

# 计算机引论

于占涛 刘长林 主编  
赵秀云 秦永录

辽宁科学技术出版社



## 内 容 提 要

本书比较全面地介绍了计算机科学、算法的分类、计算机历史、现状和发展方向；通俗易懂地介绍了计算机结构、计数系统、程序设计语言和软件的有关概念；完整、详细地讲述了Pascal语言及其在非数值运算、数值运算、人工智能等方面的应用，并给出了部分典型算法；同时还概括介绍了软件工程的概  
念以及计算机网络、分布式系统和第五代计算机的有关知识，讨论了计算机在使用方面的伦理学和计算机对社会的潜在冲击。各章附有一定数量的练习和习题。

本书主要根据美国特拉华(州)大学教材“Principle of Computer Science”，并结合我国国情编译而成。本书注重了系统性和实用性。既可作为高等院校“计算机引论”和“Pascal语言”课程的教材，也可作为从事计算机应用的工程技术人员、科学研究人员参考书。

# 前 言

目前,我国大部分院校的计算机应用专业和计算机软件专业都设有“计算引论”或“计算机引论”课,该课程目的是使同学入学后立即接触计算机,较全面地了解计算机科学,了解计算机的历史、现状及其未来,了解计算机的各个应用领域及其对社会的影响,并通过本课程的教学对学生进行高级语言编程训练,使之熟练掌握一种高级语言,培养一种良好的程序设计风格,为后继课学习奠定基础,但是能达此目的教材较少。

我们将李华天教授推荐的美国特拉华(州)大学教材“Principle of Computer Science”作为本课程教材,试用两年,效果较好。该书于1986年由美国计算机科学出版公司出版。它涉及的内容较多、范围较广,有一定的深度。从内容的组织和重点选择上具有自己的风格和特点,既注重了先进性、理论性和系统性,又列举了大量的应用实例,突出了实用性。我们认为它是“计算机引论”和“Pascal语言”课的极好教材,也是从事计算机应用的广大工程技术人员、科研人员不可多得的一本参考书。

本书是在齐志儒老师主持编译的内部教材的基础上,根据两年的教学实践,进行了有选择的翻译、编译和适当增补一些内容编辑而成的,使之更适合于我国大学教学用书的需要。

全书共分十八章。第1~2章比较全面地介绍了计算机科学、计算机科学子域、算法的种类、适用范围和优缺点。第3~6章讲述了计数系统,介绍了计算机结构、程序设计语言和系统的概念。第7~11章详细讲述了目前国际上流行的、结构最好的程序设计语言—Pascal语言。第12~15章介绍了计算机在非数值运算、数值运算、人工智能等方面的应用,并给出了部分典型算法;强调了不同问题的求解技术。第16章介绍了软件工程的有关概念。第17~18章介绍了计算机的发展方向、计算机网络、分布式系统和第五代计算机的有关知识,讨论了计算机在使用方面的伦理学和计算机对社会的潜在冲击。

本书第1、2章由齐志儒编译,第3、5、6章由赵秀云译,第4章由王俊圣编译,第7、8章由刘长林编译,第9、14章由徐雪初编译,第10章由王冬生编译,第11章由刘本忠编译,第12章由高福祥编译,第13、18章由秦水录编译,第15章由张敬亭编译,第16章由于占涛编译,第17章由王向荣编译。

朱家铨同志组织了本书的编译工作。

全书由于占涛、刘长林、赵秀云、秦水录(以姓氏笔划为序)同志主编。

由齐志儒、何文兴同志主审。

本书在组织出版和出版的过程中得到辽宁科学技术出版社和辽宁省计算机学会教育与培训委员会的大力支持和帮助,我们表示衷心的感谢。

由于时间仓促和编译者水平所限,书中错误和不妥之处在所难免,殷切希望广大读者批评指正。

编译者

1990年5月

# 目 录

## 前言

第一章 计算机科学：发展中的科学.....	1
1.1 作为一门学科的计算机科学.....	1
1.2 什么是计算机.....	2
1.3 计算机的发展历史.....	4
1.4 本章练习.....	5
第二章 算法和问题求解.....	7
2.1 算法的性质.....	7
2.2 计算的算法.....	8
2.3 算法的分类.....	13
2.4 算法的分析.....	20
2.5 结构化程序设计.....	21
2.6 本章小结.....	22
2.7 本章练习.....	22
第三章 计算机中数据的表示方式.....	24
3.1 按位计数系统.....	24
3.2 位，字节和字.....	31
3.3 本章小结.....	38
3.4 本章练习.....	38
第四章 计算机结构.....	40
4.1 数字计算机的组成.....	40
4.2 假想的机器语言.....	43
4.3 符号程序设计.....	47
4.4 寻址方式（任选）.....	49
4.5 本章练习.....	51
第五章 程序设计语言概念.....	52
5.1 编程方法.....	54
5.2 符号语言学（形式语言学）.....	55
5.3 BNF（巴科斯—诺尔）范式.....	56
5.4 二义性.....	60
5.5 本章小结.....	64
5.6 本章练习.....	64
第六章 计算机系统软件.....	66
6.1 程序设计语言.....	66
6.2 翻译程序.....	66
6.3 操作系统.....	70

6.4	本章小结 .....	74
6.5	本章练习 .....	74
<b>第七章</b>	<b>基本数据结构 .....</b>	<b>76</b>
7.1	语言的基本元素 .....	76
7.2	数据类型 .....	78
7.3	变量说明 .....	80
7.4	常量说明 .....	80
7.5	赋值语句 .....	81
7.6	Pascal 程序 .....	81
7.7	表达式 .....	82
7.8	离散类型的序数特性 .....	88
7.9	输入语句 .....	89
7.10	输出语句 .....	91
7.11	Pascal 实例 .....	92
7.12	本章小结 .....	92
7.13	本章练习 .....	92
<b>第八章</b>	<b>控制结构 .....</b>	<b>95</b>
8.1	复合语句 .....	95
8.2	重复语句 .....	95
8.3	选择语句 .....	101
8.4	著名的 GOTO 语句 .....	106
8.5	本章小结 .....	108
8.6	本章练习 .....	108
<b>第九章</b>	<b>复合的数据结构 .....</b>	<b>110</b>
9.1	数组 .....	110
9.2	多维数组 .....	114
9.3	串 .....	118
9.4	记录 .....	120
9.5	本章小结 .....	124
9.6	本章练习 .....	124
<b>第十章</b>	<b>过程和函数 .....</b>	<b>127</b>
10.1	过程 .....	127
10.2	函数 .....	132
10.3	内存分配 .....	134
10.4	模块的通讯与调用 .....	138
10.5	过程参数与函数参数 .....	141
10.6	递归 .....	143
10.7	本章练习 .....	148
<b>第十一章</b>	<b>动态数据结构 .....</b>	<b>149</b>
11.1	动态数据结构的特点 .....	149

11.2	列表	155
11.3	堆栈和队列	157
11.4	树	159
11.5	本章小结	161
11.6	本章练习	161
<b>第十二章</b>	<b>非数值算法</b>	<b>163</b>
12.1	检索	163
12.2	排序技术	165
12.3	快速排序：一种有效的交换排序算法	170
12.4	归并	174
12.5	本章小结	177
12.6	本章练习	177
<b>第十三章</b>	<b>信息系统</b>	<b>178</b>
13.1	文件概述	178
13.2	Pascal 文件	178
13.3	顺序文件、直接存取文件和索引顺序文件	182
13.4	归并外部文件	187
13.5	数据库管理系统	188
13.6	本章小结	190
13.7	本章练习	190
<b>第十四章</b>	<b>科学计算</b>	<b>191</b>
14.1	牛顿——辛普生法	191
14.2	高斯消去法	192
14.3	随机数	195
14.4	蒙特卡洛法	195
14.5	计算中的误差	198
14.6	本章小结	199
14.7	本章练习	199
<b>第十五章</b>	<b>人工智能</b>	<b>201</b>
15.1	智能机器：有很大争议的问题	201
15.2	面向问题的技术	201
15.3	游戏程序	207
15.4	专家系统	208
15.5	自然语言处理	211
15.6	重新回到有争议的大问题	212
15.7	本章练习	213
<b>第十六章</b>	<b>软件工程</b>	<b>214</b>
16.1	什么是软件工程	214
16.2	软件开发计划的制定	218
16.3	需求分析	218

16.4	软件设计	218
16.5	说明举例	219
16.6	实现	223
16.7	软件测试	224
16.8	文档	227
16.9	维护	227
16.10	本章小结	228
16.11	本章练习	228
<b>第十七章</b>	<b>发展方向</b>	<b>229</b>
17.1	计算机网络	229
17.2	程序包	232
17.3	第五代计算机	237
17.4	本章小结	238
<b>第十八章</b>	<b>计算机对社会的冲击</b>	<b>239</b>
18.1	日常生活中的计算机	239
18.2	有关保守私人秘密的问题	240
18.3	计算机病毒	241
18.4	用计算机进行犯罪活动	242
18.5	用计算机查找罪犯	243
18.6	计算机与军事	244
18.7	计算机与就业	244
18.8	本章练习	245

# 第一章 计算机科学：发展中的科学

计算机科学是一门极其重要而又尚未完全定义的学科。其重要性与它提出的问题及开发它的专业人员的心理习惯有关。它未完全定义是因为它年轻繁杂和正在不断发展中。在本章我们将试图描述这个学科的范围并概观它的子域，然后，追溯它的历史。

## 1.1 作为一门学科的计算机科学

回答“什么是计算机科学？”并非不可能，但是很困难。部分原因是该领域相对来说是新的且仍在发展，然而，更多的困难似乎来自于学科的范围，它所包括的学科的数量和种类（如：数学，机械和电子工程，语言学），及它的理论与应用的交织程度。

面向工程的观点可能定义计算机科学为计算机的研究。这样的定义强调了物理工具和它们的能力，但忽视了这个工具的用途和使用它们的技术。

另一个极端，可能用程序设计这个词定义计算机科学。程序设计是对计算机可解问题及解法与表达技术的研究，这种观点强调了使用计算机的机会和技术而对计算机本身漠不关心。

这些定义都是不可接受的，因为二者定义的范围太窄。合并二者并且加入一些新的内容也不是解决办法。首先，因为计算机仍在发展，我们并不知道它的完整的范围；第二，有许多不同的方面是属于计算机科学家，还是属于电子工程师，语言学家等还不清楚。例如，物理计算机的设计，有理由认为它是电子工程师的问题，计算机设计的理论明显地应由电子工程师和数学家完成，关于程序设计语言的设计中许多问题，最好由语言学家或精神语言学家回答。

幸运的是，我们不能定义计算机科学的事实使我们免于学习或研究该学科各个方面。大学有计算机科学系，工业寻找获得计算机科学学位的雇员。这表明，在某种意义上但不确切地说存在一致的意见，由此，我们暂且回避“怎样定义计算机科学？”的问题，而代之以叙述计算机科学家本身感兴趣的主要课题。

广泛地说，计算机科学家试图开发依靠计算机求解问题的理论和应用。强调计算机本身或者计算机的使用方法，强调基础理论或者应用理论的方法。这些观点引出了计算机科学课题下的子域，我们可以将计算机科学划分为四个子域，其中每一个子域都有几个方面，图 1.1 进行了描述。虽然子域可以独立地叙述，但是他们之间还有大量的重叠，设计要使用的系统，开发便于系统设计的理论。

### 1. 形式计算机科学

1936 年通用图灵机的定义标志形式计算机科学的开始。图灵推测他的机器可以计算任何可计算的函数，相反，所有其它的函数都是不可计算的。这个推测通常假定为真，部分原因是没有一个人能推导出反例。因此，通常假设任何可制造的模仿图灵机的机器足以解决任何可解问题。

虽然计算机科学保留了对可计算性问题的兴趣，但其它形式理论研究领域已变得相对重要，其中包括自动机理论，等价算法和程序正确性的证明。

### 2. 计算机系统

计算机系统领域由两个主要分支——硬件技术和软件技术组成。前者涉及利用最少和最



先进的物理元件的计算机系统的设计和逻辑结构。计算机网络，用电缆或卫星连接结构的设计，通常认为是一个工程任务。

软件技术涉及操作系统的结构，程序设计语言的翻译器和基本控制程序。最近几年，这个分支也与“程序设计环境”或“程序开发系统”，一些规则，工具和使用说明，编程，测试，证明和开发管理大型系统软件的约束相联系。

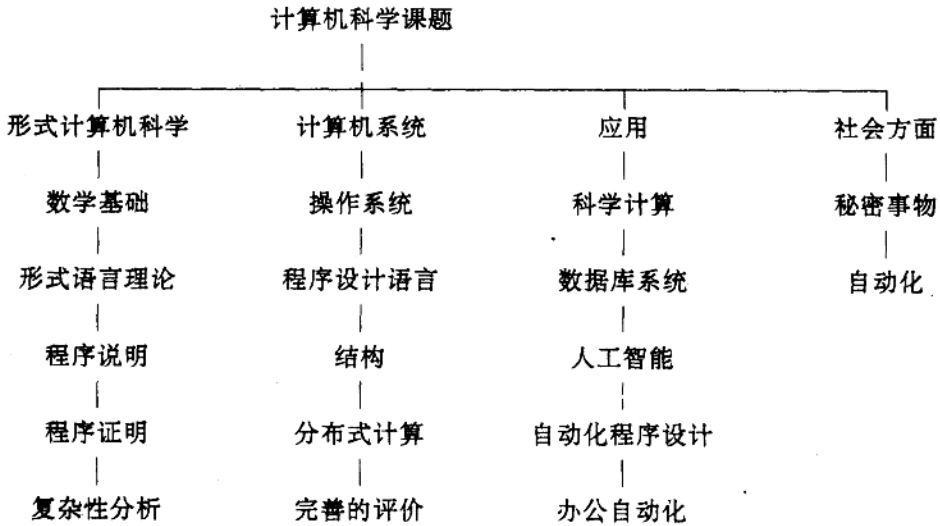


图1.1 计算机科学的主要子域

### 3. 应用计算

科学计算是最古老，最广泛熟悉的应用计算领域，它处理解决各类数字问题的有效过程的开发和这些过程的改进。有关效率，程序执行中的错误控制，解决一般问题的软件包的开发和模拟等问题也划分到这个范畴。这个领域的工作强调高等代数和数字程序的开发，这些程序利用计算机硬件的最新发展。由于非数值应用的迅速发展，带来了技术上的发展，科学计算不再象 50 年代和 60 年代那样支配应用计算。

作为数据库管理系统领域的例子是大学，政府管理系统，银行和企业管理系统的计算机的使用。边界领域包括库存控制，订货处理，个人经营，记账，报告生成，办公自动化，而且这个领域是应用计算的最大领域。

人工智能和机器人复盖了应用的广大范围，包括问题求解，自动化理论的证明，自然语言处理，语言和视觉信息的处理，象用于医学和教育的专家系统和用于工业生产的机器人。这个学科从关于符号处理机械的范围和性质的早期推测发展起来的。其它应用领域（例如，符号计算，分时和图形）是早期人工智能工作的派生。

实时处理和过程控制由某些不断活动的系统的物理事件来刻画，如控制温度。其它例子包括航线预订系统，工业过程控制系统，导弹导航系统。每个系统都要求必须达到满意的速度。这样的系统必须具备一个良好的用户接口，接收所有的查询并在指定的时间内响应。

### 4. 社会方面

自动化的社会影响是令人惊愕的。由于许多工作的自动化，社会必须处理诸如失业，工人退休，空闲时间的增加等问题。含有详细的个人信息的大型数据库，对个人事业产生了威胁，以前，但不太远，1984年 Gorgor Orwell 予想的一个威胁——家庭计算机不允许成为大兄弟的眼睛，监视和控制我们的行动。

## 1.2 什么是计算机？

无论我们是否赞成计算机科学包括计算机的研究，但它包括计算机的使用是无争议的。银行的出纳员可以使用一台计算机检查你的账目，而不必知道比按键更多的知识，较多直接涉及计算机的人至少要一般地了解什么是计算机及其有何功能。

计算机系统有三个主要部件：

1. 主存储器：它存储系统解决问题需要的信息，该信息包括处理的数据和执行的指令。
2. 处理器：它完成对存储器内容的计算并且具有协调所有部件的功能，以正确地执行预定的操作序列。
3. 输入输出设备：用于计算机与外界信息的交换。

这些部件及其相互连接方法如图 1.2。

我们可以把执行程序的计算机看作拥有一个计算器、一些数据和处理数据的执行程序的机械人员。计算机的存储器保存数据和程序，通过输入设备将它们送入。处理器包括计算器，由它选择下一步要执行的过程的步骤和如何解释所选步骤的信息。一旦获得答案，计算机就通过输出设备把它传送到计算机的外部。

重要的是注意怎样处理存储在存储器中的数据 and 指令及怎样处理包含在处理器中的指令的信息。就是这些性质，使计算机与计算器得以区别，使我们能够称计算机为“存储程序机器”。

计算机的主存储器可以比作一个行政办公室中的公文柜，它有固定的且有限的容量，用于存放手边需处理的问题的资料。我们需要存储的信息总数很可能超过存储器的容量，就如同行政人员要保存的资料超过公文柜所能存放资料的总数。

行政人员可在多个档案室或存储室中附加的公文柜中保存多余的——不十分紧急的资料。计算机把多余的信息存放在称为“辅助存储器”的磁盘和磁带中，其内容可以通过专用的输入输出设备传送到计算机的主存储器中或从主存储器中传出到磁盘磁带上。

我们一旦有了计算机，也许愿意用它解决问题。通过写一个算法告诉计算机怎样解决一个特定问题或某些问题，为了获得一个正确的解，必须紧跟着—组指令。算法必须准确且无二义性，因为计算机不象人，没有通常的辨别能力；算法也必须是有限的可描述的，如果不能输入完整的指令组，计算机就不能执行完整的指令组；最后，算法必须是有限可执行的，如果计算永远不能完结，那么答案将永远不能产生。

程序是用计算机能够“懂得”的语言写的算法，实际上，计算机只明白电子信号，它根据算法中遇到的上下文，有区别地解释信号序列。因此，程序是用语言书写的算法，这种语言可以翻译为一种合适的电子信号。这种翻译是由称为“编译器”的程序来完成。

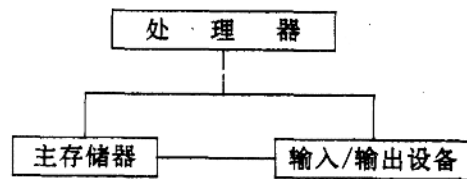


图 1.2 计算机系统的主要部件

在计算机的基本活动中编译器和其它装入的程序指挥计算机的动作，这些程序称为系统软件。负责控制程序执行的处理部分称为操作系统，也是系统软件的一部分。计算机由硬件和软件组成，处于运行状态的系统也包括正在执行的用户程序。

在编写算法和程序的过程中，我们必须考虑硬件的能力，例如，我们不能要求 TRS80 机画出一个房屋，房屋绘画机可以做到，但我们不能称其为计算机，因为它的主要功能是处理绘画，如果机器处理的是信息，我们就把它称为计算机。

把一台机器称为计算机的另一个根据就是它是通用的，可以解决各种不同的问题。为了识别我们上面讨论的部件，让我们暂时忽略这种要求，先考察一下专用计算机，一个智能恒温器。假设我们可以编程使恒温器在日间和夜间保持指定的温度。

1. 输入设备是“面板”，它有“读入”现行温度的传感器，给出当前时间的时钟和接收日间与晚间说明及日间与夜间温度。

2. 输出设备是指示当前温度的仪表和控制炉子温度的机械。

3. 需要进行的唯一计算是比较——当前时间与“日间”和“夜间”及当前温度与予定的温度。处理器必须能完成这些比较，同时也包括运转系统，它执行下面的算法：

a. 检查面板是否送入新指令，是否有任何新指令，设置“日间开始”“夜间开始”，“日间温度”和“夜间温度”。

b. 如果“当前时间”在“日间开始”与“夜间开始”之间，那么设置“需要温度”为“日间温度”，否则设置“需要温度”为“夜间温度”。

c. 如果“当前温度”低于“需要温度”，那么置“炉子控制”为接通，否则，如果“当前温度”高于“需要温度”，置“炉子控制”为断开。

4. 用户通过选择“日间开始”，“夜间开始”，“日间温度”和“夜间温度”中的一个或多个值，对恒温器编程。这是一个非常简单的程序，它只由改变的数据和数据新值的选择组成。

许多算法和程序是比较复杂的，它不仅包含数据，也包含有关数据的事件和处理事件的指令，它们被设计成按不同的输入数据组执行，以便于它们被用于处理一类问题，而不是一个特定问题。上面的运转系统程序接收来自炉子传感器、时钟和用户的输入及当输入值不同时选择不同的活动。

一个真正的计算机不同于专用恒温器，在计算机中它根据给它的程序完成不同的任务。一个程序可能使计算机结算账目，另一个程序让它下象棋，第三个程序可能让它解一组联立方程。正象先前提到的那样，这样的机器称为通用机。

### 1.3 计算机的发展历史

计算机系统包括硬件系统和软件系统。所谓硬件系统指构成计算机的所有电子的、光学的、机械的和磁性的元件和器件；它完成信息变换、信息存储、信息传送和信息处理功能，是计算机系统快速、可靠、自动工作的基础。软件系统是为方便用户，充分发挥计算机硬件功能而编制的所有程序及所有文件的总称。对于计算机系统，硬件是物质基础，软件担负指挥功能。硬件与软件是同等重要的，但它们对有效的求解问题的贡献是相当不同的。计算机的任何历史都必须注意硬件和软件两个方面的发展。现在我们就沿着硬件和软件两个方面来追溯计算机的发展历史。

第一台计算机是 1946 年诞生的，名字为 ENIAC, Electronic Numerical Integater and Calculator, 重达 30 吨，运算速度只有 5000 次/秒，尽管该机耗费大，功能不完善，但却是科学史上

一次划时代的创新，它为计算机的发展奠定了基础。自从这台计算机问世以来，计算机的发展大致经过了四代变化：

1946年开始的电子管计算机，使用真空管技术，用“第一代”这个名词表示计算机的硬件。计算机的体积庞大，运算速度一般为每秒几千次至几万次，可靠性较低，成本高。

“第二代”这个名词用于1958年以晶体管技术为基础发展起来的计算机。运算速度一般为每秒几万次至几十万次，体积缩小，可靠性提高，成本降低。

“第三代”为1965年开始由集成电路组成的计算机。运算速度提高到每秒几十万次至几百万次。体积进一步缩小，可靠性进一步提高，成本进一步降低。在此期间形成机种多样化生产系列化，使用系统化。

1971年开始的由大规模集成电路组成的计算机被称为第四代计算机，运算速度提高到每秒几百万次至几千万次。体积更进一步缩小，可靠性更进一步提高，成本更进一步降低。

目前，世界各国正在研制第五代计算机，这一代计算机的目标是加强智能，方便使用；用自然语言、图形、图象和文件进行输入输出；能用自然语言进行对话方式的信息处理；能处理和保存知识，配备各种知识库；能够自学习推理帮助人类扩展自己的才能。

计算机软件的发展受到计算机硬件发展的推动和制约。其发展过程大致可以分为三个阶段。

从第一台计算机上的第一个程序出现到实用的高级语言出现以前为第一阶段（1946—1956年），计算机主要应用于计算，编程所用的工具是低级语言，即以机器基本指令集为主的机器语言和稍加符号化的汇编语言，程序设计和编制工作复杂、繁琐、费时和容易出错。

第二阶段是从实用高级程序设计语言出现以后到软件工程出现以前（1956—1968年）。计算机除用于科学计算外，还用于大量的数据处理问题，其性质与科学计算有明显区别，涉及非数值数据。这时出现了高级程序设计语言、操作系统，也开始出现数据库及其管理系统。软件复杂程度迅速提高，研制周期长，可靠性难以保证，这就是所谓的“软件危机”。

为了解决“软件危机”，软件工程应运而生。软件工程出现以后（1968年以来）为软件发展的第三阶段。这一阶段普遍采用现代高级语言，即具有并行成分和实时成分的模块化语言，开始出现了网络软件和软件工程方法。

按照应用和虚拟机的观点，软件可分为系统软件、支援软件和应用软件三类。

系统软件是最接近硬件的一层，它与具体的应用领域无关，其它软件一般都通过系统软件发挥作用，如操作系统、编译程序等。

支援软件，顾名思义其为支援其它软件的编制和维护的软件，直接促进软件的发展，如环境数据库，各种接口软件和工具组等。

应用软件为特定应用领域专用的软件。

软件的上述分类不是绝对的，而是互相交叉和变化的。有些软件如操作系统可看作系统也可看作支援软件，图形软件、网络软件原来看作为应用软件，后来被看作为系统软件。

#### 1.4 本章练习

1. 说明计算机系统的主要部件及其功能
2. 区别：a. 硬件和软件 b. 软件和系统软件 c. 一个算法和一个程序
3. 为什么称计算机为通用机？
4. 列出下述领域中计算机的专门用途

- a. 医学    b. 政府    c. 教育    d. 工程    e. 军事防御系统  
f. 运动    g. 企业    h. 艺术    i. 图书馆    j. 法律的执行

5. 说明含有大量人的个人信息数据库的优缺点。
6. 列出社会计算机化的好处。
7. 按你的观点，计算机科学的哪个子域争论最大？列出你的理由。

## 第二章 算法和问题求解

设计和制造计算机系统的目的是为解决问题提供一个迅速,有效且经济的方法。在这个意义上,我们不确切地定义了“问题”和范围,从解一个复杂的微分方程组到准备顾客姓名和地址的邮件标签。为了利用计算机解决问题,我们必须精确地指出怎样解决这个问题,并赋予计算机完成这个过程的能力。因此,计算机科学不仅仅是对计算机系统的研究,也是对问题求解的研究。

本章集中讨论问题求解过程,着重算法结构的开发。

### 2.1 算法的性质

算法是一个形式名字,它给出怎样求解某一类特定问题的说明。在日常工作中,我们遇到许多算法,尽管通常称其为“烹饪法”,“装配手册”或“说明”。例如,一个巧克力小饼的烹饪法,指出怎样做一批同种类型的小饼;模型飞机配套零件的说明,指出怎样构造飞机模型。

为了开发一个算法,问题本身必须很好地定义,例如,为很好定义做小饼的任务,在成功地制订烹饪法之前,必须确定甜饼的类型和数量。系统分析中的许多努力都是对开发一个复杂问题进行精确定义和描述的贡献,只有完成定义后,系统分析员才能开始构造算法。

让我们检查一下图 2.1 所示的巧克力小饼的烹饪法,并确定算法的性质。这个烹饪法给

输入: 1 杯油
1 杯红糖
3/4 杯沙糖
2 个鸡蛋
1 茶匙香精
1 茶匙发酵苏打
1/2 茶匙盐
7/3 杯筛面粉
3/2 杯半甜巧克力片
输出: 5 打巧克力小饼
说明: 搅拌油,红糖,沙糖和香料 100 次;加两个鸡蛋,搅拌 200 次,然后,加入 1/2 茶匙盐和一茶匙发酵苏打;加 7/3 杯面并搅拌 100 次。加 3/2 杯巧克力片,搅拌 25 次;用 1 茶匙糊状物做 60 个弧形团,在 375 度的烘炉中烘 8 分钟;从炉中取出并冷却。

图 2.1 巧克力甜饼烹饪法

出了制作巧克力小饼的明确过程,列出了配料(油,鸡蛋等),配料的形式(筛面分),和每种配料的数量(1 杯油,2 个鸡蛋)。这些配料是算法的输入,并组成了算法指令的应用项,烹饪法的输出,5 打巧克力小饼,它是准确地按烹饪法去作而获得的。这对应于算法的输出,或一个人执行算法所得到的结果。

烹饪法的说明指明了要进行的操作（加两个鸡蛋），执行的次序（加 2 个鸡蛋，搅拌 200 次，然后加入 1/2 茶匙盐，...）和必须重复进行操作的次数（搅拌 200 次）。操作的次数，包括重复操作的次数是有限的，因此，保证了烹调能达到烹饪法的终点并产生予定的输出（假定烹饪法是正确的，且勤奋地按着它去做）。

我们看到的烹饪法，一般称为算法，必须具备下述性质：

1. 算法输入的说明，包括任何特征，诸如，类型，形式或数量
2. 算法输出的说明——具有规定输入的算法的执行结果
3. 从输入到输出必须执行的操作的明确描述
  - a. 必须给出算法中执行操作的顺序
  - b. 必须明确规定下一步操作，不允许从几个可用操作中选择
  - c. 操作的次数，包括重复操作，必须是有限的（然而，准确的操作次数，对于不同的输入是变化的）

具有性质 3b 的指令称为限定，意思是对于给定的一组输入，不管算法执行多少遍，输出结果是相同的。在我们的例子中，按烹饪法操作，总是产生 5 打巧克力小饼，并不是第一次产生 3 打燕麦饼，第二次产生 5 打巧克力饼，而第三次也许是一种奶油糕点。在计算机科学中，一直分析无限过程，且在理论上具有重要意义，但本书所研究的过程都是有限过程。

如果输入满足输入说明，我们不仅要求算法连续工作，而且要求它最终产生理想的结果。性质 3c 保证算法每次执行都在有限的时间内结束，算法一旦结束，则产生予定的输出，而且算法对于将来的使用也是有效的。

虽然，烹饪法是算法的一个例子，但制作巧克力小饼超出了当前计算机系统的能力。然而，已经制造了进行重复装配线工作，在办公大楼中传递邮件和按照打印的英文命令管理大楼的机器人。有理由希望将来某一天机器人将具有读入，解释和执行烹饪法的能力。

## 2.2 计算的算法

当代计算机可执行的算法类型是处理输入数据，产生一组输出数据。输出数据可能是一个问题的答案或解，一组顾客邮件表，一个修改的银行事物磁带，或者是一个法人的报告。我们称这样的问题为“计算”。

本章，我们将用特定的程序设计语言，研究计算算法，但不陷入它执行时的细节。然而，我们需要一些描述算法的方法，为此，介绍一种简单的类英语程序设计语言（PDL）。读者将发现，用这种语言表示的算法非常容易转换为实际的程序设计语言，如 PASCAL，FORTRAN，PL/1 等等。

我们的 PDL 将计算的公式具体化，用类英语的句子表示命令，IF—THEN—ELSE 结构指出条件执行，WHILE 结构说明循环执行一个指令序列。在下面几个段落中，通过例子介绍并解释每个命令。

### 2.2.1 赋值语句

在计算算法中，我们需要保留数据项的值，对它们进行运算及提交他们的新值的方法。为此，我们使用助记符（“助记符号”意思是“帮助记忆”）提交一个数据项和算术表达式来指明数据当前值的计算。例如，表示一个工资的数据可以命名为“SALARY”或者“SAL”，把名字放入以字母为顺序的表中的计算，可以命名为“SORT—NAMES”。

指令  $ANSWER \leftarrow EXPRESSION$

描述计算右侧表达式的值与左侧助记符相联系，我们说变量 ANSWER 被“赋予”适当的值。无论何时提交这个符号，它将有被赋予的新值，然而，后面的程序再次在赋值指令的左面使用它时，可以赋给它一个新值，原值被丢掉。一个助记符号在任何时刻只有一个值，例如，指令  $SUM \leftarrow 25 + 4$

给助记符号 SUM 赋一个值 29，执行后续指令

$TIMES2 \leftarrow SUM \times 2$

给助记符号 TIMES2 赋一值 58，没有替换符号 SUM 的值。执行指令

$SUM \leftarrow SUM + 15$

符号 SUM 的值增加 15，这就是 SUM 的当前值加 15，计算后的值 44 作为新的当前值赋给符号 SUM。

一组指令总是按指令出现次序一次一条地顺序执行。上面提出的步骤顺序出现在程序中为： $SUM \leftarrow 25 + 4$

$TIMES2 \leftarrow SUM \times 2$

$SUM \leftarrow SUM + 15$

如果一个人记住了三个要点，符号指令是容易解释和明白的：

1. 总是用助记符号的当前值计算箭头右侧表达式的值；
2. 箭头左边符号总是被赋予新值这个值是步骤 1 计算的右侧表达式的值；
3. 箭头左边符号的原值被清除，而其它符号的值不变。

问题 2. 1 赋值语句

模拟执行下述指令组中的每条指令，描述赋给助记符号的值怎样变化。

(1)  $A \leftarrow 4$

(2)  $BAT \leftarrow 6 + 3$

(3)  $CA \leftarrow 4$

$T \leftarrow 3 + CA$

指 令                      解                      释

---

(1)  $A \leftarrow 4$           现在符号 A 的值是 4 .

(2)  $BAT \leftarrow 6 + 3$       现在符号 BAT 的值是 9 (3 和 6 的和)

(3)  $CA \leftarrow 4$           现在符号 CA 的值是 4，下一个符号 T 接收值 7，它由 3 和  $T \leftarrow 3 + CA$  符号 CA 的当前值求和确定的

### 2. 2. 2 循环

我们考查一个简单的计算问题，求一组整数的和，并且为了启发循环的概念或算法部分重复执行的概念，分析几个求和的算法。

可能的算法 A：

输入：整数 4 和 6

输出：整数 4 和 6 的和

指令： $SUM \leftarrow 4 + 6$

Output the value of SUM



这个算法成功地输出整数4与6的和，然而为了求任何其它成对整数的和，必须重新书写这个算法。该算法只能求解特定问题，而不能求解一类问题。让我们重新形式化算法 A，计算任一对三位整数和。

可能的算法 B:

输入：两个任意整数，每个整数绝对值小于1000

输出：一对输入的整数和

指令：X ← first input integer

Y ← second input integer

SUM ← X + Y

Output the value of SUM

这个算法成功地输出任何三位数字的整数和。如果输入数据是44和236，那么，X 接收44，Y 接收236，SUM 接收280，然后输出 SUM 的当前值280，这就是我们期望的答案。虽然这个算法比算法 A 更一般化，但它仍受到限制，只能对一对整数求和，如果我们想求 4个整数和，我们的算法变为：

X ← first input integer

Y ← second input integer

Z ← third input integer

W ← fourth input integer

SUM ← X + Y + Z + W

Output the value of SUM

很显然，这样的对于一般情况的过程是不可接受的，试想，我们需要求1000个整数和的指令！必须找到一个更好的方法，引导我们到循环的概念或重复执行一组指令的概念。考虑下述算法：

SUM ← 0

Repeat the sequence of instructions between BEGIN1 and END1

BEGIN1

X ← next input integer

SUM ← SUM + X

END1

开始，符号SUM 的值设置为 0，“Repeat”指出包括在 BEGIN1和 END1 之间的两条指令顺序执行，然后再次执行，等等。注意符号 SUM 用作累加器——它累加任何时刻输入的所有整数和。这一组指令成功地计算任意一组整数的和。但它违背了算法的一个基本性质——即，操作的次数，包括重复，必须是有限的。因此，我们必须重新编制指令，因此，最终要结束括号中操作的重复。算法 C 和 D 都能实现我们的目的。

算法C:

输入：任何一组有限的整数

输出：一组输入整数的和

指令：SUM ← 0

While input data remain, perform the sequence of  
instructions between BEGIN1 and END1