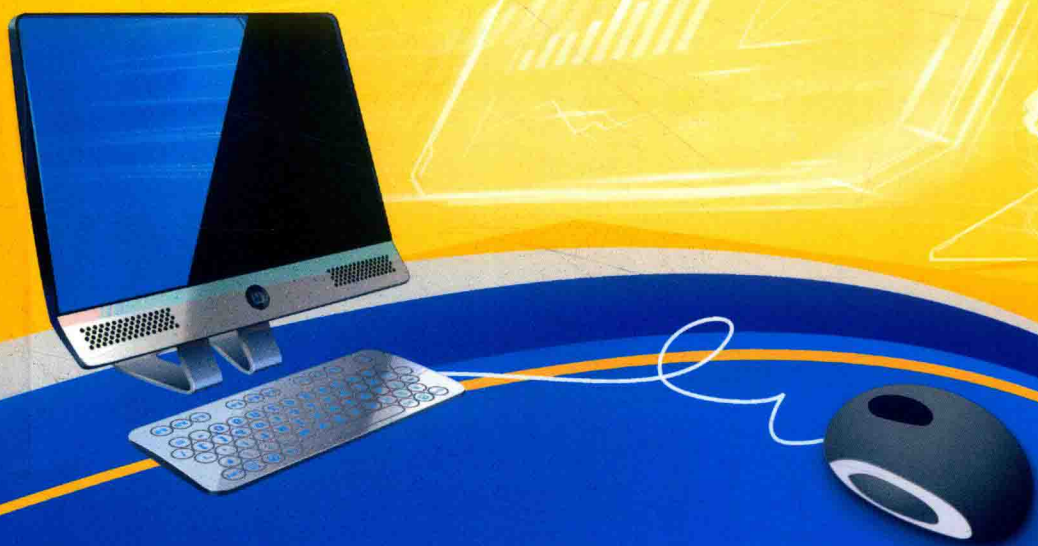


LISAN SHUXUE ZHONG DE
SUANFA SHEJI YU SHIXIAN

离散数学中的 算法设计与实现

陈业纲 / 编著



非外借

离散数学中的算法设计与实现

编 著 陈业纲

编 委 潘小琴 安建梅

西南交通大学出版社

· 成 都 ·

图书在版编目 (CIP) 数据

离散数学中的算法设计与实现 / 陈业纲编著. —成都: 西南交通大学出版社, 2018.11
ISBN 978-7-5643-6507-3

I. ①离… II. ①陈… III. ①离散数学—算法设计
IV. ①O158

中国版本图书馆 CIP 数据核字 (2018) 第 247474 号

离散数学中的算法设计与实现

陈业纲 / 编 著

责任编辑 / 穆 丰
助理编辑 / 何明飞
封面设计 / 墨创文化

西南交通大学出版社出版发行
(四川省成都市二环路北一段 111 号西南交通大学创新大厦 21 楼 610031)
发行部电话: 028-87600564 028-87600533
网址: <http://www.xnjdcbs.com>
印刷: 成都中永印务有限责任公司

成品尺寸 185 mm × 260 mm
印张 12.25 字数 260 千
版次 2018 年 11 月第 1 版 印次 2018 年 11 月第 1 次

书号 ISBN 978-7-5643-6507-3
定价 35.00 元

课件咨询电话: 028-87600533
图书如有印装质量问题 本社负责退换
版权所有 盗版必究 举报电话: 028-87600562



前 言

离散数学是计算机科学与技术专业一门理论性较强的基础理论课,也是该专业的核心课程和主干课程。目前,国内离散数学教学主要是让学生掌握书中的概念、定理、公式并能灵活地解题,而对实验教学课程进行的研究较少,忽视了实践课程对理论课程的辅助和促进作用,也忽视了该课程与数据结构等课程的有机联系。但是,对于计算机专业的学生而言,学习离散数学更重要的是实践。为了改善这种局面,帮助学生更好地理解和掌握所学的基本概念和方法,并能运用所学知识,上机解决一些典型问题,在教学中应加强离散数学的实践教学,利用小组合作学习或者组织全班讨论,开展研究性学习活动。在实验过程中,依靠计算机,让学生主动参与探索、解决问题,使他们从中获得用离散数学研究、解决实际问题的过程体验、情感体验,产生成就感,进而提高学习的兴趣、实际动手的能力并激发创新潜能。

本书在编写中力求做到:理论与算法并重;去掉冗长的证明,保留简洁、有特色、能体现典型的数学思想和方法的内容;强调算法的基本思想和计算机实现。全书共分7部分:数理逻辑、图、关系、函数与集合、树、代数系统、递推方程的求解。书中所有代码均在VC++环境下调试通过。

本书由陈业纲和潘小琴、安建梅三位老师编写,其中第1~5章由陈业纲编写;第6章由潘小琴编写;第7章由安建梅编写。本书在编写的过程中,参考了部分院校ACM竞赛相关的资料和文献,兄弟院校的同行也提出了宝贵的意见和建议,在这里一并表示感谢。同时,还要感谢李柳柏教授项目的无私帮助。

由于编者水平有限,书中难免存在不足和疏漏之处,恳请广大读者批评指正。

编 者

2018年6月

目 录

第 1 章	数理逻辑	1
1.1	真值表、主析取范式 and 主合取范式	1
1.2	推理	11
第 2 章	图	16
2.1	图的基本概念	17
2.2	二分图、欧拉图和哈密顿图	24
2.3	最短路径、关键路径	39
2.4	广度优先、深度优先算法	55
2.5	桥、割点	60
第 3 章	关 系	66
3.1	关系的性质	66
3.2	偏序集极小极大元最小最大元	75
3.3	矩阵与关系闭包	81
3.4	布尔矩阵交并积	86
3.5	关系的传递闭包	90
3.6	最小等价关系	97
第 4 章	函数与集合	103
4.1	单满射一一映射	103
4.2	集合的运算	110
4.3	并查集	112
4.4	排列组合	115
4.5	商 集	124
4.6	格雷码	133
4.7	数字拆解	135
第 5 章	树	139
5.1	表达式运算	139
5.2	树的判定和遍历	144

5.3	最小生成树	155
5.4	哈夫曼树	165
第 6 章	代数系统	167
6.1	群的判断	167
6.2	格	169
第 7 章	递推方程的求解	174
7.1	递 归	174
7.2	递推数列	176
参考文献		189

第 1 章 数理逻辑

1.1 真值表、主析取范式 and 主合取范式

1.1.1 基本概念

1. 真值表

真值表是在逻辑中使用的一类数学表，用来确定一个表达式是否为真或有效。它表征逻辑事件输入和输出之间全部可能状态，列出命题公式真假值。通常以 1 表示真，0 表示假。命题公式的取值由组成命题公式的命题变元的取值和命题联结词决定，命题联结词的真值表给出了真假值的算法。

2. 范 式

就是命题公式形式的规范形式。这里约定在范式中（只含有联结词 $\neg$ 、 $\vee$ 和 $\wedge$ ）。

主析取范式：在含有 n 个命题变元的简单合取式中，若每个命题变元与其否定不同时存在，而两者之一出现一次且仅出现一次，称该简单合取式为小项。由若干个不同的小项组成的析取式称为主析取范式；与 A 等价的主析取范式称为 A 的主析取范式。任意含 n 个命题变元的非永假命题公式 A 都存在与其等价的主析取范式，并且是唯一的。

主合取范式：在含有 n 个命题变元的简单析取式中，若每个命题变元与其否定不同时存在，而两者之一出现一次且仅出现一次，称该简单析取式为大项。由若干个不同的大项组成的合取式称为主合取范式；与 A 等价的主合取范式称为 A 的主合取范式。任意含 n 个命题变元的非永真命题公式 A 都存在与其等价的主合取范式，并且是唯一的。

1.1.2 程序实例

以下程序实现了如下功能：

- (1) 输出真值表。
- (2) 输出主析取范式和主合取范式。



代码:

```
#include "stdio.h"
#include "stdlib.h"
#include "string.h"
#include "conio.h"
#include "math.h"
#define N 50
void panduan(int b[N],int f);//赋值函数
int tkh (char sz[N], char ccu[N], int icu[N], int h0);//分级运算函数
int fkh (char sz[N], char ccu[N], int icu[N], int h0);//主运算函数
void display()
{
#include "stdio.h"
#include "stdlib.h"
#include "string.h"
#include "conio.h"
#include "math.h"
#define N 50
void panduan(int b[N],int f);//赋值函数
int tkh (char sz[N], char ccu[N], int icu[N], int h0);//分级运算函数
int fkh (char sz[N], char ccu[N], int icu[N], int h0);//主运算函数
void display()
{
printf("*****\n");//标语
printf("***                               **\n");
printf("***           欢迎进行范式运算           **\n");
printf("***           (真值表、主范式、括号)           **\n");
printf("***                               **\n");
printf("***           用!表示非                               **\n");
printf("***           用&表示与                               **\n");
printf("***           用|表示或                               **\n");
printf("***           用^表示蕴含                               **\n");
printf("***           用~表示等值                               **\n");
printf("***                               **\n");
printf("*****\n\n");
}
```



```

main()
{
    int i1,i2,d=1,icu[N],kh=0,jg,j=0,h0;//icu[N]用于存放变量值,kh 括号计数,jg 存放
结果
    int bj=0,hq[N],h=0,x=0,xq[N];//hq[N]存放合取结果 xq[N]存放析取结果
    char sz[N],ccu[N],sz0[N],s;//sz[N]存放式子,ccu[N]存放变量,sz0[N]也是用于存
放式子
    hq[0]=-1;
    xq[0]=-1;
    display();
    printf("请输入一个合法的命题公式:\n");//输入式子
    gets(sz);//读取式子
    strcpy(sz0,sz);//复制式子
    for(i1=0;i1<strlen(sz);i1++)
    {
        if(sz[i1]=='(' || sz[i1]==')')//存储括号数量
            kh++;
        if(sz[i1]>='a' && sz[i1]<='z' || sz[i1]>='A' && sz[i1]<='Z')
        {
            for(i2=0;i2<j;i2++) //判断并储存变量
                if(ccu[i2]==sz[i1])//去除重复变量
                    d=0;
            if(d==1)
            {
                ccu[j]=sz[i1];
                j++;
            }
            d=1;
        }
    }
    printf("\nd 该式子中的变量个数为: %d\n",j);//输出变量个数
    h0=j;
    printf("\n 输出真值表如下: \n \n");//输出真值表表头
    for(i1=0;i1<h0;i1++)
        printf(" %c ",ccu[i1]);
    printf(" ");

```



```
puts(sz);
printf("\n");
for(i1=0;i1<j;i1++) //先将所有的变量赋值为零。
    icu[i1]=0;
for(i2=0;i2<j;i2++)//输出真值表前项
    printf(" %d ",icu[i2]);
jg=tkh(sz,ccu,icu,h0); //用函数求结果
if(jg==0)//结果为 0，合取加 1
    hq[h++]=bj;
else //否则，析取加 1
    xq[x++]=bj;
printf("      %d\n",jg);//输出运算结果
strcpy(sz,sz0);
for(i1=0;i1<(int)pow(2,j)-1;i1++)
{
    ++bj;
    panduan(icu,j-1); //赋值变量
    jg=tkh(sz,ccu,icu,h0);
    if(jg==0)//结果为 0，合取加 1
        hq[h++]=bj;
    else //否则，析取加 1
        xq[x++]=bj;
    strcpy(sz,sz0); //恢复被修改的数组
    for(i2=0;i2<j;i2++)
        printf(" %d ",icu[i2]); //输出真值表前项
    printf("      %d\n",jg); //输出运算结果
}
if(hq[0]==-1)//不存在合取范式时
    printf("\n 该命题公式不存在主合取范式。 \n");
else
{
    printf("\n 该命题公式的主合取范式: \n\t");
    for(i1=0;i1<h;i1++)
    {
        if (i1>0)//判断并添加符号
            printf("\n\t");
```

```
        printf("M(%d)",(int)pow(2,j)-1-hq[i1]); //输出主合取范式
    }
}
if(xq[0]==-1)//不存在析取范式时
    printf("\n 该命题公式不存在主析取范式。 \n");
else
{
    printf("\n\n 该命题公式的主析取范式： \n\t");
    for(i1=0;i1<x;i1++)
    {
        if (i1>0)//判断并添加符号
            printf("\\\\");
        printf("m(%d)",xq[i1]); //输出主析取范式
    }
}
printf("\n");
printf("\n 欢迎下次再次使用！ \n "); //结束
getch();
}
```

```
void panduan(int b[N],int f) // 二进制赋值
{
    int i;
    i=f;
    if(b[f]==0)//加 1
        b[f]=1;
    else//进位
    {
        b[f]=0;
        panduan(b,--i);
    }
}
```

```
int tkh (char sz[N],char ccu[N],int icu[N],int h0)//分级运算函数
{
    int i,j,h,s,kh=0,wz[N],a;
```



```
char xs1[N],ckh[N]; //xs1 用来保存括号内的字符 ckh 用来保存括号。
s=strlen(sz);
for(i=0;i<s;i++)
if(sz[i]=='(' || sz[i]==')')//判断括号
{
    wz[kh]=i;//存储括号位置
    ckh[kh]=sz[i];//存储括号类型
    kh++;
}
if(kh==0)
    return fkh(sz,ccu,icu,h0);//如果无括号，直接运行
else
{
    for(i=0;i<kh;i++)
    if(ckh[i]==')')//找到第一个 )
        break;
    for(j=wz[i-1]+1,h=0;j<wz[i];j++,h++) //存储最内级括号中的内容
        xs1[h]=sz[j];
    xs1[h]='\0';
    a=fkh(xs1,ccu,icu,h0);//运行最内级括号的式子，得到结果
    if(a==1)//判断并存储结果
        sz[wz[i-1]]=1;
    else
        sz[wz[i-1]]=-2;
    for(j=wz[i-1]+1;j<s+wz[i]-wz[i];j++)//将括号后内容前移
        sz[j]=sz[j+wz[i]-wz[i-1]];
    sz[j]='\0';
    return tkh(sz,ccu,icu,h0);//循环执行
}
}

int fkh(char sz[N],char ccu[N],int icu[N],int h0)//主运算函数
{
    int i,h=0,j=0,j1=0,j2=0,j3=0,j4=0,j5=0,i1,i2,p1=-1,p2=-1,s;
    char dt[N];
    s=strlen(sz);
```



```

if(s==1)
if(sz[0]==-2)//判断是否是最后一项
    return 0;
else
    return 1; //1 就是 sz[0]的值
else
{
    for(i=0;i<s-j;i++) //先处理非
    if(sz[i]=='!')
    {
        for(i1=0;i1<h0;i1++)
        if(sz[i+1]==ccu[i1])//将变量赋值并给 p1
            p1=icu[i1];
        if(sz[i+1]==-2)//如果是前运算结果的 0, 则 p1 等于 0
            p1=0;
        if(p1==-1)//如果是数字, 直接给 p1
            p1=sz[i+1];
        dt[j+2]=!p1;//非运算
        sz[i]=j+2;
        j++;
        p1=0;
        for(i1=i+1;i1<s-j;i1++)
            sz[i1]=sz[i1+1];//将后续式子前移一项
    }
    p1=-1;
    j1=j;
    for(i=0;i<s-j1-2*j2;i++) // 处理与
    if(sz[i]=='&')
    {
        for(i1=0;i1<h0;i1++)
        {
            if(sz[i-1]==ccu[i1])//将变量赋值并给 p1
                p1=icu[i1];
            if(sz[i+1]==ccu[i1])//将变量赋值并给 p2
                p2=icu[i1];
        }
    }
}

```



```
for(i2=2;i2<j+2;i2++)
{
    if(sz[i-1]==i2) //如果为前计算结果，将结果赋值并给 p1
        p1=dt[i2];
    if(sz[i+1]==i2) //如果为前计算结果，将结果赋值并给 p2
        p2=dt[i2];
}
if(sz[i-1]==-2)//如果是前运算结果的 0，则 p1 等于 0
    p1=0;
if(sz[i+1]==-2)//如果是前运算结果的 0，则 p2 等于 0
    p2=0;
if(p1==-1) //如果是数字，直接给 p1
    p1=(int)(sz[i-1]);
if(p2==-1)//如果是数字，直接给 p2
    p2=(int)(sz[i+1]);
dt[j+2]=p1 && p2;//与运算
sz[i-1]=j+2;
j++;
j2++;
p1=-1;
p2=-1;
for(i1=i;i1<s-j1-2*j2;i1++)//将后续式子前移两项
    sz[i1]=sz[i1+2];
i=i-1;
}
for(i=0;i<s-j1-2*j2-2*j3;i++) // 处理或
    if(sz[i]=='|')
    {
        for(i1=0;i1<h0;i1++)
        {
            if(sz[i-1]==ccu[i1])//将变量赋值并给 p1
                p1=icu[i1];
            if(sz[i+1]==ccu[i1])//将变量赋值并给 p2
                p2=icu[i1];
        }
        for(i2=2;i2<j+2;i2++)
```



```

{
    if(sz[i-1]==i2) //如果为前计算结果，将结果赋值并给 p1
        p1=dt[i2];
    if(sz[i+1]==i2)//如果为前计算结果，将结果赋值并给 p2
        p2=dt[i2];
}
if(sz[i-1]==-2)//如果是前运算结果的 0，则 p1 等于 0
    p1=0;
if(sz[i+1]==-2)//如果是前运算结果的 0，则 p2 等于 0
    p2=0;
if(p1==-1)//如果是数字，直接给 p1
    p1=sz[i-1];
if(p2==-1)//如果是数字，直接给 p2
    p2=sz[i+1];
dt[j+2]=p1 || p2;//或运算
sz[i-1]=j+2;
j++;
j3++;
p1=-1;
p2=-1;
for(i1=i;i1<s-j1-2*j2-2*j3;i1++)//将后续式子前移两项
    sz[i1]=sz[i1+2];
i--;
}
for(i=0;i<s-j1-2*j2-2*j3-2*j4;i++) // 处理蕴含
if(sz[i]=='^')
{
    for(i1=0;i1<h0;i1++)
    {
        if(sz[i-1]==ccu[i1])//将变量赋值并给 p1
            p1=icu[i1];
        if(sz[i+1]==ccu[i1])//将变量赋值并给 p2
            p2=icu[i1];
    }
    for(i2=2;i2<j+2;i2++)
    {

```



```
    if(sz[i-1]==i2) //如果为前计算结果，将结果赋值并给 p1
        p1=dt[i2];
    if(sz[i+1]==i2) //如果为前计算结果，将结果赋值并给 p2
        p2=dt[i2];
}
if(sz[i-1]==-2)//如果是前运算结果的 0，则 p1 等于 0
    p1=0;
if(sz[i+1]==-2)//如果是前运算结果的 0，则 p2 等于 0
    p2=0;
if(p1==-1)//如果是数字，直接给 p1
    p1=sz[i-1];
if(p2==-1)//如果是数字，直接给 p2
    p2=sz[i+1];
dt[j+2]=!p1 || p2;//蕴含运算
sz[i-1]=j+2;
j++;
j4++;
p1=-1;
p2=-1;
for(i1=i;i1<s-j1-2*j2-2*j3-2*j4;i1++)//将后续式子前移两项
    sz[i1]=sz[i1+2];
i--;
}
for(i=0;i<s-j1-2*j2-2*j3-2*j4-2*j5;i++) // 处理等值
if(sz[i]=='~')
{
    for(i1=0;i1<h0;i1++)
    {
        if(sz[i-1]==ccu[i1])//将变量赋值并给 p1
            p1=icu[i1];
        if(sz[i+1]==ccu[i1])//将变量赋值并给 p2
            p2=icu[i1];
    }
    for(i2=2;i2<j+2;i2++)
    {
        if(sz[i-1]==i2) //如果为前计算结果，将结果赋值并给 p1
```



```

        p1=dt[i2];
        if(sz[i+1]==i2) //如果为前计算结果，将结果赋值并给 p2
            p2=dt[i2];
    }
    if(sz[i-1]==-2) //如果是前运算结果的 0，则 p1 等于 0
        p1=0;
    if(sz[i+1]==-2) //如果是前运算结果的 0，则 p2 等于 0
        p2=0;
    if(p1==-1) //如果是数字，直接给 p1
        p1=sz[i-1];
    if(p2==-1) //如果是数字，直接给 p2
        p2=sz[i+1];
    dt[j+2]=(!p1 || p2)&&(!p2 || p1); //等值运算
    sz[i-1]=j+2;
    j++;
    j5++;
    p1=-1;
    p2=-1;
    for(il=i;il<s-j1-2*j2-2*j3-2*j4-2*j5;il++) //将后续式子前移两项
        sz[il]=sz[il+2];
    i--;
}
return dt[j+1]; //返回结果
}
}

```

1.2 推理

1.2.1 基本概念

推理就是根据一个或几个已知的判断得出一个新的判断的思维过程。这些已知的判断被称为前提，得到的新的判断为前提的有效结论。

实际上，推理的过程就是证明永真蕴含式的过程，即令 $H_1, H_2, \dots, H_n$ 是已知的命题公式（前提），若有