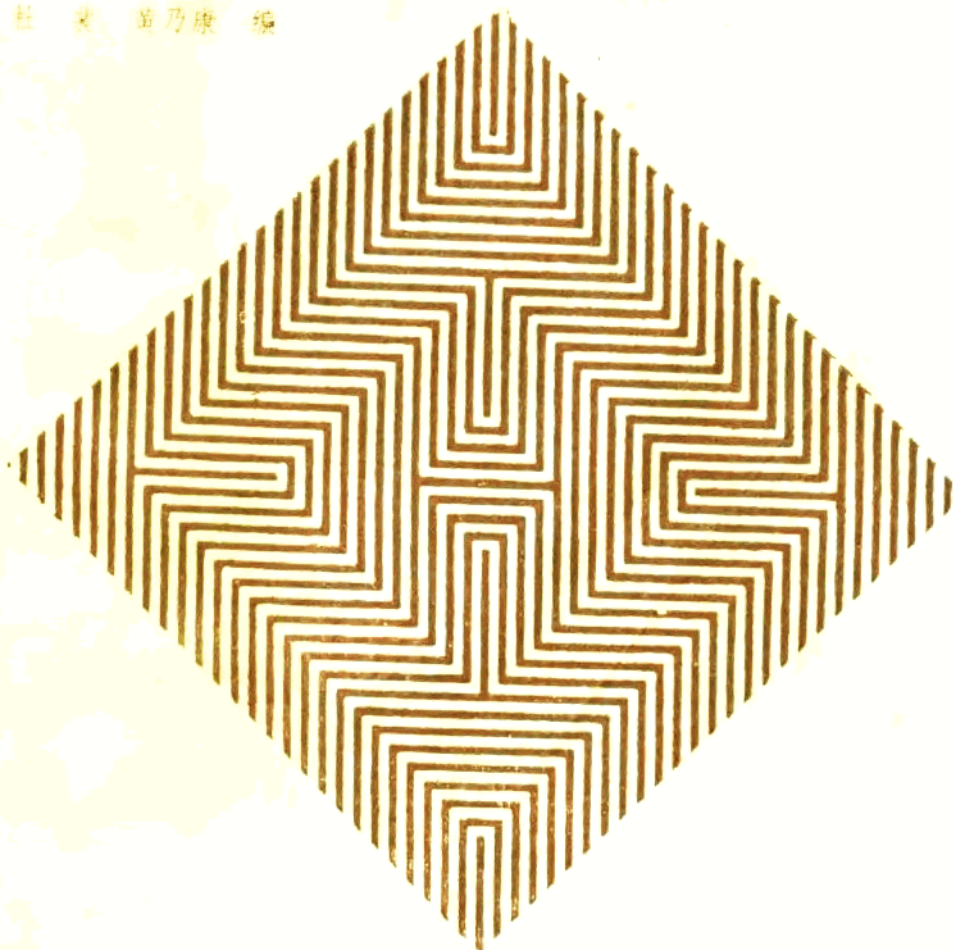


计算机辅助 工艺过程 设计原理

杜 英 黄乃康 编



2

北京航空航天大学出版社

前 言

本书根据航空院校机械制造专业计算机辅助工艺过程设计原理课程教学大纲编写而成。

多年来,国内工科院校早已在机械制造工艺学、计算机辅助制造或成组技术等课程中讲授计算机辅助工艺过程设计原理。为了适应计算机辅助设计和计算机辅助制造技术的发展,不少院校开设了计算机辅助工艺过程设计原理课程。本书就是根据编者多年讲授计算机辅助工艺过程设计原理和从事相应课题研究的经验,并参阅了近年发表的大量资料文献,整理编写而成。

本书第一章从制造系统概念出发,讨论计算机辅助工艺过程设计的意义、现况和发展。第二章和第三章是计算机辅助工艺过程设计的基础知识,编者根据全面而扼要的原则,力求以较少的篇幅介绍这些知识。第四章、第五章、第六章和第七章全面介绍了计算机辅助工艺过程设计的基本原理和系统设计方法,选择介绍国内外一些典型系统。由于人工智能技术是计算机辅助工艺过程设计的发展方向,本书对此也作了相应介绍。第八章评述发展趋势。

本书由北京航空航天大学杜裴和西北工业大学黄乃康编写,由杜裴任主编,第一章引论,第三章零件信息描述,第四章派生式计算机辅助工艺过程系统和第七章人工智能技术在计算机辅助工艺过程设计中的应用由杜裴编写;第二章数据结构,第五章生成式计算机辅助工艺过程设计系统,第六章工艺过程设计中的工序设计和第八章计算机辅助工艺过程设计的发展趋势由黄乃康编写。吴风英、杨美伯、田锡天、刘志刚和白昌军同志参加本书的编写工作。

在本书的编写过程中,北京航空航天大学 and 西北工业大学的许多老师提出不少可贵的意见,谨向他们表示谢意!

本书原稿请北京理工大学胡永生教授审阅,提出了许多宝贵的意见和建议,在此谨致衷心谢意!

本书编者虽然都有多年讲授本课程和从事相应课题研究的经验,但缺点和错误在所难免,恳请读者批评指正。

编 者

1990年9月

目 录

第一章 引 论

§ 1-1 机械制造与制造系统	1
§ 1-2 计算机综合制造系统	1
§ 1-3 发展计算机辅助工艺过程设计的意义	3
§ 1-4 现状与发展	4

第二章 数据结构

§ 2-1 引言	8
§ 2-2 线性列表 (<i>List</i>) 结构	8
§ 2-3 单向链表 (<i>Linked List</i>)	12
§ 2-4 循环链表与多重链表	17
§ 2-5 串 (<i>String</i>)	19
§ 2-6 树 (<i>Tree</i>)	23
§ 2-7 图 (<i>Graph</i>)	32
§ 2-8 排序与查找	37
§ 2-9 数据管理技术与数据库系统	43

第三章 零件信息描述

§ 3-1 零件分类编码	53
§ 3-2 零件表面元素描述法	76
§ 3-3 机械零件计算机辅助设计	88

第四章 派生式计算机辅助工艺过程设计系统

§ 4-1 派生式 <i>CAPP</i> 系统的原理与设计方法	100
§ 4-2 典型系统分析	112

第五章 生成式计算机辅助工艺过程设计系统

§ 5-1 生成式 <i>CAPP</i> 系统的原理与设计方法	120
§ 5-2 判定表与判定树	127
§ 5-3 典型系统分析	141

第六章 计算机辅助工艺过程设计中的工序设计

§ 6-1 工序设计的内容.....	151
§ 6-2 工艺尺寸换算.....	153
§ 6-3 切削参数选择与加工过程优化.....	157
§ 6-4 CAOS: 一种生成式工序设计系统.....	161

第七章 人工智能技术在计算机辅助工艺过程设计中的应用

§ 7-1 专家系统基本概念.....	167
§ 7-2 知识表达方法.....	170
§ 7-3 应用人工智能技术的工艺过程设计系统.....	188
§ 7-4 用于人工智能的程序语言.....	194

第八章 计算机辅助工艺过程设计的发展趋势

§ 8-1 CAPP的发展趋势.....	205
§ 8-2 集成化CAPP系统评介.....	208

参考文献.....	223
-----------	-----

第一章 引 论

§ 1-1 机械制造与制造系统

机械制造业是国民经济最重要的部门之一。机械制造业不仅直接提供人民生活所需的产品,而且为国民经济各部门提供技术装备。农业、石油、化工、矿山、电力、交通运输……等部门全都离不开机械制造业。一个国家的机械制造业水准直接关系到国家的工业发展和国防现代化。机械制造业状况是衡量国家技术力量的重要标志。当前,国际和国内市场竞争十分激烈,机械制造业对于保证产品质量、缩短生产周期、降低成本和提高经济效益都起着关键作用。工业化国家都把加速制造业的发展看成是具有长远战略意义的现实任务。在我国,尽快提高机械制造业的生产技术和管理水平,提高产品质量,提高经济效益,提高生产力,已成为航空航天工业当前和今后长期的重要艰巨任务。

在传统的狭义定义中,制造是指由原材料变成为需要的最终产品的变换过程,是通过一系列适当的方法手段和能量的作用,使材料发生规定的变化而成为所需要的产品的变换过程。

五十年代以后,随着生产的发展和科学技术的进步,以及人们认识的深化,制造的概念有了很大的发展。1983年国际生产工程研究会(International Institution for Production Engineering Research CIRP)对制造两字赋予新的定义:制造是制造业内的一系列有关活动和过程,其中包括产品设计、材料选择、规划、制造生产、质量保证、管理和销售等。国际生产工程研究会还确定了制造系统的定义:制造系统是“制造业内为进行制造生产而形成的一种组织。在机械和电气工业中,制造系统一般为许多功能的集成组合,这就是销售设计、制造生产和发运等功能。研究功能可以为其它一种或数种功能提供服务”。

§ 1-2 计算机综合制造系统

在上节中说到制造概念的发展,是与自动化技术发展相关联的。在五十年代以前,机械工业的生产自动化,仅限于制造过程的自动化,主要是加工自动化。汽车工业中以自动机床和组合机床组成自动化流水线生产,是加工自动化的典型代表。这是所谓刚性自动化。刚性自动化适用于大量生产。刚性自动化流水线是专为某一产品零件设计制造的。当产品设计改型,这条自动流水线便不能使用。这种自动化的特点是生产效率高,但产品很难变更,对市场需求响应非常慢,而且要花费大量投资,不适合中小批量生产。由于在制造业中小批量生产日益占主要地位,如美国商业部曾指出,美国有95%的产品是在批量等于或小于50的规模下生产的。而且,一些传统的大量生产企业,出于市场竞争的需要,必需迅速响应市场变化,以新制胜,CAD/CAM技术就在这样的背景下得到迅速发展。

1952年在美国麻省理工学院(Massachusetts Institute of Technology MIT)研制出

第一台数字控制机床，这是一台三坐标立式铣床。数控机床的问世，开创了计算机技术用于制造系统的新纪元。十八世纪后半叶兴起的工业革命，使制造技术从手工为基础的工场手工业过渡到使用机器的工厂生产，使生产发生质的变化，这是制造技术发展上的第一次飞跃。数字控制机床的出现，实现了用计算机器自动控制机器，开始了制造技术发展上的第二次飞跃。

世界现在还正处在这个飞跃的过程中。过去，人们主要依靠手去操作机器。现在，机器接受来自计算机网络的信息，人机是分离的，机器可以在无人照管下自动运行。计算机综合制造系统 (*Computer Integrated Manufacturing System*, 或译为计算机集成制造系统, *CIMS*) 将是这个飞跃的重要成果。发达国家都以在本世纪末或下世纪初实现计算机综合制造系统为奋斗目标。我国的高技术计划，也将发展计算机综合制造系统列入研究课题。

计算机综合制造系统就是将计算机集成应用于企业的各个领域，将企业各部门的工作置于计算机统一管理和统一控制之下。一个计算机综合制造系统可以由若干个子系统组成。欧洲信息技术研究战略计划 (ESPRIT) 的计算机综合制造系统由下列五个子系统组成：

1. 计算机辅助设计 (*Computer Aided Design, CAD*)
2. 计算机辅助生产工程 (*Computer Aided Production Engineering, CAPE*)
3. 计算机辅助制造 (*Computer Aided Manufacturing, CAM*)
4. 计算机辅助存贮和运输 (*Computer Aided Storage and Transportation, CAST*)
5. 计算机辅助生产规划 (*Computer Aided Production Planning, CAPP*)

计算机辅助生产工程的任务是决定怎样制造出设计确定的产品。计算机辅助生产工程包括下列几个主要领域：

- 1) 计算机辅助工艺过程设计 (*Computer Aided Process Planning, CAPP*)；
- 2) 计算机辅助工厂布置；
- 3) 计算机辅助零件编程；
- 4) 计算机辅助生产工具和夹具设计；
- 5) 材料运输。

需要指出，目前，对计算机综合制造系统所下的定义，对计算机综合制造系统应分为哪几个子系统，以及这些子系统的名称符号，都不统一。例如，大多数国家及大多数文献资料将计算机辅助工艺过程设计简称为 *CAPP*，但也有把它看成是计算机辅助生产规划的简写，多数人把计算机辅助工艺过程设计简称为 *CAPP*，而德国人却习惯称其为 *CAP*。表 1-1 表明各国在名词定义上的差异。

从以上叙述中可以看到，计算机辅助工艺过程设计是连接 *CAD* 和 *CAM* 的中间环节。*CAD* 数据库信息只有经过 *CAPP* 系统才能变成 *CAM* 的加工信息。生产管理和计划调度等部门，也必须依靠 *CAPP* 系统的输出数据。正是由于这个原因，设在美国的计算机辅助制造国际组织 *CAM-I* (*Computer Aided Manufacturing-International*) 在选择研究开发项目时，认为工艺过程设计是企业各部门信息交叉汇集的环节，从而决定在数控研究之外第一个开发项目就是计算机辅助工艺过程设计。在 *CAPP* 发展史上具有里程碑意义的、最著名的、也是应用最广泛的 *CAPP* 系统，就是 *CAM-I* 基于上述分析而开发成功的。

表1-1 对CAD/CAM名词的不同定义

设 计	制 造 准 备	制 造	使 用 地 区
CAD	CAPP CAP	CAM	美国、日本、 西德、经互会
CAD			西柏林、西德、 美国(标准委员会)、 美国、西德 (部分)
CAD		CAM	
	CAM		
CAD	CAM		

§ 1-3 发展计算机辅助工艺过程设计的意义

传统上, 工艺过程设计由有丰富生产经验的工程师负责。进行工艺过程设计的工程师必需:

1. 具有丰富的生产经验;
2. 熟知企业的各种设备使用情况;
3. 熟知企业内各种生产工艺方法;
4. 熟知企业内各种与生产加工有关的规范;
5. 熟知与生产管理有关的各种规章制度;
6. 能与有关各方保持友好协作。

具有丰富生产经验的工艺工程师, 在发达国家常感人数不足。美国空军的一次调查报告称, 工艺过程设计人员一般年龄应在40岁以上, 并有丰富的生产车间工作经验。美国工业现在需要二十到三十万这样的人员, 而实际只有十五到二十万人。英国工艺工程师平均年龄为55岁。对年龄数据的统计, 反映了要求工艺工程师有多年的生产实践经验。1986年美国的P. A. Event在美国空军材料实验室制造工程部资助下, 对美国的一些学者和专家进行了调研, 大都认为, 工艺过程设计可以应用人工智能的多种技术。因此, 工艺专家人数不足, 可以部分地应用具有智能的工艺过程设计专家系统来弥补。

人工设计工艺规程, 存在缺点之一是一致性差, 而且质量不稳定, 难以达到优化的目标。

我们把下面这样的事实称为一致性差。

有人曾进行过试验: 将图1-1的产品零件图分发给四名经历各不相同的工艺工程师(甲、乙、丙、丁)请他们分别给这个零件设计工艺规程。得到的结果列于表1-2。

象这样一种现象, 即同样的产品零件由不同的工艺工程师设计工艺规程, 得到的结果不

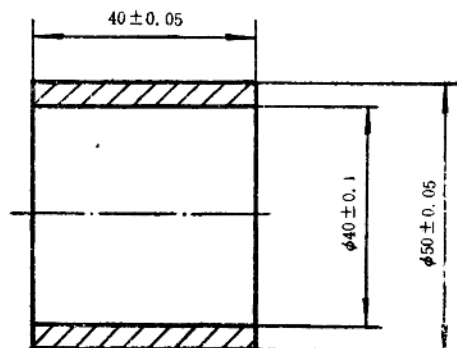


图 1-1

表 1-2

工序号	甲	乙	丙	丁
1.	车端面	分两次钻 a. $\phi 20$ b. $\phi 38$	车外圆	分两次钻 a. $\phi 30$ b. $\phi 40$
2.	钻孔 a. $\phi 10$ b. $\phi 38$ c. $\phi 40$	车端面	一次钻至 $\phi 40$	车外圆 $\phi 50$
3.	车外圆 $\phi 50$	切断	车端面	车面
4.	切断	车另一端面	切断	切断
5.	车另一端面	车外圆 $\phi 50$	车另一端面	车另一端面
6.		镗孔 $\phi 40$		

相同,我们称之为一致性差。相似的零件,原应具有相似的工艺过程,却人为地为它们确定了不同的工艺过程。这样,对组织生产和提高工人操作熟练程度,都很不利。原本可以在同样的机床上生产,却不必要地增购新机器。美国辛辛那提工厂,生产 425 种直齿轮,人工设计工艺规程的结果,使得有 377 份不同的工艺规程,使用 57 台不同的机床。

人工设计工艺规程的另一个缺点是速度慢,费时,不能适应机械制造业中日益增强的产品更新换代快的要求:一个新产品问世,要经历设计—生产准备—加工生产等三个大阶段。生产准备包括工艺过程设计、专用夹具和刀具设计、专用机器及其它工艺装备设计等。工艺过程设计所占的比重,约为整个生产准备时间的 40%。

美国联邦技术研究中心的报告称,一个先进的 CAPP 系统可以使好几个方面获得节约,并使在制品数量减少。

在一个零件的成本中,其费用组成以及运用 CAPP 系统可节省的百分比如下:

项 目	费用占总费用的百分比	项 目	节省的百分比
工艺过程设计费	8 %	工艺过程设计费	58 %
材料费	23 %	材料费	4 %
工时费	28 %	工时费	10 %
返修及废品费	4 %	返修及废品费	10 %
刀具费	7 %	刀具费	12 %
管理、利润等	30 %		

根据上面的数字,可以算出,采用 CAPP 可以使零件生产成本降低 9.6%。这个数字还不包括采用 CAPP 所带来的全部利益。例如,采用 CAPP 后可以使生产准备时间减少,产品试制周期缩短,并增强企业竞争能力等。

§ 1-4 现状与发展

工艺过程设计自动化这一课题是在六十年代后期开始提出来的。在 CAPP 发展史上具有

里程碑意义的是CAM-I于1976年推出的CAM-I'S Automated Process Planning 系统。取其字首的第一个字母,称为CAPP系统。现在对CAPP这个缩写表示法虽然还有不同的释义,但把计算机辅助工艺过程设计称为CAPP已经举世公认。在1976年以后,特别是近几年来,对CAPP的研究已经成为研究制造系统的一个大课题。

实现CAPP有两种方法:派生法(Varoiant Approach)和生成法(Generative Approach)。在CAPP发展初期,主要模式是派生法。人们对生成法也进行了很多研究探索。人工智能技术的发展,为CAPP进一步发展开辟了新的道路。进入八十年代后,以应用人工智能技术为基础的生成法CAPP系统,受到广泛重视。美国机械工程协会(ASME)在1985年和1986年连续召开学术会议,讨论智能化制造系统和智能化工艺过程自动设计。英国于1984年在Nottingham大学召开学术会,讨论专家系统在制造系统中应用。制造系统可以应用专家系统的领域是很广泛的。在这次会议上提出的论文,大多为工艺过程中的专家系统。1986年,英国专门召开会议,交流讨论专家系统在CAPP中的应用。国际生产工程研究会在1985年1月和1987年6月两次召开计算机辅助工艺过程设计专题学术讨论会,发表了大量论文。

现在已经开发的CAPP系统,一部分是学术性的,另一部分是公开出售的商品。表1-3中摘录介绍一部分CAPP系统:

表 1-3

系 统 名 称		CAP	AUTAP	AUTOPLAN	APPAS	AUTOPROS	CAD/CAM
零件类型	回转体		○	○		○	
	棱形体				○		孔加工
	钣 料	○	○			○	
设计方法	派生法			○			
	生成法		○		○		○
自动化功能	工艺路线/工序设计		○	○	○	○	○
	机床选择		○	○			
	刀具选择		○	○	○		○
	夹具选择			○			
	确定加工参数		○	○	○		○
	确定刀具路径		○				
用户提供		工艺规程	决策表	标准工艺过程			
人工编辑		○		○			
商品C/学术性A			C	C	A	C	A
开发单位		Lockheed	阿亨大学	Metcut	普渡大学	NAKK	
注 释		Tulkoff 1981	Eversheim 1980	Tempelhof 1980	Wysk 1977	Chisolm 1973	APPAS的扩展 Chang, Wysk 1980, 1981

续表 1-3

系统名称		CAPP	CIMS/PRO	CPPP	DCLASS	GARI	GENPLAN
零件类型	回转体	○		○	○		○
	棱形体	○	○		○	○	○
	钣金料	○			○		○
设计方法	派生法	○					
	生成法		○	○	○	○	○
自动化功能	工艺路线/工序设计	○	○	○	○	○	○
	机床选择		○	○	○		○
	刀具选择		○	○	○		○
	夹具选择			○			
	确定加工参数						
	确定刀具路径		○				
用户提供		标准工艺		加工模型	决策树	产生式规则	
人工编辑		○		○	○		
商品C/学术性A		C		C	C	A	C
开发单位		CAM-I	Iwata	UTRC	Allen/U. Utah		Lockheed
注 释		LINK 1976	Iwata 1980	Dunn, Mann 1978	1980	MACLISP Descotte 1981	Tulkoff 1981
系统名称		MIPLAN	PI-CAPP	RPO	XPS-I	XPS-E	CAPSY
零件类型	回转体	○	○	○	○	○	○
	棱形体	○	○		○	○	
	钣金料		○				
设计方法	派生法	○	○	○	○		
	生成法				○	○	○
自动化功能	工艺路线/工序设计	○	○	○	○	○	○
	机床选择				○		○
	刀具选择				○		○
	夹具选择				○		○
	确定加工参数			○	○		○
	确定刀具路径						○
用户提供		标准工艺	标准工艺		决策表	产生式规则	
人工编辑		○	○	○	○		
商品C/学术性A		C	C	C	C	C	
开发单位		GE/OIR		GE/Metcut	CAM-I	CAM-I	柏林大学
注 释		TNO 1981	有图形输出 1981	AUTOPLAN 的发展 1980	CAM-I 1980		研制中 1982

续表 1-3

系 统 名 称		TOJCAPP	BHCAPP	BHCAP	EXCAPP	BITCAPP	NHCAPP	CAOS
零件类型	回转体	○	○	○	○	○	○	○
	棱形体							
	钣金							
设计方法	派生法	○	○			○	○	
	生成法		○	○	○	○	○	○
自动化功能	工艺路线/工序设计	○	○	○	○	○	○	○
	机床选择	○	○	○	○	○	○	
	刀具选择	○	○	○	○	○	○	○
	夹具选择		○	○	○	○	○	
	确定加工参数	○				○		○
	确定刀具路径							
用户提供				产生式规则	产生式规则			
人工编辑		○			○			
商品C/学术性A			A	A	A		A	
开发单位		同济大学	北航	北航	北航	北工	南航	西工大
注 释		1983	1984 (BASIC)	1987 (PASCAL)	1988 (PROLOG)	1987 FORTRAN	1987	1982一版 1986二版 FORTRAN

据预测,到1995年,50%的零件制造和装配工艺规程将由计算机产生。

尽管现在有不少CAPP系统,但工厂实际使用情况却不能令人满意。原因是多方面的。一个重要因素是,机械制造业范围很广,不同的工厂,甚至同一工厂内的不同车间,产品对象不同,生产批量不同,生产设备、工艺方法和工艺习惯都不一样。这一个工厂适用的工艺规程,不一定适合另一个工厂。不能设想有一个到处通用的CAPP系统。解决这个问题可能有两条道路。一条道路是根据工厂或车间生产情况,总结归纳,构造派生式CAPP系统;另一条道路是构造工艺过程设计自动化的开发环境,建造专家系统外壳。知识库的内容,一部分在系统设计时给出,另一部分由用户工厂填充。把一般性的方法内容和特殊性的要求相结合。这两条道路不是互相排斥冲突的。很可能把它们结合在一起,解决变化多端的实际问题。

第二章 数据结构

§ 2-1 引言

计算机辅助工艺过程设计是典型的复杂问题，它不仅涉及到单纯的数值运算，也要处理图象、字符、表格等较复杂的数据。要处理这些种类的数据，不了解各种数据之间的内在联系和有效的组织是不行的。因此，我们需要讨论一些有关数据和数据结构的基本知识。

所谓数据，从广义角度，它不仅包括数值，而且也泛指可以送入计算机并可被计算机处理的各种字符、数值、其它符号，甚至语音信号等。这些符号和信号，常常称为信息。“数据”和“信息”两者在侧重点上是不相同的，但也常常被当作同义词来使用。

数据类型是指在程序设计语言中各个变量所具有的数据种类。例如，在 *FORT RAN* 语言中，数据类型可为整型、实型、复型、逻辑型和字符型等等。实际上，每种程序设计语言都配有一组内部数据类型，并提供相应的运算处理方法。

数据对象是指某类数据的元素的集合，如整数的数据对象是集合 $D = \{0, \pm 1, \pm 2, \dots\}$ ；英文字符的数据对象是集合 $D = \{'A', 'B', 'C', \dots, 'Z'\}$ 。

数据结构不仅要描述数据对象，而且要描述数据对象各元素之间的相互关系，但它并不涉及数据元素的具体内容。所谓关系，即所描述的数据对象中各元素之间的运算及运算的规则。因此，可以认为数据集合中的数据元素及其相互关系的总称就是数据结构。

数据结构主要研究数据元素之间的逻辑结构（指数据元素实际内容之间的逻辑关系，即面对用户的结构）和物理结构（指数据元素在计算机内的表示，即逻辑结构在计算机内的存储方式），以及这两种结构之间的相互关系。

数据结构类型有线性（如线性列表、数组、串、队列、栈）和非线性（树、图）两大类。有关数据结构的内容还包括排序和查找。本章将选择一些最基本的内容加以介绍。

同时，数据结构的实现是和算法分不开的，而算法又是任何程序设计包括 *CAPP* 系统的基础，所以我们要引入算法的概念。所谓算法（*Algorithm*），就是解决某个特定问题的运算步骤序列。这些步骤也就是在计算机上容易实现的解题的最小运算单位，它应反映解决问题的规则，并且步骤序列应是有穷的。

此外，由于数据库在计算机辅助工艺过程设计中的作用越来越重要，我们在本章的最后对数据库的基本概念也作了一些浅近的介绍。

§ 2-2 线性列表（List）结构

线性列表是最基本的，也是用途最广泛的数据结构形式。其具体形式有如下几种。

一、线性表 (list)

线性表是 $n(n \geq 0)$ 个数据元素的有限序列, 当 $n \neq 0$ 时, 线性表中的每一个元素除第一个及最后一个外, 都有且仅有一个直接前趋, 也都有且仅有一个直接后继。换句话说, 线性表或者是一个空表, 或者可以表达成下述逻辑结构:

$$(a_1, a_2, a_3, \dots, a_i, \dots, a_n)$$

其中, a_i 是某个数据对象的元素。

从上述描述可知线性表的结构特性是:

(1) 线性表是数据元素一个有限序列。线性表的长度定义为线性表中数据元素的个数 n 。当 $n = 0$ 时为空表。

(2) 数据元素在线性表中的位置只取决于它们自己的序号, 数据元素之间的相对位置是线性的。例如, 在上述线性表中, a_1 是第一个数据元素, a_n 是最后一个数据元素; 当 $1 < i < n$ 时, a_i 的前一个数据元素(直接前趋)是 a_{i-1} , 后一个数据元素(直接后继)是 a_{i+1} 。因此, 线性表是一个线性结构。

以上说的是线性表数据元素的逻辑结构。一般说来, 线性表在计算机存储器中的存储形式, 即物理结构也是按顺序分配的原则, 用一组连续的存储单元依次存储各个元素。这种形式也称为“向量”形式。这就意味着数据元素在存储器中的存放地址和该元素的下标之间存在着一一对应的关系。例如, 设有一个线性表 $A(n)$, 假设每一元素占用存储器的长度为 l , 并假设第一个元素在存储器中的起始地址为 $Loc(a_1)$, 则第 i 个元素的起始存储地址为:

$$Loc(a_i) = Loc(a_1) + (i - 1) \times l$$

所以, 线性表元素的存取是比较方便的。

对于线性表的运算, 主要有插入和删除两种, 下面分别讨论。

1. 删除运算

为了保证线性表的线性连续, 在删除若干个元素后必须移动若干个其他元素的位置。例如, 要从长度为 n 的线性表 $A(n)$ 中删除第 i 个元素 a_i , 就要移动从 a_{i+1} 起的各元素的位置。这相当于将 a_i 从线性表中“挤”出来。根据待删除元素位置的不同, 具体算法如下:

- (1) 如 $i > n$ 或 $i < 1$, 则表中无此元素可删除, 给出出错信息后返回;
- (2) 如 $i = n$, 则只需改变表的长度, 即进行 $n = n - 1$ 赋值动作, 然后返回;
- (3) 否则, 依次前移从 a_{i+1} 起的各元素, 即进行 $a_{(j-1)} = a_j$ ($j = i + 1, i + 2, \dots, n$), 并作 $n = n - 1$ 赋值动作, 然后返回。

删除运算示意在图2-1中。

2. 插入运算

如果要在长度为 n 的线性表 $A(n)$ 中的第 i 个元素之前插入一个新元素 x , 就必须把表中从第 i 个元素 a_i 起的各元素都依次往后移动一个位置, 给新元素腾出一个位置, 如图2-2所示。其算法如下:

- (1) 如给定的插入位置 $i > n$, 则作 $n = n + 1$, $a_n = x$, 然后返回;

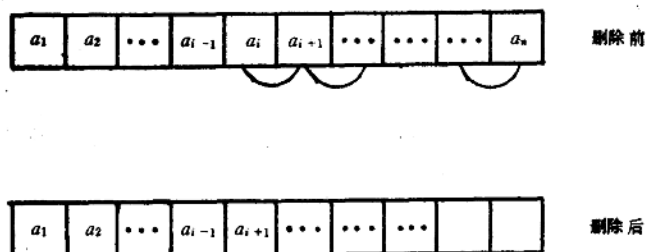


图2-1 线性表元素的删除

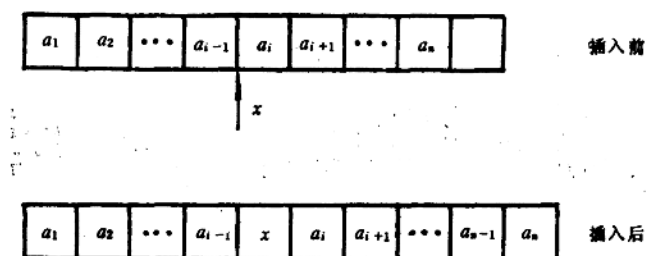


图2-2 线性表元素的插入

(2) 如 $i < 1$, 则作 $a_{(j+1)} = a_j$, ($j = n, n-1, n-2, \dots, 1$), $a_1 = x$, $n = n+1$, 然后返回;

(3) 否则, 作 $a_{(j+1)} = a_j$, ($j = n, n-1, n-2, \dots, i+1, i$), $a_i = x$, $n = n+1$, 然后返回。

从以上介绍的两种运算来看, 一般情况下都需要进行大量元素的移动。如果表的长度比较长, 或者删除、插入运算相当频繁, 就不可避免地要花费很多时间。因此, 线性表比较适用于表中元素不常变动的情况。

二、数组(array)

数组可以看成是线性表定义的扩充。在上述线性表定义中, 如果线性表中的元素是一个个的单个数据, 那么这样的线性表实际上就是一个一维数组。但是, 如果表中的元素本身是一个个包含多个同类型数据的记录, 那么这样的线性表就是二维数组。而如果表中的元素本身是类似二维数组的数据, 那么这样的线性表就是三维数组。类似地, 可以有更多维数的数组。由于数组本身就是线性表, 因此线性表的运算步骤同样适用于数组。

数组的物理结构, 即存储方式也采用顺序分配的原则, 亦即在计算机存储器中需要占用一块连续的存储空间, 依次存放各个数组元素。但是, 具体存储方式有按行、或按列存储两种。所谓按行存储, 就是在存储器中, 数组元素以行为单位, 一行接一行连续存放。反之, 如果以列为单位一列接一列存放, 则称为按列存储。

在FORTRAN语言中, 数组元素是按列存储的。例如, 对于一个二维数组 $A(3,4)$, 它

的全部元素在存储器中的存放形式如图2-3所示。

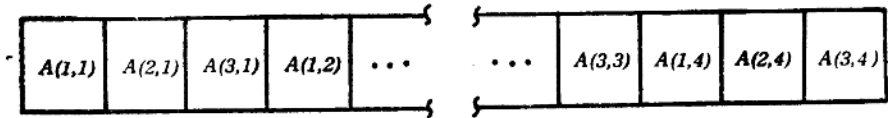


图2-3 按列存储的数组元素

不管是按列还是按行，从数组元素的下标可以很容易地算出它的存储地址，因此要访问任一元素都是比较简单的。例如，设数组 $A(m,n)$ 按行存放，并已知其第一个元素 $A(1,1)$ 的存储地址是 $Loc(1,1)$ 。那么，待访问元素 $A(i,j)$ 的地址就可按下式算出：

$$Loc(i,j) = Loc(1,1) + (j-1) \times m + i - 1$$

式中还假设每一元素只占用一个存储单元。

由此可知，对于数组中任何元素的访问是方便的，且所需时间也是相同的。事实上，在一般高级语言中，访问数组元素时，只需给出它的下标即可，它的存储地址是由编译程序自动给出的。因此，只有一些特殊数组，如存储稀疏矩阵或上（下）三角矩阵的数组，才需要按一定的公式计算存储地址。

三、栈和队列

栈和队列是非常有用的两种线性结构。它们是线性表的特例，即删除和插入位置有限的线性表。

1. 栈(Stack)，也称堆栈

栈在逻辑结构上相当于一端开口，另一端封闭的线性表。对表中元素进行删除和插入运算只能在开口的一端进行，这一端称为栈顶，而另一端，称为栈底，则被看作是封闭的，不允许进行任何运算。栈的逻辑结构形式如图2-4示意。

栈中元素的插入通常称为“进栈”，元素的删除通常称为“出栈”。由于插入和删除只能在同一端进行，所以当前被删除的元素一定是删除运算进行前最后插入的。所以，栈也称为“后进先出”（LIFO: Last In First Out）表。

对栈中元素进行删除和插入比较简单。设有一个栈为 $S(n)$ ，其栈顶指针（指示栈顶位置的一个变量）为 TOP ，则其算法如下：

当要往栈中插入一个新元素 x 时，只需作：

(1) $TOP = TOP + 1$;

(2) $S(TOP) = x$ 。

当要从栈中取出一个元素，并把它值赋给另一变量 x 时，也只需要作两步操作：

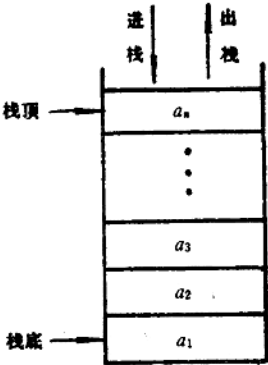


图2-4 栈结构示意图

(1) $x = S(TOP)$;

(2) $TOP = TOP - 1$ 。

上述算法没有考虑“上溢”和“下溢”现象。事实上，在给栈分配了固定的存储单元后，上溢和下溢都可能发生。例如，当栈顶元素已达到所分配存储单元的最大位置，即“栈满”时如还需插入一个有用元素，便是“上溢”；而当我们企图删除一个并不存在，即超出栈底位置的元素时，便发生“下溢”。要想避免上溢和下溢，这对顺序分配的线性表，当能利用的存储空间有限（通常总是如此），而又同时有若干个栈时，原则上是不可能的。这个问题只有按后面要讲到的链式分配存储空间时，才能较好的解决。

在计算机程序设计中，栈的应用很广。最常见的例子是在处理子程序（或过程）的多层次嵌套调用时，总是用附设的栈来控制各层次的返回。

2. 队(Queue)也称队列

队的逻辑结构相当于两端均开口，但插入元素只能从一端进入，而删除元素只能从另一端除去的线性表。允许元素进入的一端称为“排尾”，而允许元素离去的一端称为“排头”。

队的这种结构决定了队中的元素一定是先进队的被首先删除。所以，它和栈相反，是一个“先进先出”(FIFO: First In First Out)表。在对队进行运算时，一般要设两个指针，分别指出当时的排头(Front)和排尾(Rear)。

队的逻辑结构如图2-5示意。

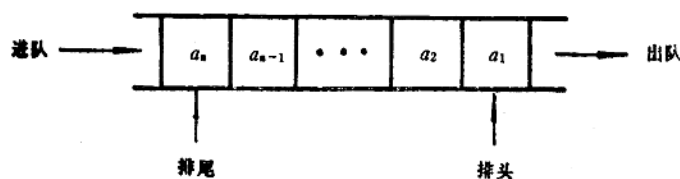


图2-5 队结构示意图

进队和出队的算法也很简单，这里不再赘述。与栈类似，实际的队也有溢出现象。

队的概念，在日常生活中是司空见惯的。在计算机的基本软件中，也是常用的，而在解决工程问题的应用程序中，有时也要用到。

§ 2-3 单向链表(Linked List)

从上一节的介绍可知，线性表结构简单，存储分配容易实现，元素的插入、删除算法也很简单。这些优点使它的各种形式都获得了广泛的应用。但是线性表的顺序分配的存储方式，也给它带来了一些固有的缺点：比如作删除、插入运算时要引起大量数据元素“搬家”，比较费时；表的容量不易扩充；当有多个表时，难以合理分配存储空间，常常要出现不应有的溢出或“假溢出”现象。为了克服这些缺点，就出现了另出一种数据结构形式——链表结构。

链表结构在性质上仍然属于线性表，但是表中的元素已不再是简单的一个数据或是包含

多个相同类型的数据记录，而是由多个数据域组织成的一个统一体，常称为“结点”。结点内所具有的不同数据域中，可以包括不同类型的数据。最简单的链表是单向链表，它的元素，即结点包含两个域：一个是数据域，用来存放该元素的数据值；另一个是链域，用来存放指示它的后继结点的存放地址的指针，如图2-6所示。这就是说，链表结构上的连续性不需要依靠分配连续的存储单元来维持，而是依靠每个结点的链接指针来保证。这样，我们就可以利用任意位置的存储单元来存放链表中的各个数据元素，使得它们的物理结构相当灵活。这是它的第一个优点。

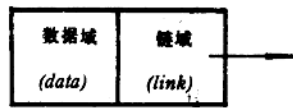


图2-6 链表的结点

在叙述链表结构时，我们以符号 $n \cdot link$ 表示结点 n 的链域值（指针），以符号 $n \cdot data$ 表示结点 n 的数据域的值。本节，我们先介绍单向链表，其结构如图2-7示意。

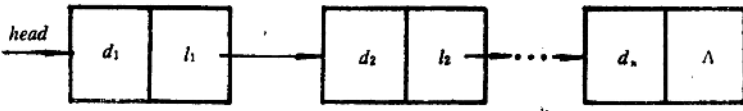


图2-7 单向链表结构

从图中可以看出，单向链表在逻辑上是由一组有序结点组成的，但在物理结构上却可以是无序的，并不需要为链表分配一组连续的存储单元。链表的每一结点，都通过指针（链）域指出了该结点的直接后继。在图2-7中还示出了表头指针head，它指出链表中第一个结点的存储地址。第 n 个结点指针域中的“ Λ ”表示是空指针，即说明该结点是最后结点，其后面已经没有其他后继结点了。

链表中元素的删除及插入运算，不需要移动元素（结点）的位置，因而可大大节约机时，这是它的又一个优点。下面分别介绍。

一、删除运算

从链表 $E(n)$ 中删除一个结点 E_i ，只需改变它的前一个结点 $E_{(i-1)}$ 的链域的指针值，使之由原来指向结点 E_i ，改为指向 E_i 的直接后继结点 $E_{(i+1)}$ 就可以了，如图2-8所示。另

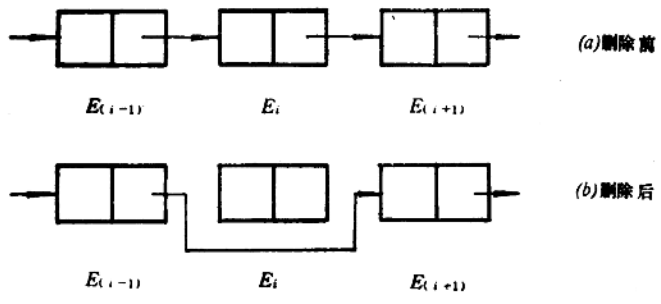


图2-8 从单向链表中删除一个结点