

吴轶华
沈祖梁 译



电脑、信息 与人类



电 脑、信 息 与 人 类

戈登·帕斯克 著
苏珊·卡 冉

吴轶华 沈祖梁 译

中国展望出版社

责任编辑 朱光祖
封面装帧 王沙城

电 脑、信 息 与 人 类

戈登·帕斯克 著
苏珊·卡 冉
吴轶华 沈祖梁 译

中国展望出版社出版
(北京西城区太平桥大街4号)

常熟周行联营印刷厂印刷

北京市新华书店发行

开本787×1092毫米 1/32 印张7.25

140千字 1987年12月北京第1版

1987年12月第1次印刷 1—1,500册

统一书号: 17271·031 定价: 1.53元

目 录

导 论	[1]
第 1 章 计算机跨入时代	[5]
第 2 章 吓人的计算机	[26]
第 3 章 计算机、意识和冲突	[48]
第 4 章 计算机群体	[65]
第 5 章 语言 and 知识	[75]
第 6 章 数据结构和知识结构	[90]
第 7 章 使用中的微处理器	[108]
第 8 章 离经叛道的机器	[129]
第 9 章 新型的教师——插件	[144]
第 10 章 信息环境	[166]
第 11 章 可能的未来世界	[188]
第 12 章 机器人和人的变革	[212]

导 论

这不是一本关于计算机的书，而是一本关于人类与计算机发展相互关系的书。过去，我们人类制成了计算机；现在，计算机正在形成未来的人类和人类的环境。

自第二次世界大战后发明了可程序的电子机器以来，计算机经历了很大的演变。晶体管的发明以及紧接着的微电子技术的发展，使计算机和被称为微处理器的计算设备体积缩小，价格便宜，运算可靠，功能齐全。计算、通信和控制设备的急剧增长将形成所谓的“信息环境”。不管怎样，我们相信这种变化不只是体现在量的方面，在其后面有着关于人类与机器之间关系的更为深刻的质变，我们还相信这些变化将导致称为“微型人”的新种族的诞生。

当人们制造出新工具：铁铲、推土机或计算机后，由于使用了这些新工具，人们往往受到通常无法预料的可能事情的影响。由于与新工具打交道，他们的思想也在演变，结果，这些工具成为思想变化的见证。

直到最近，人们才逐步摆脱仅靠一些相当简单的工具

来设计和制造计算机这种局面。人们开始采用更复杂的机器——如计算机本身——来辅助设计计算机。今天在不要或几乎不要人的帮助之下,计算机能设计计算机;微处理机控制的机器能制造计算机。计算机不再仅仅是工具,它们本身已经独立形成一个新的种族,虽然依赖于我们,但不从属于我们。

结果,现在果真出现了一个信息处理和传输的工具,而且,不管你是否喜欢它,它正在逐步地演变和进化。在人类和计算机两个群体间将存在什么样的关系呢?是对抗还是合作?我们希望并相信两个种族将合作,通过合作促进对方的不断进化。出于这种相互合作,将会出现一种转化——人脑和计算机本质的转变,导致一对相互依赖的新种族的产生。

我们可以进一步探索这个问题,不是考虑人和计算机两个种族,而是一个种族——心理脑(指心理上而非生理上的实实在在的大脑——译者注),它可驻存于某人或许多人中,也可驻存于机器或目前还无法想象的构造之中。在这本书中,我们将剖析一下已知的人脑和精心编程的计算机“思维”间的异同点。我们相信,随着生物体和非生物体间的界限的逐步模糊,人类思维与机器思维间的本质区别将站不住脚。我们面临着一场关于心理脑的革命性扩展,即心理脑可以是属于个人的(现在就是如此,每个人有其自己的心理),也可属于社会的,甚至属于用非人脑材料制成的某种构造。

不过,在考虑扩展之前,我们必须考虑目前机器的限

制。目前,计算机在两个方面受到了限制。其一是结构,即计算机处理数和符号的方式,其二是并非所有的实体都能根据逻辑来处理数和符号,建立起它的模型。现在的计算操作建立在顺序的思想基础上(概念一个接着一个)。这种模型可理解为一个时间序列或因果序列,而这两种序列都可用一根拉紧的绳来表示其线性本质,用绳上的结表示因果链或时间的瞬间。

这种简单的绳—结模型反映了至少在西方已经流行很久的一种关于人脑工作方式的观点。而且的确,在伽利略出生到爱因斯坦相对论的诞生,其间几个世纪内,有序的、串行的和推导的思维方法在给宇宙提供一个有前因后果的逻辑解释方面取得了最大的成功。可是只要稍微考虑一下,我们就会发现绳—结模型的缺点。当我们说或写某件事时,我们好象立刻注意到这件事。这一点对我们集中注意力很有好处。可事实上另一方面,潜意识也可能决定我们将把注意力集中在其它什么地方。这样,任何非逻辑推理式的思维活动——创造、想象和发明,需要在大脑中同时存在着几条思路,并且以一种非顺序的方式进行处理。

我们认为,建立在一个顺序的、逻辑的和推导的理想化西方思维模型基础上的计算,已逐步显露出限制计算机应用的缺点。这种模型的处理,从根本上说是以一种逻辑的方式编制程序,并假设计算机不可能创造、发明和类比。

我们觉得这种限制是毫无必要的。我们可以用种种方法来设计计算机或计算机网络,为它们编制程序,从而可以克服这种限制,在本书中我们将会看到其中的一些方法。我

们相信这些方法将在未来的信息环境中大量采用，使得计算机以一种非逻辑顺序的方式来思考，并且控制用串行模型不能模仿的环境。正是在这一点上，使计算机与人类思维过程中的同时性与间断性之间建立起良好的接口成为可能。

计算机能象人一样思考、交谈和动作的时代还很遥远，甚至有人还认定它永远不会到来。我们不低估困难，但我们相信人工智能拥有无限的潜力，单个的或群体的机器脑将在越来越多的领域内超过人脑，其潜在的力量既使人激动又让人害怕。我们承认危险，但依靠智慧和冒险精神，也许还有勇气。我们将会看到一个具有人类传统的、美好得无法想象的丰富多彩的世界。

这是一个微型人的世界，今天它正在逐步形成。并且，我们可以发现，人们在接受从心理学到建筑学、从教育学到认识论等各种知识时，他们的思想已经开始反映出这个新的时代。本书的目的之一是综合他们的这些思想，至少描绘出一种轮廓，并且指出，这些随着计算的发展而形成的思想又是怎样由一些相互关联的想法组成的。本书的另一个目的是帮助你——读者——熟悉这个新世界，因为正在形成的是你的世界，而你又必将投身进去，使这一新的世界得以形成、出现。

第1章 计算机跨入时代

对很多计算机程序员和操作员来说，计算机是一台庞大的、强有力的、有时甚至是令人恼怒的计算机器，但其实际作用与潜在能力远不止于此。

现代计算机既简单又复杂。说它简单是因为它真正做的只是加、减和比较；说它复杂是因为所有要它干的工作都必须分解为这些简单的操作。

计算始于计数，计数始于认识“1”不同于“2”或“许多”。在“两头牛”和“两只羊”之间有着重要的相似之处，而在“两头牛”和“三头牛”之间却有着重要的差别。一旦数从量中分离出来，摆弄数字便成为可能，因此也就有了计算。数能用许多方式进行处理，而物体却不能。一头牛的平方根是多少？问题是无意义的，但一个数的平方根在数学系统中却是有意义的。

从发明算盘到十九世纪发明穿孔卡片机这段非凡的岁月里，人类将其大量才智贡献于自动计算，目的是使数的处理更精确、更灵便。事实上，十九世纪中叶之前，计算一直局

限于数的计算。计算机区别于计算器的关键是它能处理包括数在内的各种各样的符号，如果这些符号能够遵循一个约定的系统的话。计算机可以用无数种方式进行这种处理。如果把人看成一个超级的符号处理器，那人也是“计算机”。

处理信号和符号的基本目的是什么呢？是通信、控制或二者兼之。计算、通信和控制间正在发展着的关系，比任何别的东西更能解释计算机怎样发展到今天，以及它将可能把我们引向何方。由于这种计算，迫使我们重新审查通信、控制、记忆、空间和时间等概念，因此它正在改变着二十世纪末的世界，正如数的发现曾改变过远祖的世界一样。

算术的机械化

让我们花点时间比较一下一台现代电子计算机和一台简单的、象算盘一样的计算设备。计算机用户并不主动监视计算机操作的每一步；控制交由一段程序或一组自动执行的指令来完成。即使是机器的设计者，他也不可能确切了解在给定时刻，机器究竟干了什么？现代计算机是一个“黑箱”。相比较而言，算盘的使用者在沿着小棍滑动算盘子时，就能确切了解所发生的一切。一部计算机的历史其实就是控制的能力逐渐增强，机器的奥秘不断增多的历史。每提高一次自动化程度就需要相应地加强人一机的交流。我们必须想法告诉机器干什么，同样，机器完成了难懂的“芭蕾”后，必须将结果告诉我们。机器和我们的交接必须在一个实用的水准上——我们输入某些东西，得到某些输出；并还必须在一个智能的水准上——机器必须弄清我们将什么输

入，我们必须理解它们将什么输出。

最简单地讲，就是在一组称为程序指令的引导下，现代计算机通过执行加、减和比较操作来处理符号。整个复杂的数学处理的世界能分解成加法和减法（实际上这些运算还可进一步简化为一个操作，不过这是另外一件事）。乘就是重复的加，除则是重复的减，即使最复杂的运算也可用各种方法加以处理。例如，对数有助于将运算分解成加法和减法。计算机在执行指令时，比较将是很重要的。它必须能检查两个符号或符号串是否相同，由此决定应执行哪条指令。

我们不妨简略地回顾一下电子计算机的发展概史。算盘，它发明于十三世纪的中国，至今还应用于一些远东银行。过去欧洲也用过，但它的使用，还没有到使一些数学家，诸如布雷斯·帕斯卡和葛特弗瑞德·威尔赫姆·莱布尼茨等泄气的程度，他们不肯放弃发明复杂的齿轮和轴的组合装置，以装备一架简单的运算器。算盘由熟练的操作员来使用时，其效率相当高，在与桌式电子计算器的比较实验中，经过一系列算术运算，算盘显示出与计算机同样的速度和精度。

算盘的设计有许多不同的方法，但其工作原理都是相同的。一只现代的日本算盘，将862加到164298上，需要八个动作。在帕斯卡1642年发明的计算机上，相同的求和只要三步：输入第一个数、输入第二个数、启动加法机械。但一次只能处理两个数，而且结果并不总是正确的。后来由莱布尼茨发明的计算机又加进了乘除运算的装置。

差 分 机

大约 150 年后，一位英国数学家名叫查尔斯·巴贝奇（他家在伦敦，并在伦敦有个工场），设计了一台理论上比实践上更有意义的机器：差分机。其目的是制作包括对数表在内的种种数学用表，并将其精度提高到以前不可能达到的程度。

巴贝奇的装置很大程度上应归功于早期计算机器的设计，但寄存器（机器中可以输入并存储数的单元）却是他的独特贡献之一。帕斯卡的机器仅有一个给轮子定位的装置，同一时刻只能保持两个数。巴贝奇的机器首先可以将所有的数输入到寄存器，这些数指示后来接着计算所需要的数。输入完毕后再进行处理。其输出概念也更复杂，机器将结果刻在铜版上，以便能直接印刷。每算一次，机器提供给用户一个结果记录，并且避免可能发生的印刷错误。

不幸得很，当时的机械技术配不上他的才智。尽管他在差分机上干了整整二十年，却无法找到一个能构造它的工程师，他将大部分的精力花在差分机所产生的实际问题上去。虽然如此，差分机的理论对后来计算机的发展还是很有影响的。在他之后一对瑞典父子皮尔·约格和爱德华·斯彻兹，设计了一台稍微逊色的机器。在本世纪，有几台机械传动的差分机还在商业界投入使用。

分 析 机

1832年，当巴贝奇还在与差分机的各种技术问题苦战

的时候，他又将注意力转移到一个更宏伟的工程：分析机。差分机只用于产生数学用表，分析机则能间接地（虚拟地）完成任何运算。其存贮的数据能给任何次序使用（差分机只能按固定的次序使用存贮的数据），而且在计算过程中，其中间结果也可存贮起来，另外它还能接受各种各样的指令（差分机的指令固定于机器中）。用现代术语来说就是分析机有一个读写存贮器，数据既可读出也可写入。它还可编程序，即能输入各种指令，例如指明怎样处理有关的数据。

巴贝奇设计的这架分析机，还采用了两个不同的子部件：一个算术/逻辑部件和一个控制部件。前者根据算术或逻辑规则来处理数据，后者按正确的次序解释指令组，并启动算术/逻辑部件具体执行某一条指令。

在十九世纪初，现代计算机的轮廓已见端倪，分析机及其现代后裔都有下列特点：

输入指令和数据的方法

存贮信息的存贮器

按正确顺序执行指令的控制部件

执行必要操作的算术/逻辑部件

输出结果的方法

从一个工程师的观点来看，分析机比差分机更难于构造。即使是它的一个简单变体也要五万多个活动部件。巴贝奇坚持存贮一千个数，每个数必须达到一个荒谬的精度：绝不能少于小数点后面五十位数。律师兼数学家的洛德·莫尔敦动人地描绘了巴贝奇，说他的新思想太多，他的雄心太大，以至于当时的技术根本无法实现。“我一生中令人悲

哀的记忆是拜访一位被赞美的数学家和发明家——巴贝奇先生。他年纪已很大了，但精力总是那么充沛。他带我参观他的工场。在第一间房屋里我看见原来的计算机器的零部件。这台机器虽然曾经用过，但很多年以来就一直处于未完成的状态。我向他询问机器现在的情况。‘我还没有完成它，因为在工作过程中，分析机的想法突然冒出来了。分析机能干差分机所能干的一切，而且还远远不止这些。的确，这些要简单得多，以至于完成差分机比完整地构造另外一台要花更多的精力，所以我将注意力转移到分析机上’。我问他是否能看看，他说：‘我一直没完成，因为我想用不同的和更有效的方法来完成这件工作，这样，继续走老路便毫无意义’。然后，我们走进了第二个房间，那里躺着一些零散的机件，没有任何迹象表明有一台能运行的机器。我很小心地接近这个问题，并得到了令人沮丧的回答。‘它还没构造好，但我正在努力将其组装到一块’。我带着沉重的心情离开了老人。”

很不幸，巴贝奇对他的分析机和该机的特色没有留下清晰明白的描述。然而，他的一些追随者，包括洛德·拜伦的女儿阿达·娜美丽丝女士，弥补了这个疏忽。

告诉计算机做什么

巴贝奇使用穿孔卡片存贮指令。卡片是从提花机上的卡片发展而来的，每条指令占用一张卡片，卡片以正确次序输入到分析机中。每张卡片顺序地触发下一条指令。还有一些分散的卡片，指明哪个数据存贮器中存有将被处理

的数。整堆卡片可以重复使用，而且不占用机器的数据存贮器的存贮空间，还能在进行不同计算时换上另外一堆卡片。整个十九世纪，穿孔卡片大受欢迎，适用于许多机器，包括自动售货机和自动钢琴，甚至在今天，穿孔卡片和穿孔织带机仍相当广泛地使用着。

感谢娜芙丽丝女士，她使我们知道巴贝奇在其穿孔卡片系统中试图实现的一些复杂技术。在提花机或自动钢琴的滚筒中，指令只需按其输入的顺序依次执行。然而，数学家提出的特殊问题却需要我们去完成更为复杂的排序。

娜芙丽丝女士在 1842 年勾画了现代计算机程序设计的两个基本特点：循环和分枝。循环是指一组指令仅需输入到机器内一次，机器可重复执行这组指令若干次。许多数学计算本质上重复极多，所以有必要将相同指令多次输入，或者寻找某些提供重复指令系列的方法，循环正是此种方法之一，触发机械能够识别指令系列的最末一条，然后又自动回到第一条指令。

分枝是指某条指令操作的结果可能影响其余指令的执行次序。让我们想象一个程序。如寻找 4、7 和 13 的最小公倍数（见图 1-1）。指令 2、3、4 和 5 构成一个循环，它们被重复执行一次又一次，直到指令 3 中问题的回答是“是”。指令 3 导致了程序的分枝：如果答案是“是”，接着执行指令 6；如果答案是“否”，接着执行指令 4。分枝和循环过程无需人的干预。

程序是计算机的一部分还是操作员工作的一部分？今天的计算机用户倾向于将程序看作是计算机的一部分，这一点并不是因为所有现代计算机内都有许多指令集（包括

语言编译器)。为了区分机械和指令,前者称为硬件,后者称为软件。

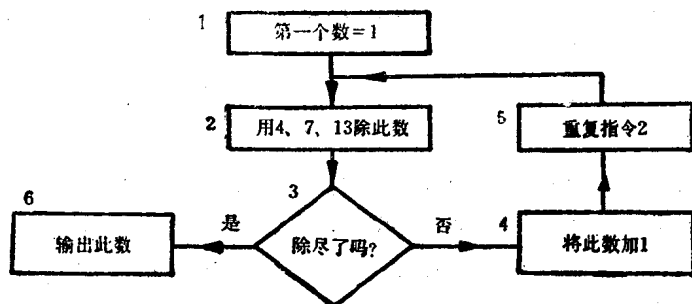


图 1-1

程序(软件)导致计算机按预定次序执行程序员的指令。程序员可能不知道计算机执行他的指令的次序。事实上,在一个更复杂的程序中,对一个程序员来说,碰巧设计了一个程序,以至于计算机不管输入什么数据都无法执行其中的某些指令,这并不少见。虽然如此,不管程序员是否能预料程序怎样运行,计算机只能按程序员提供的规则来运行。如果没有机器故障,程序和数据均相同,那末计算机每次运行结果也将相同。

根据执行前面顺序指令的结果,再以不同的顺序来运行后面指令,这一点是现代计算机的基本特点之一。也许,这比任何一点更能将计算机与计算器区分开来。可编程序计算器能被指定去执行一连串的操作,但如果这一连串的指令可以变化,最好将它们称为计算机。

算法和程序设计语言

自巴贝奇以来,计算机程序设计技术已有很大发展。两个主要的进展是:形成了算法理论和程序设计语言的概念。算法理论主要由苏联数学家 A.A. 马尔柯夫在五十年代总结出来。

算法是一组有程序的步骤,它能提出如何解题的概要方法。为了计算,算法作为一个框架,用于构造一种程序,这种程序就会告诉计算机如何去做,图 1-1 的简单流程图本身就是一个算法。

程序设计语言本质上是一组约定,使得人的指令能转换为计算机可处理的编码。自从巴贝奇以来,计算器和计算机一直只处理简单的两种状态或二进制码的信息。穿孔卡片便是二进制码的一个例子:它以有孔或无孔来表示所存贮的信息。现代电子计算机以有脉冲和无脉冲来表示信息,在以后的电子时代中,计算器的信息将用电磁转换来贮存,但也只能贮存两种状态。我们一般用 0 和 1 来表示这两种状态。数、字母和别的符号,不管它们代表数据或程序,在计算机能处理之前,都得转换成二进制码,并且有一个国际公认的标准来完成转换。例如,从 1 到 10 这几个数可变成二进制 001、010、011、100、101、110、111、1000、1001、1010。

很明显,转换为一长串 1 和 0 的指令系列很不方便,尤其是书写须由漫无止境的 0 和 1 的符号串组成程序,既困难又乏味。程序设计语言致力于解决这个问题,用更易对付和更具智能的符号来代替二进制码。假设计算机