度为 n 位的代码,其中应包括 r 个校验位,有效信息位为 $n-r$, r 的值应满足以下公式:

$$2^r - 1 \geq n$$

下面我们举一个例子简单说明哈明码的原理。以7位字组的二进制编码的十进制数的传送为例,根据以上公式,有效信息为4位,校验位为3位。安排3、5、6、7四位为信息位,而1、2、4三位为校验位,如下表所示。

发送时,信息位的内容当然是根据所要发送的十进制数是几而定的,1、2、4三个校验位的内容是按以下规则自动生成的:

校验位1:由1、3、5、7四位的偶校验决定校验位1的内容;

校验位2:由2、3、6、7四位的偶校验决定校验位2的内容;

校验位4:由4、5、6、7四位的偶校验决定校验位4的内容。

也就是说,比如对校验位1,若3、5、7三位中“1”的个数为奇数,则校验位1置为“1”;若3、5、7三位中“1”的个数为偶数,则校验位1置为“0”,其余类推。

十进制数	位 置						
	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0
1	1	1	0	1	0	0	1
2	0	1	0	1	0	1	0
3	1	0	0	0	0	1	1
4	1	0	0	1	1	0	0
5	0	1	0	0	1	0	1
6	1	1	0	0	1	1	0
7	0	0	0	1	1	1	1
8	1	1	1	0	0	0	0
9	0	0	1	1	0	0	1

这样形成的7位代码发送出去以后,若到了接收方发生错误,就能检测出来并可自动纠正。举例说,发送的数是“6”,应为1100110,但接收到的却



是 1110110, 则通过对上述三组 4 位代码的偶校验, 发现第 1 和第 2 两组中“1”的个数都为奇可断定发生错误; 错的是哪一位呢? 这可通过如下办法确定: 哪一组的偶校验通过, 记为 0; 偶校验出错, 记为 1, 第一组到第三组按从右到左的次序排列所形成的二进制数就确定了出错列的位置。这里是“011”, 即 3, 可断定左起第 3 位出了错, 把它反过来 (这里是把“1”变成“0”) 就是了。同理, 若接收结果为 1100111, 则三组偶校验均出错, 记为“111”, 指明第 7 位出错, 把它反过来即可。

大家看, 多么巧妙! 当然这个例子仅仅是最简单的情况。现在, 包括哈明码在内的整个编码学已建立在十分复杂而严格的数学理论基础之上, 要用到抽象代数 (abstract algebra), 包括伽洛瓦理论 (Galois theory) 等。

哈明码的发明是为了解决通信中的误码问题, 但对计算机同样有用。因为计算机的 CPU、内外存、各种外部设备之间的代码传送同样存在着误码的可能。例如, 计算机的存储器差错校验 (memory error checking and correction) 就常常采用哈明码校验。在计算机联成网络的情况下, 数据通信的可靠性问题更为突出。ACM 在将图灵奖授予哈明的 1968 年, 计算机网络的研究刚刚开始不久, Internet 的始祖 ARPANET 是 1969 年才将最早的 4 个站点连通的。从这点看, ACM 在图灵奖的评奖中是很有远见的。

作为一名数学家, 哈明的专长是数值方法、编码与信息论、统计学和数字滤波器等。这些学科中有不少名词术语是由哈明定义, 因此而用哈明命名的, 除“哈明码”外, 常见的还有:

“哈明间距” (Hamming distance): 这指同样长度的两个码中, 对应位不同的码的个数。比如 01010 和 11001, 哈明间距为 3。

“哈明权” (Hamming weight)。这指代码中 1 的个数。如 01110 的哈明权为 3。

“哈明窗口” (Hamming window)。这指一种滤波器的通频带, 其传递函数的解析式为:

$$W(n) = \begin{cases} 0.54 - 0.46 \cos\left(\frac{2\pi n}{l-1}\right), & 0 \leq n \leq l-1 \\ 0, & n \text{ 为其他值} \end{cases}$$

哈明的论著颇多, 主要有:

《科学家和工程师用的数值方法》(Numerical Methods for Scientists and Engineers, McGraw-Hill, 1973, 第 2 版)

《数字滤波器》(Digital Filter, Prentice-Hall, 1977, 1983, 1989)

《编码和信息论》(Coding and Information Theory, Prentice-Hall, 1980, 1986)



《用于微积分、概率论和统计学的数学方法》(Methods of Mathematics Applied to Calculus, Probability, and Statistics, Prentice - Hall, 1985)

《计算机与社会》(Computers and Society, McGraw - Hill, 1972)

《实用数值分析导论》(Introduction to Applied Numerical Analysis, Hemisphere Pub. , 1989)

《概率论的技巧》(The Art of Probability, Addison - Wesley, 1991)

《从事科技工作的技巧》(The Art of Doing Science and Engineering, Gordon and Breach Science Pub. , 1997)

哈明有一句名言:“计算的目的在于数据,而在于洞察事物”(“The purpose of computing is insight, not numbers”)。此外,他还非常欣赏孔子的话:“学而时习之,不亦悦乎”,把这句话印在他著的《科学家和工程师用的数值方法》那本书的卷首作为座右铭(英文是 To study, and when the occasion arises to what one has learned into practice——is that not deeply satisfying?)。纵观哈明的一生,他自己就是实践这两句话的一生。

哈明是美国工程院院士,1958—1960 年曾出任 ACM 的第七届主席。除获得图灵奖外,1979 年他获得 IEEE 的 Piore 奖,1981 年获得 H. Pender 奖(这是宾夕法尼亚大学所设立的一个奖项),1996 年获得 Rhein 基金会奖。有趣的是,IEEE 设立了一种以哈明命名的奖章,1991 年把这种奖章颁给了哈明本人。

哈明在接受图灵奖时发表了题为“我对计算机科学的看法”(One Man's View of Computer Science)的演说,刊载于 Journal of ACM,1969 年 1 月,3~12 页,也可见《前 20 年的图灵奖演说集》(ACM Turing Award Lectures——The First 20 Years, 1966~1985, ACM Pr.), 207~218 页。



罗伯特·弗洛伊德

历届图灵奖得主基本上都有高学历、高学位，绝大多数有博士头衔。这是可以理解的，因为创新型人才需要有很好的文化素养，丰富的知识底蕴；因而必须接受良好的教育。但事情总有例外。1978 年图灵奖获得者、斯坦福大学计算机科学系教授罗伯特·弗洛伊德 (Robert W. Floyd) 就是一位“自学成才的计算机科学家”(a Self-Taught Computer Scientist)。



罗伯特·弗洛伊德

弗洛伊德 1936 年 6 月 8 日生于纽约。说他“自学成才”并不是说他没有接受过高等教育，他是芝加哥大学的毕业生，但学的不是数学或电气工程等与计算机密切相关的专业，而是文学，1953 年获得文学士学位。20 世纪 50 年代初期美国经济不太景气，找工作比较困难，因学习文学而没有任何专门技能的弗洛伊德在就业上遇到很大麻烦，无奈之中到西屋电气公司当了一名计算机操作员，在 IBM650 机房值夜班。我们知道，早期的计算机都是以批处理方式工作的，计算机操作员的任务就是把程序员编写好的程序在卡片穿孔机(这是脱机的辅助外部设备)

上穿成卡片，然后把卡片叠放在读卡机上输入计算机，以便运行程序。因此，操作员的工作比较简单，同打字员类似，不需要懂计算机，也不需要懂程序设计。但弗洛伊德毕竟是一个受过高等教育的人，又是一个有心人，干了一段操作员，很快对计算机产生了兴趣，决心弄懂它，掌握它，于是他借了有关书籍资料在值班空闲时间刻苦学习钻研，有问题就虚心向程序员请教。白天不值班，他又回母校去听讲有关课程。这样，他不但在 1958 年又获得了理科学士学位，而且逐渐从计算机的门外汉变成计算机的行家里手。1956 年他离开西屋电气公司，到芝加哥的装甲研究基金会 (Armour Research Foundation)，开始还是当操作员，后来就当了程序员。1962 年他被马萨诸塞州的 Computer Associates 公司聘为分析员。1965 年他应聘成为卡内基-梅隆大学的副教授，3 年后转入斯坦福大学，1970 年被聘任为教授。之所以能这样快地步步高升，关键就在于弗洛伊德通过勤奋学习和深入研究，在计算机科学的诸多领域：算法，程序设计语言的逻辑和语义，自动程序综合，自动程序验证，编译器的理论和实现等方面都作出创造性的贡献。其中包括：1962 年，弗洛伊德



完成了 ALGOL 60 编译器的开发,成功投入使用,这是世界上最早的 ALGOL 60 编译器之一,而且弗洛伊德在这个编译器的开发中率先融入了优化的思想,使编译所生成的目标代码占用空间少,运行时间短。弗洛伊德优化编译的思想对编译器技术的发展产生了深刻的影响。随后,他又对语法分析进行了系统研究,大家现在熟知的优先文法 (precedence grammar), 限界上下文文法 (bounded context grammar) 等都是弗洛伊德在这个时期首先提出来的。优先文法解决了自底向上的语法分析中的首要任务:如何找到“句柄”,也就是当前需要进行归约的符号串。弗洛伊德通过对不同的符号定义不同的优先级,解决了这个问题。限界上下文文法则通过对上下文无关文法 G 中的两个推导:

$$\begin{aligned} S &\overset{*}{\Rightarrow} \beta A r \Rightarrow \beta \alpha \gamma \\ S &\overset{+}{\Rightarrow} \delta \alpha \omega \end{aligned}$$

进行比较以确定 α 是否是 $\delta \alpha \omega$ 的句柄,以及产生式 $A \rightarrow \alpha$ 是否是唯一可进行归约的产生式。弗洛伊德经过研究,给出其充分必要条件为: β 和 δ 的最后 m 个符号相同, γ 和 ω 的最初 n 个终结符相同。这样一个上下文无关文法 G 就称为 (m, n) 限界上下文文法。

在算法方面,弗洛伊德和威廉姆斯 (J. Williams) 在 1964 年共同发明了著名的堆排序算法 HEAPSORT,这是与英国学者霍尔 (C. A. R. Hoare, 1980 年图灵奖获得者) 发明的 QUICKSORT 齐名的高效排序算法之一。此外还有直接以弗洛伊德命名的求最短路的算法,这是弗洛伊德利用动态规划 (dynamic programming) 的原理设计的一个高效算法。

在程序设计方面,计算机科学家非常关心的一个重要问题是如何表达和描述程序的逻辑,如何验证程序的正确性。1967 年,在美国数学会 AMS 举行的应用数学讨论会上,弗洛伊德发表了那篇引起轰动并产生了深远影响的论文,即“如何确定程序的意义” (Assigning Meanings to Programs)。这篇论文在程序逻辑研究的历史上,是继麦卡锡 (J. McCarthy, 1971 年图灵奖获得者) 在 1963 年提出用递归函数作为程序的模型这一方法以后最重大的一个进展。麦卡锡倡导的方法对于一般程序,包括大型软件确实是行之有效的,但它有一个不足,即对于许多以命令方式编写的软件,其中包括赋值语句,条件语句,用 While 实现循环的语句……对这样的程序用递归定义的函数去证明其正确性就很不方便了。正是为了解决这个问题,弗洛伊德在上述论文中提出了一种基于流程图的表达程序逻辑的方法。这个方法的主要特点就是在流程图的每一弧线上放置一个“标记” (tag),也就是一个逻辑断言,并且保证只要当控制经过这个弧线时该断言一定成立。弗洛伊德的主要贡献在于解决



了基于这种标记的形式系统的细节,证明了这种系统的完备性,解决了如何证明程序终结的问题。弗洛伊德还引入了验证条件的概念,包括流程图的一个组成部分(方框、圆框等)及其人口和出口处的标记。为了证明带标记的流程图的正确性,只要证明其中每一组成部分的验证条件成立就行了。弗洛伊德提出的方法被叫做“归纳断言法”(inductive assertion method),或前后断言法(pre - and post - assertion method)。在框图每个断点 i 上所加的逻辑断言即标记就叫 i 点的归纳断言,说明程序执行经过此点时在各输入变量 x 和各程序变量 y 之间应存在的关系,以谓词 $P_i(x, y)$ 的形式表示。若程序从断点 i 经过路段 α 到下一断点 j 的验证条件以 $R_\alpha(x, y)$ 表示, y 的值在 α 上的变化以 $h_\alpha(x, y)$ 表示,则只要能证明下式恒真:

$$(\forall x)(\forall y)[P_i(x, y) \wedge R_\alpha(x, y) \supset P_j(x, h_\alpha(x, y))]$$

程序从 i 到 j 的部分正确性也就证明了。

虽然用归纳断言法不能证明程序的完全正确性,因为它必须以程序能够终结为前提,但由于弗洛伊德在论文中同时也考虑了如何证明程序终结的问题,因此弗洛伊德的归纳断言法也就有了普遍的意义。

弗洛伊德在同年发表于《ACM 学报》(Journal of ACM) 10 月号上的另一篇论文中,还第一次把“不确定性”概念引入程序。所谓“不确定性程序”(nondeterministic program)就是根据操作规则有多种操作可供选择,而只选其中之一搜索下去的程序。这对人工智能问题的研究具有十分重要的意义。

此外,弗洛伊德还和伊万斯(R. O. Evans, 因设计世界上第一个类比推理程序 Analogy 而闻名于世的学者。Analogy 是可以判定几何图形是否类似的人工智能程序)一起设计了一种称为产生式语言的特殊的程序设计语言 FPL (Floyd - Evans Production Language),用来编写计算机语言的语法分析程序。之所以称它为产生式语言,是因为用它编写的程序由一系列产生式(或称归约式)组成。实际上,用 FPL 编好语法分析程序以后,如果再插入语义子程序,就可以构成一个完整的编译器。用 FPL 语言编写的程序简称 FP 程序,由以下 5 个部分按自左至右顺序组成:

1. 标号(可有可无);
2. 栈顶符号串;
3. 前看符号串(或称窗口符号串);
4. 归约符号;
5. 语义动作。

执行一个 FP 程序的方法是:依次检视各 FP 的第三部分。若某 FP 的第三部分和输入的前看符号串一致,则进一步检视此 FP 的第四部分,若非空,表示要进行归约,此时把它的第二部分和当前实际的栈顶符号串相比。如果



能匹配上,则实行归约,即删去实际的栈顶符号串,用第四部分代替之,然后执行第五部分的动作。若此 FP 的第四部分为空,表示当前无归约可做,直接执行第五部分的动作即可。

弗洛伊德是 1978 年 12 月 4 日在华盛顿举行的 ACM 年会上接受图灵奖的。他发表了题为“程序设计的风范”(The Paradigms Of Programming)的演说。演说全文刊于 Communications of ACM,1979 年 8 月,455 ~ 460 页,也可见《前 20 年的 ACM 图灵奖演说集》(ACM TuringAward Lectures——The First 20 Years:1966—1985,ACM Pr.),131 ~ 142 页。弗洛伊德在演说中对结构化程序设计,递归协同例程(recursive coroutine),动态程序设计,基于规则的系统,状态变换机制(state - transition mechanism)等各种不同程序设计风范进行了比较,并介绍了自己在研究工作中如何根据具体情况应用不同风范的例子,很给人以启示。时间虽然已过去 30 多年,他的例子也许有些过时,但他的观点至今仍然是有效的。