

# 《编译原理教程(第四版)》 习题解析与上机指导

◎主 编 胡元义  
◎副主编 孙钦东 邓亚玲 谈姝辰

BIANYIYUANLIJIAOCHENGDISIBAN  
XITIJEXIYUSHANGJI ZHIDAO



高等学校计算机类“十三五”规划教材

# 《编译原理教程(第四版)》

## 习题解析与上机指导

主 编 胡元义

副主编 孙钦东 邓亚玲 谈姝辰

西安电子科技大学出版社



## 内 容 简 介

本书是与《编译原理教程(第四版)》(胡元义主编,西安电子科技大学出版社)一书配套的习题解析与上机指导教材,也可单独使用。本书的习题解析部分对《编译原理教程(第四版)》中的习题进行了深入、细致的分析和解答,为读者熟练掌握编译原理知识,抓住重点,突破难点提供了有益的帮助。本书的上机指导部分包括由高级程序语言到中间语言,由汇编语言到机器语言的翻译,使编译的主要翻译阶段和环节都能微观且实时地显示出来,较好地解决了编译原理的理论与实践的衔接问题。此外,我们结合自己开发的8086/8088小汇编指令到机器代码的翻译成果,将8086/8088汇编指令到机器代码的翻译方法引入到书中,有利于读者了解低级语言的翻译过程和实现方法。

### 图书在版编目(CIP)数据

《编译原理教程(第四版)》习题解析与上机指导/胡元义主编.—4版.

—西安:西安电子科技大学出版社,2017.2

高等学校计算机类“十三五”规划教材

ISBN 978-7-5606-3899-7

I. ①编… II. ①胡… III. ①编译程序—程序设计—高等学校—教学参考资料 IV. ①TP314

中国版本图书馆CIP数据核字(2015)第321756号

策 划 胡华霖 马乐惠

责任编辑 张晓燕

出版发行 西安电子科技大学出版社(西安市太白南路2号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfxb001@163.com

经 销 新华书店

印刷单位 陕西利达印务有限责任公司

版 次 2017年2月第4版 2017年2月第5次印刷

开 本 787毫米×1092毫米 1/16 印张 13.25

字 数 312千字

印 数 15 001~18 000册

定 价 23.00元

ISBN 979-7-5606-3899-7/TP

**XDUP 4191004-5**

\*\*\* 如有印装问题可调换 \*\*\*

本社图书封面为激光防伪覆膜,谨防盗版。

# 前 言

编译原理是计算机专业的一门核心课程，在计算机本科教学中占有十分重要的地位。由于编译原理课程具有很强的理论性和实践性，因而学生在学习时普遍感到内容抽象，不易理解，掌握起来难度较大。本书通过习题解析的方式来帮助读者理解编译技术的原理和概念，掌握编译原理的相关方法，提高分析问题与解决问题的能力。本书的上机指导部分则给读者提供了一个完整的小型编译程序，以便读者上机实践，较好地解决了编译原理与实践的衔接问题，使读者对编译原理有一个形象、直观和透彻的认识和感受，能够更深入地了解 and 掌握编译原理的内容和实现方法。

本书是与编者在西安电子科技大学出版社出版的《编译原理教程(第四版)》一书相配套的习题解析与上机指导教材，它也可以与目前各种编译原理教材配套使用。本书分为两篇，第一篇为编译原理习题解析部分。为了便于读者正确理解编译原理的概念，掌握解题方法，本篇对《编译原理教程(第四版)》一书中各章的习题都给出了详尽的解题过程，对书中引用到的概念、原理和公式给出了出处；对有代表性的习题和疑难习题也给出了详细的分析和解答。此外，对某些习题，本书还给出了一些新的解题思路和方法。本书的第二篇是编译原理上机指导部分，给出了一个完整的小型编译程序，该程序涵盖了编译原理的词法分析、语法分析、中间代码生成等各阶段的内容。此外，还给出了 8086/8088 汇编语言到机器语言的翻译程序。本书中的小型编译程序可接受本书中文法规定的高级语言程序，将其翻译成四元式代码形式的中间语言程序，并且使编译的主要翻译阶段和环节都能微观且实时地显示出来，有利于读者深入了解编译的内部过程和实现细节，并为读者开拓了进一步学习和运用编译原理的视野。对目标代码生成，国内的编译教材只是笼统地介绍了从中间代码到假想机汇编这一级的翻译，对于究竟计算机是如何实现将汇编语言翻译成可执行的机器代码却均无介绍。我们结合自己开发的 8086/8088 小汇编指令到机器代码的翻译成果，将 8086/8088 汇编指令到机器代码的翻译方法引入到书中，有利于读者了解低级语言的翻译过程和实现方法。

本书在第三版基础上重点对各章第一题的单项选择题进行了补充、完善和修改，对第三章的习题进行了添加和补充。

由于编者水平所限，本书难免存在差错和不足，敬请广大读者批评指正。

编者

2015年5月

# 目 录

## 第一篇 编译原理习题解析

第一章	绪论.....	1
第二章	词法分析.....	3
第三章	语法分析.....	11
第四章	语义分析和中间代码生成.....	54
第五章	代码优化.....	62
第六章	运行时存储空间组织.....	70
第七章	目标代码生成.....	76
第八章	符号表与错误处理.....	81

## 第二篇 编译原理上机指导

第九章	小型编译程序介绍.....	85
9.1	小型编译程序结构.....	85
9.2	小型编译程序关于高级语言的规定.....	85
9.3	小型编译程序关于单词的内部定义.....	87
9.4	小型编译程序的 LR 分析表.....	88
9.5	小型编译程序运行实例分析.....	90
第十章	上机实验内容.....	95
10.1	实验一 编译程序的分析与验证.....	95
10.2	实验二 算术表达式的扩充.....	95
10.3	实验三 添加新的程序语句(一).....	96
10.4	实验四 添加新的程序语句(二).....	97
10.5	编译原理课程设计.....	97
第十一章	高级语言到四元式的编译程序.....	99
第十二章	8086/8088 小汇编的设计与实现.....	124
12.1	汇编指令系统的分析.....	124
12.2	8086/8088 小汇编的设计实现.....	131
12.3	8086/8088 小汇编实验.....	139
12.4	8086/8088 小汇编程序.....	141
附录 1	8086/8088 指令码汇总表.....	193
附录 2	8086/8088 指令编码空间表.....	198
参考文献	.....	200

# 第一篇 编译原理习题解析

## 第一章 绪 论

1.1 完成下列选择题:

- (1) 下面叙述中正确的是\_\_\_\_\_。
- A. 编译程序是将高级语言程序翻译成等价的机器语言程序的程序
  - B. 机器语言因其使用过于困难, 所以现在计算机根本不使用机器语言
  - C. 汇编语言是计算机唯一能够直接识别并接受的语言
  - D. 高级语言接近人们的自然语言, 但其依赖具体机器的特性是无法改变的
- (2) 将编译过程分成若干“遍”是为了\_\_\_\_\_。
- A. 提高编译程序的执行效率
  - B. 使编译程序的结构更加清晰
  - C. 利用有限的机器内存并提高机器的执行效率
  - D. 利用有限的机器内存但降低了机器的执行效率
- (3) 构造编译程序应掌握\_\_\_\_\_。
- A. 源程序
  - B. 目标语言
  - C. 编译方法
  - D. A~C 项
- (4) 编译程序绝大多数时间花在\_\_\_\_\_上。
- A. 出错处理
  - B. 词法分析
  - B. 目标代码生成
  - D. 表格管理
- (5) 编译程序是对\_\_\_\_\_。
- A. 汇编程序的翻译
  - B. 高级语言程序的解释执行
  - C. 机器语言的执行
  - D. 高级语言的翻译

### 【解答】

- (1) 编译程序可以将用高级语言编写的源程序转换成与之在逻辑上等价的的目标程序, 而目标程序可以是汇编语言程序或机器语言程序。故选 A。
- (2) 分多遍完成编译过程可使整个编译程序的逻辑结构更加清晰。故选 B。
- (3) 构造编译程序应掌握源程序、目标语言和编译方法这三方面内容。故选 D。
- (4) 编译各阶段的工作都涉及到构造、查找或更新有关表格, 即编译过程的绝大部分时间都用在造表、查表和更新表格的事务上。故选 D。
- (5) 由(1)可知, 编译程序实际上实现了对高级语言程序的翻译。故选 D。

1.2 计算机执行用高级语言编写的程序有哪些途径？它们之间的主要区别是什么？

**【解答】** 计算机执行用高级语言编写的程序主要有两种途径：解释和编译。

在解释方式下，翻译程序事先并不采用将高级语言程序全部翻译成机器代码程序，然后执行这个机器代码程序的方法，而是每读入一条源程序的语句，就将其解释(翻译)成对应其功能的机器代码语句串并执行，然后再读入下一条源程序语句并解释执行，而所翻译的机器代码语句串在该语句执行后并不保留。这种方法是按源程序中语句的动态执行顺序逐句解释(翻译)执行的，如果一语句处于一循环体中，则每次循环执行到该语句时，都要将其翻译成机器代码后再执行。

在编译方式下，高级语言程序的执行是分两步进行的：第一步首先将高级语言程序全部翻译成机器代码程序，第二步才是执行这个机器代码程序。因此，编译对源程序的处理是先翻译，后执行。

从执行速度上看，编译型的高级语言比解释型的高级语言要快，但解释方式下的人机界面比编译型好，便于程序调试。

这两种途径的主要区别在于：解释方式下不生成目标代码程序，而编译方式下生成目标代码程序。

1.3 请画出编译程序的总框图。如果你是一个编译程序的总设计师，设计编译程序时应当考虑哪些问题？

**【解答】** 编译程序总框图如图 1-1 所示。



图 1-1 编译程序总框图

作为一个编译程序的总设计师，首先要深刻理解被编译的源语言其语法及语义；其次，要充分掌握目标指令的功能及特点，如果目标语言是机器指令，还要搞清楚机器的硬件结构以及操作系统的功能；第三，对编译的方法及使用的软件工具也必须准确化。总之，总设计师在设计编译程序时必须估量系统功能要求、硬件设备及软件工具等诸因素对编译程序构造的影响。

## 第二章 词法分析

2.1 完成下列选择题:

- (1) 词法分析所依据的是\_\_\_\_\_。
 

A. 语义规则	B. 构词规则
C. 语法规则	D. 等价变换规则
- (2) 词法分析器的输入是\_\_\_\_\_。
 

A. 单词符号串	B. 源程序
C. 语法单位	D. 目标程序
- (3) 词法分析器的输出是\_\_\_\_\_。
 

A. 单词的种别编码	B. 单词的种别编码和自身的值
C. 单词在符号表中的位置	D. 单词自身值
- (4) 状态转换图(见图 2-1)接受的字集为 \_\_\_\_\_。
 

A. 以 0 开头的二进制数组成的集合	B. 以 0 结尾的二进制数组成的集合
C. 含奇数个 0 的二进制数组成的集合	D. 含偶数个 0 的二进制数组成的集合

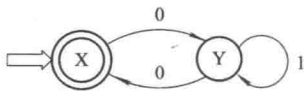


图 2-1 习题 2.1 的 DFA M

- (5) 对于任一给定的 NFA M, \_\_\_\_\_ 一个 DFA M', 使  $L(M) = L(M')$ .
 

A. 一定不存在	B. 一定存在	C. 可能存在	D. 可能不存在
----------	---------	---------	----------
- (6) DFA 适用于\_\_\_\_\_。
 

A. 定理证明	B. 语法分析	C. 词法分析	D. 语义加工
---------	---------	---------	---------
- (7) 下面用正规表达式描述词法的论述中, 不正确的是\_\_\_\_\_。
 

A. 词法规则简单, 采用正规表达式已足以描述	B. 正规表达式的表示比上下文无关文法更加简洁、直观和易于理解	C. 正规表达式描述能力强于上下文无关文法	D. 有限自动机的构造比下推自动机简单且分析效率高
-------------------------	---------------------------------	-----------------------	---------------------------
- (8) 与  $(alb)^*(alb)$  等价的正规式是\_\_\_\_\_。
 

A. $(alb)(alb)^*$	B. $a^*lb^*$	C. $(ab)^*(alb)^*$	D. $(alb)^*$
-------------------	--------------	--------------------	--------------
- (9) 在状态转换图的实现中, \_\_\_\_\_ 一般对应一个循环语句。
 

A. 不含回路的分叉结点	B. 含回路的状态结点
C. 终态结点	D. A~C 都不是
- (10) 已知 DFA  $M_d = (\{s_0, s_1, s_2\}, \{a, b\}, f, s_0, \{s_2\})$ , 且有:



$$f(s_0, a) = s_1 \qquad f(s_1, a) = s_2$$

$$f(s_2, a) = s_2 \qquad f(s_2, b) = s_2$$

则该 DFA M 所能接受的语言可以用正规表达式表示为\_\_\_\_\_。

- A.  $(a|b)^*$     B.  $aa(a|b)^*$     C.  $(a|b)^*aa$     D.  $a(a|b)^*a$

**【解答】**

- (1) 由教材第一章 1.3 节中的词法分析, 可知词法分析所遵循的是语言的构词规则。故选 B。
- (2) 词法分析器的功能是输入源程序, 输出单词符号。故选 B。
- (3) 词法分析器输出的单词符号通常表示为二元式: (单词种别, 单词自身的值)。故选 B。
- (4) 虽然选项 A、B、D 都满足题意, 但选项 D 更准确。故选 D。
- (5) NFA 可以有 DFA 与之等价, 即两者描述能力相同; 也即, 对于任一给定的 NFA M, 一定存在一个 DFA M', 使  $L(M)=L(M')$ 。故选 B。
- (6) DFA 便于识别, 易于计算机实现, 而 NFA 便于定理的证明。故选 C。
- (7) 本题虽然是第二章的题, 但答案参见第三章 3.1.3 节。即选 C。
- (8) 由于正则闭包  $R^+ = R^*R = RR^*$ , 故  $(alb)^*(alb) = (alb)(alb)^*$ 。故选 A。
- (9) 含回路的状态结点一般对应一个循环语句。故选 B。
- (10) DFA  $M_d$  所对应的 DFA 如图 2-2 所示。故选 B。

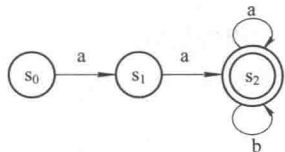


图 2-2 DFA M

**2.2 什么是扫描器? 扫描器的功能是什么?**

**【解答】** 扫描器就是词法分析器, 它接受输入的源程序, 对源程序进行词法分析并识别出一个个单词符号, 其输出结果是单词符号, 供语法分析器使用。通常把词法分析器作为一个子程序, 每当语法分析器需要一个单词符号时就调用这个子程序。每次调用时, 词法分析器就从输入串中识别出一个单词符号交给语法分析器。

**2.3 设  $M = (\{x, y\}, \{a, b\}, f, x, \{y\})$  为一非确定的有限自动机, 其中  $f$  定义如下:**

$$\begin{aligned} f(x, a) &= \{x, y\} & f(x, b) &= \{y\} \\ f(y, a) &= \Phi & f(y, b) &= \{x, y\} \end{aligned}$$

试构造相应的确定有限自动机  $M'$ 。

**【解答】** 对照自动机的定义  $M = (S, \Sigma, f, s_0, Z)$ , 由  $f$  的定义可知  $f(x, a)$ 、 $f(y, b)$  均为多值函数, 因此  $M$  是一非确定有限自动机。

先画出 NFA M 相应的状态图, 如图 2-3 所示。

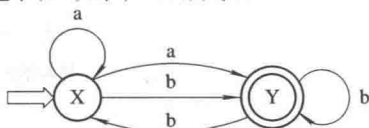


图 2-3 习题 2.3 的 NFA M

用子集法构造状态转换矩阵，如表 2-1 所示。

表 2-1 状态转换矩阵

I	$I_a$	$I_b$
{x}	{x,y}	{y}
{y}	—	{x,y}
{x,y}	{x,y}	{x,y}

将转换矩阵中的所有子集重新命名，形成表 2-2 所示的状态转换矩阵，即得到  $M' = (\{0,1,2\}, \{a,b\}, f, 0, \{1,2\})$ ，其状态转换图如图 2-4 所示。

表 2-2 重命名后的状态转换矩阵

状态 \ 字符	f	a	b
	0	2	1
1	—	—	2
2	2	2	2

将图 2-4 所示的 DFA  $M'$  最小化。首先，将  $M'$  的状态分成终态组  $\{1,2\}$  与非终态组  $\{0\}$ 。其次，考察  $\{1,2\}$ 。由于  $\{1,2\}_a = \{1,2\}$ ,  $\{1,2\}_b = \{2\} \subset \{1,2\}$ ，因此不再将其划分了，也即整个划分只有两组： $\{0\}$  和  $\{1,2\}$ 。令状态 1 代表  $\{1,2\}$ ，即把原来到达 2 的弧都导向 1，并删除状态 2。最后，得到如图 2-5 所示的化简了的 DFA  $M'$ 。

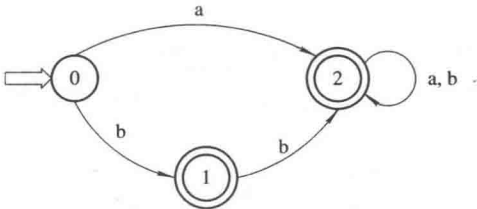


图 2-4 习题 2.3 的 DFA  $M'$

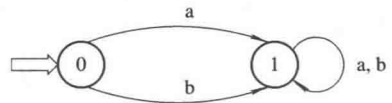


图 2-5 图 2-3 化简后的 DFA  $M'$

2.4 正规式  $(ab)^*a$  与正规式  $a(ba)^*$  是否等价？请说明理由。

【解答】正规式  $(ab)^*a$  对应的 NFA 如图 2-6 所示，正规式  $a(ba)^*$  对应的 NFA 如图 2-7 所示。

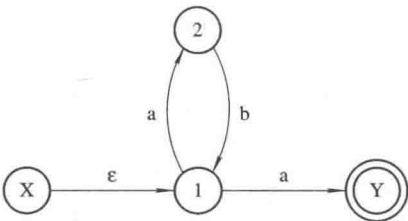


图 2-6 正规式  $(ab)^*a$  对应的 NFA

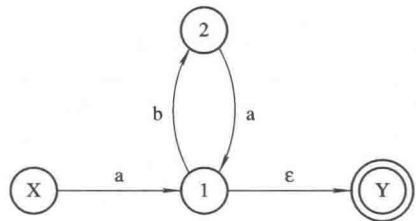


图 2-7 正规式  $a(ba)^*$  对应的 NFA

用子集法将图 2-6 和图 2-7 分别确定化为如图 2-8(a)和(b)所示的状态转换矩阵，它们最终都可以得到最简 DFA，如图 2-9 所示。因此，这两个正规式等价。

I	I <sub>a</sub>	I <sub>b</sub>
{X, 1}	{2, Y}	—
{2, Y}	—	{1}
{1}	{2, Y}	—

(a)

I	I <sub>a</sub>	I <sub>b</sub>
{X}	{1, Y}	{2}
{1, Y}	—	{2}
{2}	{1, Y}	—

(b)

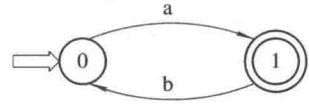
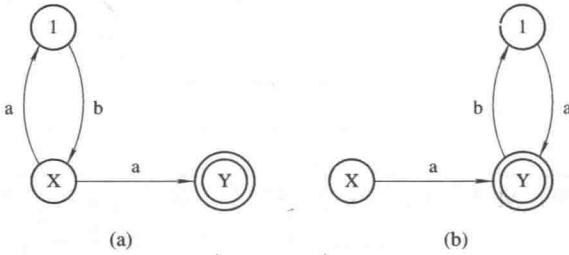


图 2-9 最简 DFA

图 2-8 图 2-6 和图 2-7 确定化后的状态转换矩阵

实际上,当闭包<sup>\*</sup>取 0 时,正规式 $(ab)^*a$ 与正规式 $a(ba)^*$ 由初态 X 到终态 Y 之间仅存在一条 a 弧。由于 $(ab)^*$ 在 a 之前,故描述 $(ab)^*$ 的弧应在初态结点 X 上;而 $(ba)^*$ 在 a 之后,故 $(ba)^*$ 对应的弧应在终态结点 Y 上。因此, $(ab)^*a$ 和 $a(ba)^*$ 所对应的 NFA 也可分别描述为如图 2-10(a)和(b)所示的形式,它们确定化并化简后仍可得到图 2-9 所示的最简 DFA。

图 2-10  $(ab)^*a$  和  $a(ba)^*$  分别对应的 NFA

2.5 设有  $L(G)=\{a^{2n+1}b^{2m}a^{2p+1} \mid n \geq 0, p \geq 0, m \geq 1\}$ 。

- 给出描述该语言的正规表达式;
- 构造识别该语言的确定有限自动机(可直接用状态图形式给出)。

**【解答】** 该语言对应的正规表达式为  $a(aa)^*bb(bb)^*a(aa)^*$ , 正规表达式对应的 NFA 如图 2-11 所示。

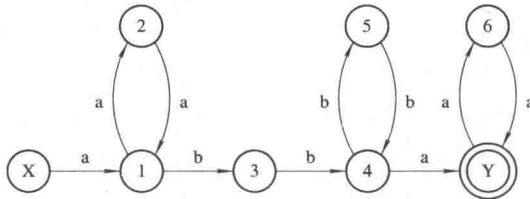


图 2-11 习题 2.5 的 NFA

用子集法将图 2-11 确定化,如图 2-12 所示。

I	I <sub>a</sub>	I <sub>b</sub>	重新命名	S	a	b
{X}	{1}	—			0	1
{1}	{2}	{3}		1	2	3
{2}	{1}	—		2	1	—
{3}	—	{4}		3	—	4
{4}	{Y}	{5}		4	7	5
{5}	—	{4}		5	—	4
{Y}	{6}	—		7	6	—
{6}	{Y}	—		6	7	—

图 2-12 习题 2.5 的状态转换矩阵

由图 2-12 重新命名后的状态转换矩阵可以看出：状态 0 和状态 2 面对输入字符 a、b 的下一状态相同，状态 3 和状态 5 面对输入字符 a、b 的下一状态相同，即得到划分后的状态子集为

$$\{0, 2\} \{1\} \{3, 5\} \{4\} \{6\} \{7\}$$

按顺序重新命名为 0、1、2、3、4、5 后得到最简的 DFA 如图 2-13 所示。

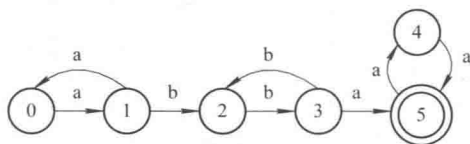


图 2-13 习题 2.5 的最简 DFA

注意，如果将状态 4 和状态 6 作为等价状态，即得到划分后的状态子集为

$$\{0, 2\} \{1\} \{3, 5\} \{4, 6\} \{7\}$$

按顺序重新命名为 0、1、2、3、4 后得到最简的 DFA' 如图 2-14 所示。由图 2-14 可以看出，由状态 4 输入 a 可以到达状态 3，由状态 3 输入 b 可以到达状态 2，即可形成如下的字符串：

$$aa \cdots abb \cdots baa \cdots abb \cdots baa \cdots abb \cdots baa \cdots a$$

而不是本题正规表达式可形成的字符串： $aa \cdots abb \cdots baa \cdots a$ 。

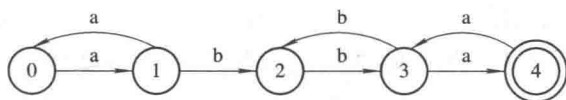


图 2-14 习题 2.5 的最简 DFA'

2.6 有语言  $L = \{w \mid w \in (0,1)^+, \text{ 并且 } w \text{ 中至少有两个 } 1, \text{ 又在任何两个 } 1 \text{ 之间有偶数个 } 0\}$ ，试构造接受该语言的确有限状态自动机(DFA)。

**【解答】** 对于语言 L，w 中至少有两个 1，且任意两个 1 之间必须有偶数个 0；也即在第一个 1 之前和最后一个 1 之后，对 0 的个数没有要求。据此我们求出 L 的正规式为  $0^*1(00(00)^*1)^*00(00)^*10^*$ ，画出与正规式对应的 NFA，如图 2-15 所示。

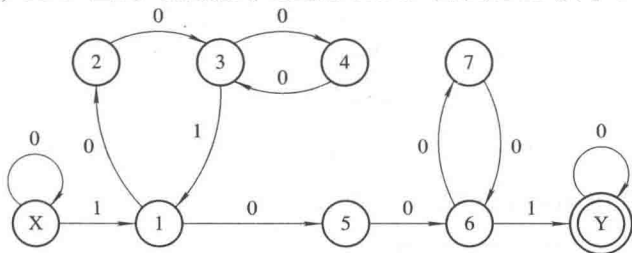


图 2-15 习题 2.6 的 NFA

用子集法将图 2-15 所示的 NFA 确定化，如图 2-16 所示。

由图 2-16 可看出非终态 2 和 4 的下一状态相同，终态 6 和 8 的下一状态相同，即得到最简状态为

$$\{0\} \{1\} \{2,4\} \{3\} \{5\} \{6,8\} \{7\}$$

按顺序重新命名为 0、1、2、3、4、5、6，则得到最简 DFA，如图 2-17 所示。

I	I <sub>0</sub>	I <sub>1</sub>
{X}	{X}	{1}
{1}	{2,5}	—
{2,5}	{3,6}	—
{3,6}	{4,7}	{1,Y}
{4,7}	{3,6}	—
{1,Y}	{2,5,Y}	—
{2,5,Y}	{3,6,Y}	—
{3,6,Y}	{4,7,Y}	{1,Y}
{4,7,Y}	{3,6,Y}	—

重新命名  
→

S	0	1
0	0	1
1	2	—
2	3	—
3	4	5
4	3	—
5	6	—
6	7	—
7	8	5
8	7	—

图 2-16 习题 2.6 的状态转换矩阵

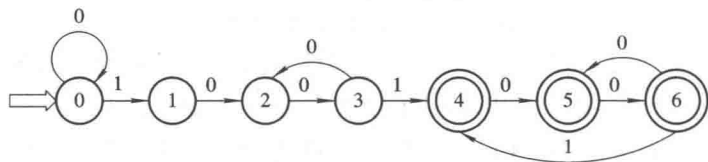


图 2-17 习题 2.6 的最简 DFA

2.7 已知正规式  $((a|b)^*|aa)^*b$  和正规式  $(a|b)^*b$ 。

- (1) 试用有限自动机的等价性证明这两个正规式是等价的；
- (2) 给出相应的正规文法。

【解答】 (1) 正规式  $((a|b)^*|aa)^*b$  对应的 NFA 如图 2-18 所示。

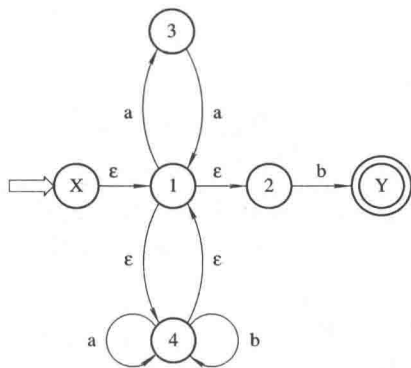


图 2-18 正规式  $((a|b)^*|aa)^*b$  对应的 NFA

用子集法将图 2-18 所示的 NFA 确定化为 DFA，如图 2-19 所示。

I	I <sub>a</sub>	I <sub>b</sub>
{X,1,2,4}	{1,2,3,4}	{1,2,4,Y}
{1,2,3,4}	{1,2,3,4}	{1,2,4,Y}
{1,2,4,Y}	{1,2,3,4}	{1,2,4,Y}

重新命名  
→

S	a	b
1	2	3
2	2	3
3	2	3

图 2-19 图 2-18 确定化后的状态转换矩阵

由于对非终态的状态 1、2 来说，它们输入 a、b 的下一状态是一样的，故状态 1 和状态 2 可以合并，将合并后的终态 3 命名为 2，则得到表 2-3(注意，终态和非终态即使输入 a、b 的下一状态相同也不能合并)。

表 2-3 合并后的状态转换矩阵

S	a	b
1	1	2
2	1	2

由此得到最简 DFA，如图 2-20 所示。

正规式  $(a|b)^*b$  对应的 NFA 如图 2-21 所示。

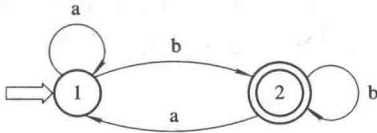


图 2-20 习题 2.7 的最简 DFA

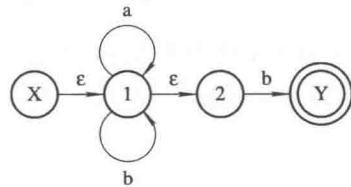


图 2-21 正规式  $(a|b)^*b$  对应的 NFA

用子集法将图 2-21 所示的 NFA 确定化为如图 2-22 所示的状态转换矩阵。

I	$I_a$	$I_b$	重新命名 →	S	a	b
{X,1,2}	{1,2}	{1,2,Y}		1	2	3
{1,2}	{1,2}	{1,2,Y}		2	2	3
{1,2,Y}	{1,2}	{1,2,Y}		3	2	3

图 2-22 图 2-21 确定化后的状态转换矩阵

比较图 2-22 与图 2-19，重新命名后的转换矩阵是完全一样的，也即正规式  $(a|b)^*b$  可以同样得到化简后的 DFA 如图 2-20 所示。因此，两个自动机完全一样，即两个正规文法等价。

(2) 对图 2-20，令 A 对应状态 1，B 对应状态 2，则相应的正规文法  $G[A]$  为

$$G[A]: A \rightarrow aA | bB | b$$

$$B \rightarrow aA | bB | b$$

$G[A]$  可进一步化简为  $G[S]: S \rightarrow aS | bS | b$  (非终结符 B 对应的产生式与 A 对应的产生式相同，故两非终结符等价，即可合并为一个产生式)。

2.8 构造一个 DFA，它接收  $\Sigma = \{a, b\}$  上所有不含子串 abb 的字符串。

【解答】 本题对应的正规表达式为  $b^*(a|ab)^*$ ，对应的 NFA 如图 2-23 所示。

用子集法将图 2-23 所示的 NFA 确定化为 DFA，如图 2-24 所示。

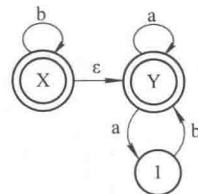


图 2-23 正规式  $b^*(a|ab)^*$  对应的 NFA

I	$I_a$	$I_b$	重新命名 →	S	a	b
{X, Y}	{1, Y}	{X, Y}		0	1	0
{1, Y}	{1, Y}	{Y}		1	1	2
Y	{1, Y}	—		2	1	—

图 2-24 图 2-23 确定化后的状态转换矩阵

由图 2-24 重新命名后的转换矩阵可以看出：状态 0、状态 1 和状态 2 对输入字符 b 的下一状态都是不一样的，故状态 0、状态 1 和状态 2 已为最简状态。由此得到最简 DFA，如图 2-25 所示。

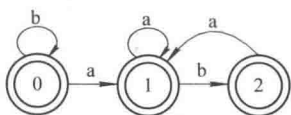


图 2-25 习题 2.8 的最简 DFA

注意，诸如  $a^*b^*$  这类正规式简化的 NFA 只能画成图 2-26 的形式，而不能画成图 2-27 的形式，图 2-27 对应的是正规式  $(a|b)^*$ 。本题对应的另一个正规表达式为  $b^*(a|ba)^*(ab)^*$ 。

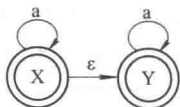


图 2-26  $a^*b^*$  的 NFA



图 2-27  $(a|b)^*$  的 NFA

2.9 构造一个 DFA，它接收  $\Sigma=\{a,b\}$  上所有含偶数个 a 的字符串。

【解答】 根据题意可以构造出字符串中含偶数个 a 的正规表达式： $(b|ab^*a)^*$ 。根据此正规表达式画出相应的 NFA M 如图 2-28 所示。

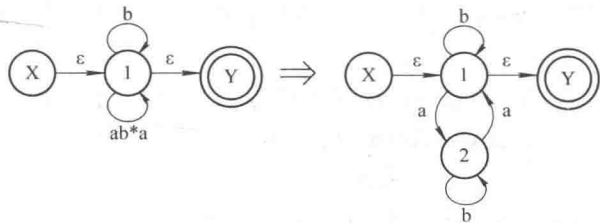


图 2-28 习题 2.9 的 NFA M

用子集法将图 2-28 所示的 NFA 确定化为 DFA，如图 2-29 所示。

I	$I_a$	$I_b$
{X, 1, Y}	{2}	{1, Y}
{2}	{1, Y}	{2}
{1, Y}	{2}	{1, Y}

→

S	a	b
0	1	2
1	2	1
2	1	2

图 2-29 图 2-28 确定化后的状态转换矩阵

由图 2-29 重新命名后的转换矩阵可以看出：状态 0 和状态 2 对输入字符 a、b 的下一状态都是一样的，故状态 0 和状态 2 可合并为一个状态。最终得到最简 DFA 如图 2-30 所示。

当然，我们也可以将图 2-28 中的状态 X 和状态 Y 与状态 1 合并而直接得到图 2-30。

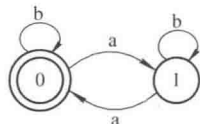


图 2-30 习题 2.9 的最简 DFA M

2.10 下列程序段以 B 表示循环体，A 表示初始化，I 表示增量，T 表示测试：

```

I=1;
while (I<=n)
{
    sun=sun+a[I];
    I=I+1;
}
    
```

请用正规表达式表示这个程序段可能的执行序列。

【解答】 用正规表达式表示程序段可能的执行序列为  $AT(BIT)^*$ 。

2.11 将图 2-31 所示的非确定有限自动机(NFA)变换成等价的确有限自动机(DFA)。其中，X 为初态，Y 为终态。

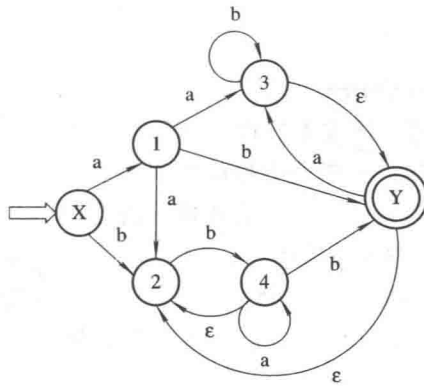


图 2-31 习题 2.11 的 NFA

【解答】 用子集法将 NFA 确定化，如图 2-32 所示。

I	I <sub>a</sub>	I <sub>b</sub>
{X}	{1}	{3}
{1}	{2,3,Y}	{3,Y}
{3}	—	{3,4}
{2,3,Y}	{2,3,Y}	{2,3,4,Y}
{3,Y}	{2,3,Y}	{3,4}
{3,4}	{3,4}	{3,4,Y}
{2,3,4,Y}	{2,3,4,Y}	{2,3,4,Y}
{3,4,Y}	{2,3,4,Y}	{3,4,Y}

重新命名

S	a	b
0	1	2
1	3	4
2	—	5
3	3	6
4	3	5
5	5	7
6	6	6
7	6	7

图 2-32 习题 2.11 的状态转换矩阵

图 2-32 所对应的 DFA 如图 2-33 所示。

对图 2-33 所示的 DFA 进行最小化。首先将状态分为非终态集和终态集两部分： $\{0,1,2,5\}$  和  $\{3,4,6,7\}$ 。由终态集可知，对于状态 3、6、7，无论输入字符是 a 还是 b 的下一状态均为终态集，而状态 4 在输入字符 b 的下一状态落入非终态集，故将其划分为

$$\{0,1,2,5\}, \{4\}, \{3,6,7\}$$



对于非终态集, 在输入字符 a、b 后按其下一状态落入的状态集不同而最终划分为

{0}, {1}, {2}, {5}, {4}, {3,6,7}

按顺序重新命名为 0、1、2、3、4、5, 得到最简 DFA 如图 2-34 所示。

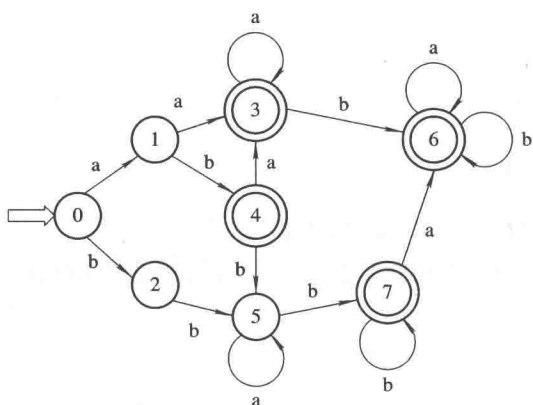


图 2-33 习题 2.11 的 DFA

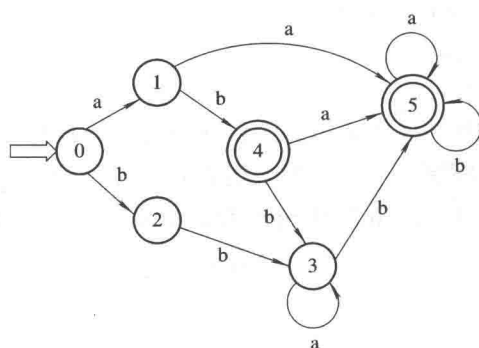


图 2-34 习题 2.11 的最简 DFA

2.12 有一台自动售货机, 接收 1 分和 2 分硬币, 出售 3 分钱一块的硬糖。顾客每次向机器中投放大于等于 3 分的硬币, 便可得到一块糖(注意: 只给一块并且不找钱)。

- (1) 写出售货机售糖的正规表达式;
- (2) 构造识别上述正规式的最简 DFA。

【解答】 (1) 设  $a=1$ ,  $b=2$ , 则售货机售糖的正规表达式为  $a(b|a(a|b))|b(a|b)$ 。

(2) 画出与正规表达式  $a(b|a(a|b))|b(a|b)$  对应的 NFA, 如图 2-35 所示。

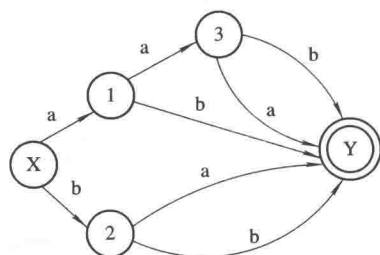


图 2-35 习题 2.12 的 NFA

用子集法将图 2-35 所示的 NFA 确定化, 如图 2-36 所示。

I	$I_a$	$I_b$
{X}	{1}	{2}
{1}	{3}	{Y}
{2}	{Y}	{Y}
{3}	{Y}	{Y}
{Y}	—	—

重新命名  $\Rightarrow$

S	a	b
0	1	2
1	3	4
2	4	4
3	4	4
4	—	—

图 2-36 习题 2.12 的状态转换矩阵

由图 2-36 可看出, 非终态 2 和非终态 3 面对输入符号 a 或 b 的下一状态相同, 故合并为一个状态, 即最简状态 {0}、{1}、{2, 3}、{4}。按顺序重新命名为 0、1、2、3, 则得到最简 DFA, 如图 2-37 所示。

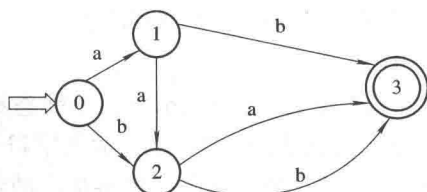


图 2-37 习题 2.12 的最简 DFA