第十章 模式、自动机和正则表达式

与模式相关的两个基本问题是模式的定义和识别，这也是本章和下一章的主题。在某些应用中，模式识别是必不可少的，比如图10.1所示的光学字符识别的例子。在有些应用中，模式识别是一个较大问题的一部分。比如，程序的模式识别是编译器的必要组成部分，即将一种语言翻译为另一种语言，如将C语言翻译为机器语言。

在计算机科学中开有许多其它应用模式的例子。模式在构造计算机和其它数字设备的电子电路中起着关键作用。在文本编辑中，模式用来查找指定的单词或字符串，比如“the letters if followed by any sequence of characters followed by then”。大多数操作系统可以在命令中使用模式，比如在UNIX操作系统中“ls \*tex”可以列出以“tex”三个字符结尾的文件。

围绕着模式定义和识别已经进行了广泛的研究，这些理论称为“自动机理论”或“语言理论”，它的基本定义和技术是计算机科学核心组成部分。

10.1 本章内容

本章处理包含字符串集合的模式。通过本章，我们将学习：

* **有穷自动机**是一种基于图形的模式表示方法。它有两种变形：**确定型有穷自动**机（10.2节）和**非确定型有穷自动机**（10.3节）。
* 确定型有穷自动机可以通过一种简单的方法转换为识别其模式的程序（10.2节）。
* 使用**子集构造**的方法可以将非确定型有穷自动机转换成识别同样模式的确定型有穷自动机（10.4节）。
* **正则表达式**是可以被自动机描述的模式的代数描述（10.5和10.7节）。
* 正则表达式可以转换为自动机（10.8节），反之亦然（10.9节）。

下一章也会讨论字符串模式，将介绍一种用来定义模式 的递归表示标记法：“上下文无关文法”。我们将会看到这种标记法可以描述自动机或正则表达式无法表达的模式。然而，在很多情况下，无法像自动机或者正则表达式那样简单的方式将文法转换转换为程序。

10.2 状态机和自动机

用来搜索模式字符串的程序通常具有特殊的结构。我们可以在代码的特定位置做些标识，这些位置表示查找某个目标模式的程序过程中的一些特殊的事件。我们称这些位置为**状态**。程序的整体行为可以看作是读入输入时从一个状态迁移到另一个状态。

为更加具体的描述这种思想，我们来看一个模式匹配的例子：“哪些英语单词包括五个元音字母，且是按照字母顺序出现的？”。为了回答这个问题，我们可以使用操作系统提供的单词表，如UNIX系统中的/usr/dict/words，其中保存了常用的英语单词，每行一个。在该文件中，包含按顺序出现的五个元音字母的单词是：

abstemious

facetious

sacrilegious

我们来用C写一个程序检查一个字符串，判断其中是否包括所有五个元音字符并且按顺序出现。