### 语义学的使用

在前面的例子中,假定基元间的相互连接只在该点发生,如图 12.26(b)所示。在更为复杂的情况下,必须搞清楚连通性的规则、其他因素的有关信息(如基元的长度和方向)、产生式可以应用的次数等因素。这些可以通过存储在知识库(见图 1.23)中的语义规则来实现。基本上,在产生式规则中句法内在的规则设定了对象的结构,但语义学处理对象的正确性。例如,像 C 语言这样的编程语言,$A = D/E$ 在句法上是正确的,但只有在 $E \neq 0$ 时,在语义上才是正确的。

设想把语义信息和前面例子中谈到的文法结合起来。语义信息可以同表 12.1 中的产生式联系起来。通过使用语义信息,可以使用少量的句法规则描述广泛的类模式(但受我们的要求限制)。例如,通过指定表 12.1 中 $\theta$ 的方向,避免了在每个方向上指定元素。同样,通过要求所有元素定位于同一个方向,可以不去考虑由图 12.26(a)代表的偏离基本形状的无意义结构。

表 12.1 产生规则附带的语义信息实例

| 产 生 式 | 语 义 信 息 |
|---|---|
| $S \rightarrow aA$ | 与 $a$ 的连接只在该点进行。表示为 $\theta$ 的 $a$ 的方向由直线的垂直等分线给出。此直线连接两条没有点的线段之终点。每条线段长度为 3 cm |
| $A \rightarrow bA$ | 与 $b$ 的连接只对该点进行。不允许多连接。$b$ 的方向与 $a$ 相同。$b$ 的长度为0.25 cm。此产生式不能应用超过 10 次 |
| $A \rightarrow bB$ | $a$ 和 $b$ 的方向必须相同。连接必须简单并只对该点进行此操作 |
| $B \rightarrow c$ | $c$ 和 $a$ 的方向必须一致。连接必须简单并只对该点进行此操作 |

### 作为串识别器的自动机

到现在为止,已经证明文法是模式的生成器。在下边的论述中,考虑识别一个模式是否属于由文法 $G$ 生成的语言 $L(G)$的模式。构成句法识别的基本概念可以由称为自动机的计算机器数学模型的发展来说明。给出一个输入模式的串,自动机有能力识别模式是否属于与这个自动机相关联的语言。这里,只关注有限自动机,它是由正则文法生成的语言识别器。

一个有限自动机定义为五元式:

$$A_f = (Q, \Sigma, \delta, q_0, F) \tag{12.3.7}$$

这里 $Q$ 是有限的、非空的状态集合,$\Sigma$是有限的输入字母集合,$\delta$ 是从 $Q \times \Sigma$(来自 $Q$ 和$\Sigma$的元素的有序对构成的集合)到 $Q$ 的所有子集集合的映射,$q_0$ 是起始状态,$F$($Q$ 的一个子集)是一个终了或接受状态集合。

**例 12.10 一个简单的自动机**

考虑由式(12.3.7)给出的自动机,$Q = \{q_0, q_1, q_2\}$,$\Sigma = \{a, b\}$,$F = \{q_0\}$及映射 $\delta(q_0, a) = \{q_2\}$,$\delta(q_0, b) = \{q_1\}$,$\delta(q_1, a) = \{q_2\}$,$\delta(q_1, b) = \{q_0\}$,$\delta(q_2, a) = \{q_0\}$和 $\delta(q_2, b) = \{q_1\}$。例如,如果自动机处在状态 $q_0$ 并且当前输入为 $a$,自动机的状态变为 $q_2$。同样,如果 $b$ 是接下来的输入,自动机的状态变为 $q_1$,等等,在这种情况下,初始和终结状态是相同的。

图 12.27 显示了刚才谈到的自动机状态图。状态图由表示每个状态的节点和表示可能的状态转换的有向弧组成。终态用双圆圈表示，每条弧用符号标注，它引起由弧连接的状态间的转换。在这种情况下，初始状态和终结状态是相同的。一个终端符号串 $w$ 如果从状态 $q_0$ 开始，在扫描了串 $w$ 的最后一个字符后，字符序列(和 $w$ 被扫描的顺序一样是从左到右读入)使自动机到达终结状态，就称符号串 $w$ 被接受了或被识别了。例如，图 12.27 中的自动机识别串 $w = abbabb$，但拒绝串 $w = aabab$。

正则文法和有限自动机是一一对应的。即，当且仅当它是由一个正则文法生成的，一种语言被有限自动机识别。一个基于刚才谈到概念的句法串识别器的设计是一个直观过程，从一个给定的正则文法得到一个有限自动机。令文法由 $G = (N, \Sigma, P, X_0)$表示，这里 $X_0 \equiv S$，并假设 $N$ 由 $X_0$ 加上 $n$ 个附加的非终端符 $X_1, X_2, \cdots, X_n$ 组成。自动机的集合 $Q$ 由引入的 $n+2$ 个状态 $\{q_0, q_1, \cdots, q_n, q_{n+1}\}$ 组成，以便 $q_i$ 和 $X_i (0 \leqslant i \leqslant n)$ 相对应，且 $q_{n+1}$ 为终结状态。

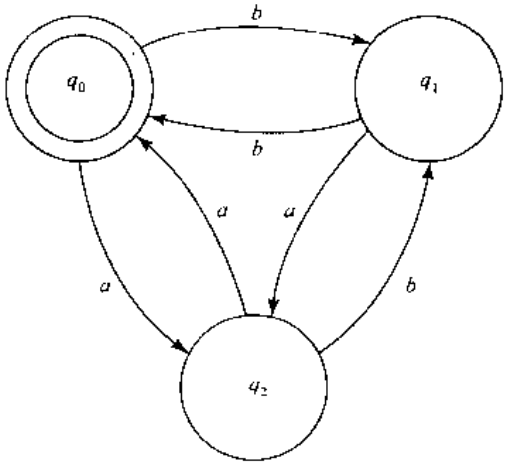



图 12.27　一个有限自动机

输入符号集合与 $G$ 中的终端符集合相一致。$\delta$ 中的映射通过使用基于 $G$ 的产生式的两条规则得到；即，对每个 $i$ 和 $j$，$0 \leqslant i \leqslant n$，$0 \leqslant j \leqslant n$：

1. 如果 $X_i \rightarrow aX_j$ 属于 $P$，则 $\delta(q_i, a)$包含 $q_j$。

2. 如果 $X_i \rightarrow a$ 属于 $P$，则 $\delta(q_i, a)$包含 $q_{n+1}$。

相反，给定一个有限自动机，$A_f = (Q, \Sigma, \delta, q_0, F)$，令 $Q$ 的元素组成 $N$，起始符 $X_0$ 对应 $q_0$，$G$ 的产生式用下列方法取得：

1. 如果 $q_j$ 属于 $\delta(q_i, a)$，则 $P$ 中有产生式 $X_i \rightarrow aX_j$。

2. 如果 $F$ 中有状态属于 $\delta(q_i, a)$，则 $P$ 中有产生式 $X_i \rightarrow a$。

由此得到对应的正则文法，$G = (N, \Sigma, P, X_0)$。两种情况下的终端符号集合是一样的。

**例 12.11**　识别图 12.26 中模式的有限自动机

与图 12.26 相关的文法的有限自动机，是通过将产生式写为 $X_0 \rightarrow aX_1$，$X_1 \rightarrow bX_1$，$X_1 \rightarrow bX_2$，$X_2 \rightarrow c$，然后 $A_f = (Q, \Sigma, \delta, q_0, F)$，$Q = \{q_0, q_1, q_2, q_3\}$，$\Sigma = \{a, b, c\}$，$F = \{q_3\}$和映射 $\delta(q_0, a) = \{q_1\}$，$\delta(q_1, b) = \{q_1, q_2\}$，$\delta(q_2, c) = \{q_3\}$得到的。为了使其具有完备性，令

$\delta(q_0,b)=\delta(q_0,c)=\delta(q_1,a)=\delta(q_1,c)=\delta(q_2,a)=\delta(q_2,b)=\varnothing$，这里$\varnothing$代表空集，表示本自动机未定义这些转换。

### 12.3.4 树的句法识别

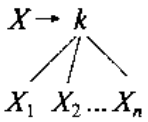
下面要谈的形式与前面讨论过的串形式相似，现在将讨论扩展到模式的树形描述。假设图像区域或感兴趣的对象通过使用如 11.5 节中所讨论的适当原始基元，以树的形式进行表达。

#### 树文法

树文法以五元式定义：

$$G=(N,\Sigma,P,r,S) \tag{12.3.8}$$

这里，和以前一样，$N$ 和$\Sigma$分别为非终端符集合和终端符集合；$S$ 包含于 $N$ 中，是起始符，一般来讲，它也可独立构成一棵树；$P$ 是形如 $T_i\rightarrow T_j$ 的产生式集合，这里 $T_i$ 和 $T_j$ 为树；$r$ 是秩函数，表示在文法中节点的直接下降(后代)数目，这些符号在文法中标为一个终端符。所讨论文法的特殊关联性是具有如下形式产生式的开销很大的树文法。




这里，$X_1,X_2,\cdots,X_n$ 为非终端符，$k$ 是终端符。

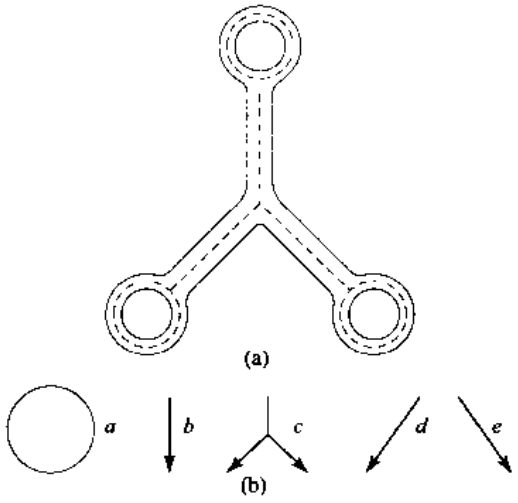
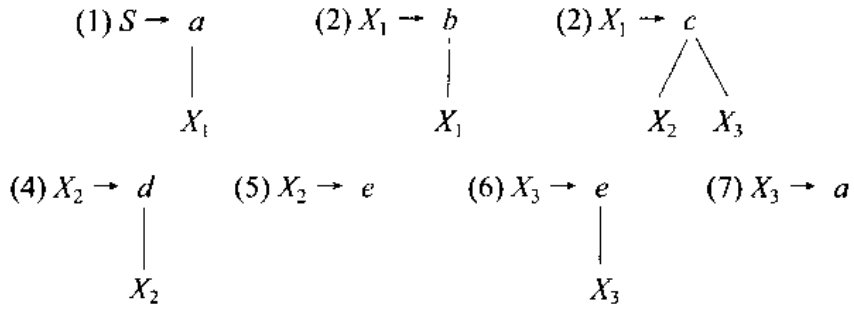



图 12.28 (a)一个对象和(b)借助于树文法表示骨架的基元

**例 12.12 一个简单的树文法**

图 12.28(a)所示的骨架结构可以使用树文法产生，其中 $N=\{X_1,X_2,X_3,S\}$，$\Sigma=\{a,b,c,d,e\}$，这里终端符代表图 12.28(b)中所示的基元。假设线性基元有首尾的连接性并沿着圆周有任意的连接，在这种考虑下的文法具有如下形式的产生式：




此时的秩函数为 $r(a)=\{0,1\}, r(b)=r(d)=\{1\}, r(e)=\{0,1\}, r(c)=\{2\}$。限制产生式(2),(4),(6)应用相同的次数会生成所有 3 个支路具有相同长度的结构。同样,产生式(4),(6)应用相同的次数会生成关于垂直轴对称的结构。这种类型的语义信息与先前表 12.1相关的讨论相似,并且基于图 1.23 的知识。

**树自动机**

尽管传统的有限自动机从左到右逐符号扫描输入字符串,但一个树自动机必须同时从输入树末端(对叶子节点采取从左到右的顺序)的每个节点开始,并沿着并行的路线向着根节点方向处理。特别地,一个末端到根的自动机定义为:

$$A_t=(Q,F,\{f_k \mid k\in\Sigma\}) \tag{12.3.9}$$

这里

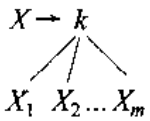
$Q$ 是一个有限的状态集合,

$F$ 是 $Q$ 的子集,是一个有限状态集,

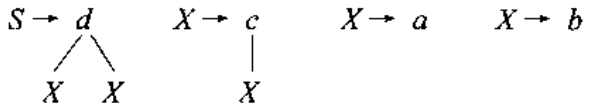
$f_k$ 是 $Q^m\times Q$ 上的关系,$m$ 是 $k$ 的秩。

符号 $Q^m$ 指 $Q$ 的 $m$ 次笛卡儿乘积:$Q^m=Q\times Q\times Q\times\cdots\times Q$。从笛卡儿乘积的定义,我们知道这个表达式意味着带有来自 $Q$ 的元素的有序 $m$ 元组集合。例如,如果 $m=3$,则 $Q^3=Q\times Q\times Q=\{x,y,z \mid x\in Q, y\in Q, z\in Q\}$。回想从集合 $A$ 到集合 $B$ 的关系 $R$ 是集合 $A$ 和 $B$ 的笛卡儿乘积的子集;即,$R\subseteq A\times B$。因此,在 $Q^m\times Q$ 上的关系仅仅是集合 $Q^m\times Q$ 的一个子集。

对一个开销很大的树文法,$G=(N,\Sigma,P,r,S)$,令 $Q=N, F=\{S\}$并对每个属于$\Sigma$的字符 $a$ 定义一个关系 $f_k$,以便当且仅当 $G$ 中有产生式:




时,$(X_1,X_2,\cdots,X_m,X)$ 在 $f_k$ 内来构造相应的树自动机。例如,考虑树文法 $G=(N,\Sigma\ P,r,S)$,其中 $N=\{S,X\}$,$\Sigma=\{a,b,c,d\}$,产生式:



并且秩为 $r(a)=\{0\}, r(b)=\{0\}, r(c)=\{1\}, r(d)=\{2\}$。相应的树自动机,$A_t=(Q,F,\{f_k \mid k\in\Sigma\})$,通过令 $Q=\{S,X\}$,$F=\{S\}$和$\{f_k \mid k\in\Sigma\}=\{f_a,f_b,f_c,f_d\}$来指定,这里关系定义为:

$f_a=\{(\varnothing,X)\}$,由产生式 $X\rightarrow a$ 产生

$f_b=\{(\varnothing,X)\}$,由产生式 $X\rightarrow b$ 产生

$f_c=\{(X,X)\}$,由产生式 $X\rightarrow$ $c$ 下接 $X$

和

$f_d=\{(X,X,S)\}$,由产生式 $S\rightarrow$ $d$ 下接 $X$ $X$ 产生。

关系 $f_a$ 的解释是,一个标记为 $a$ 的没后代(因此用空符号$\varnothing$)节点赋予状态 $X$。关系 $f_c$ 的解释是,一个标记为 $c$ 的(带有一个有状态 $X$ 的后代节点)节点赋予状态 $X$。关系 $f_d$ 的解释是,一个标记为 $d$ 的(带有两个后代节点,每个后代节点都有状态 $X$)节点赋予状态 $S$。

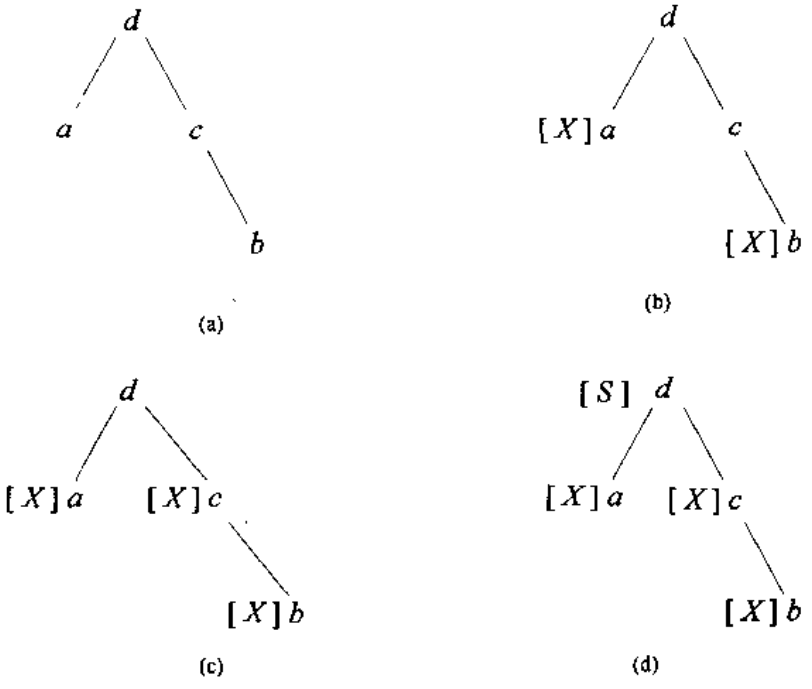



图 12.29 自底向根的树自动机的处理阶段:(a)输入树,(b)赋予叶子节点的状态,(c)赋予中间节点的状态,(d)赋予根节点的状态

为了了解这个树自动机如何识别由前面讨论过的文法生成的树,考虑图 12.29(a)显示的树。自动机 $A_t$ 首先分别通过关系 $f_a$ 和 $f_b$ 将状态赋予末端节点 $a$ 和 $b$。在这种情况下,根据这两个关系,状态 $X$ 赋给两个叶子节点,如图 12.29(b)所示。现在,自动机从末端节点向上移动一个层次并以 $f_c$ 和节点的后代状态为基础,使一个状态赋予节点 $c$。再一次基于 $f_c$ 的状态赋

值为 $X$，如图 12.29(c)所示。再向上移动一层，自动机遇到节点 $d$，由于状态已经赋予 $d$ 的两个后代，所以要求调用关系 $f_a$ 把状态 $S$ 赋予节点 $d$。因为这是最后一个节点，并且状态 $S$ 属于 $F$，自动机接受(识别)此树作为前面提到的树文法所产生语言的有效成员。图 12.29(d)显示了沿着自底向根路径的状态顺序的最终描述图。

**例 12.13**　使用树文法识别气室中现象的图像

在高能物理实验中气室中发生的现象被记录为一幅图像。这个物理实验是将一束已知性质的粒子导向已知核的目标。典型的现象由从碰撞点发散出来的次级粒子的运行轨迹组成，如图 12.30 显示的例子。引入的轨迹是水平的平行线。注意照片中心附近结果的自然树状结构。

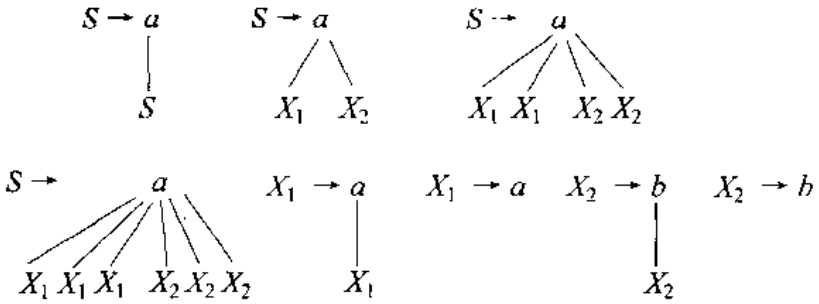
典型的实验产生数以十万计的照片，其中许多没有包含感兴趣的现象。对这些照片进行人工检查和分类费时费力，因此产生了对自动现象识别技术的需求。

一个树文法 $G=(N,\sum,P,r,S)$可以被指定为生成的这些典型事件的树状描述，代表在氢气室中确定引入带电粒子流后发现的典型现象的树。此时，$N=\{S,X_1,X_2\}$，$\sum=\{a,b\}$，元素 $a$ 和 $b$ 有如下解释：

$a$：$\frown$ 凸起弧

$b$：$\smile$ 凹下弧

$P$ 中产生式为：




秩为 $r(a)=\{0,1,2,4,6\}$，$r(b)=\{0,1\}$。分支产生式代表从碰撞点发出的发散轨迹的数目，一般成对出现，通常不超过 6 条。图 12.31(a)显示了图 12.30 中的碰撞现象被分割为凸起和凹下的部分，并且图 12.31(b)显示了对应的树表示。这棵树及其各种变形可以通过上述文法生成。

需要识别刚刚讨论的树类型的树自动机由前边概括的过程来定义。因此有，$A_t=(Q,F,\{f_k\mid k\in\sum\})$通过令 $Q=\{S,X_1,X_2\}$，$F=\{S\}$和$\{f_k\mid k\in\sum\}=\{f_a,f_b\}$指定。关系定义为：$f_a=\{(S,S),(X_1,X_2,S),(X_1,X_1,X_2,X_2,S),(X_1,X_1,X_1,X_2,X_2,X_2,S),(X_1,X_1),(\varnothing,X_1)\}$且 $f_b=\{(X_2,X_2),(\varnothing,X_2)\}$。我们将这个例子留做练习以显示这个自动机可以接受图 12.31(b)中的树。

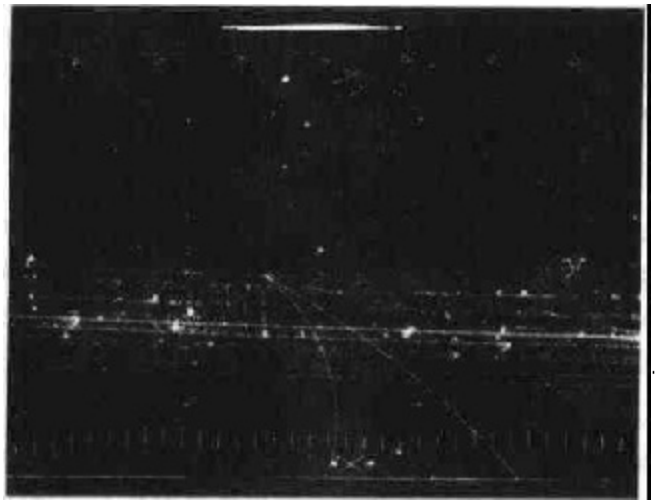


图 12.30 一张气室照片(Fu 和 Bhargava)

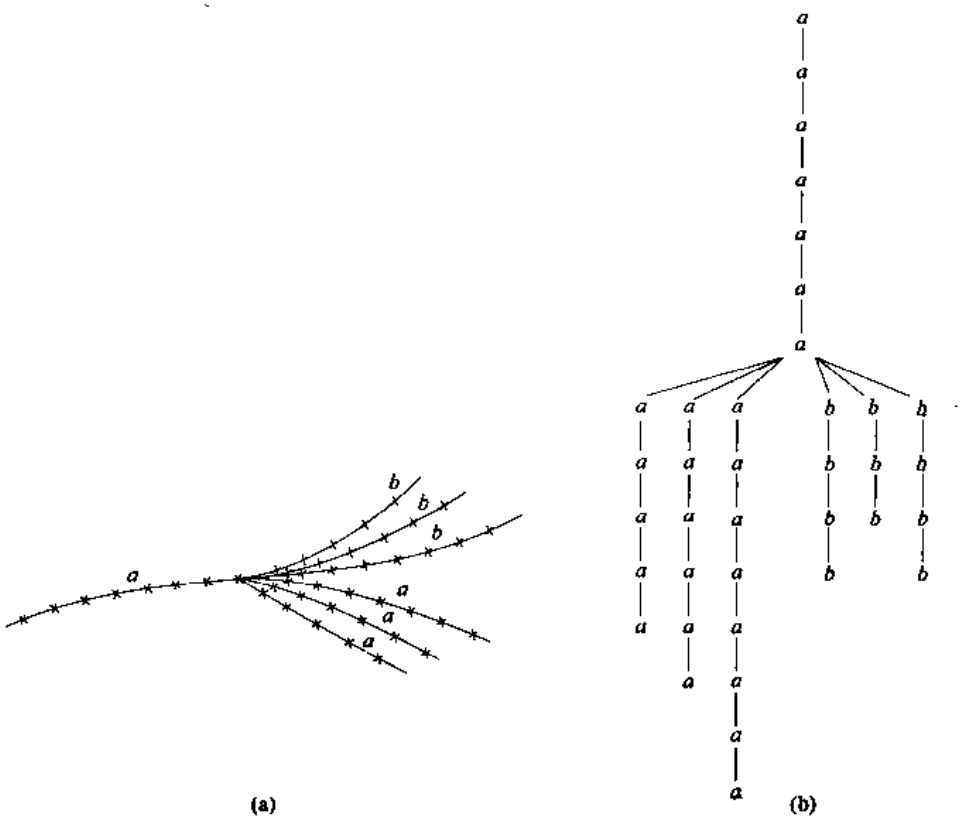



图 12.31 (a)对图 12.30 中的现象编码,(b)相应树的表示(Fu 和 Bhargava)

## 学习

前段介绍的句法识别方法需要为每个要考虑的类规定一个合适的自动机(识别器)。对于简单的情况,检验可以产生需要的自动机。在更复杂的情况下,自动机从样本模式(如串或树)中学习的算法也是需要的。因为前述的文法和自动机一一对应,直接由样本模式学习文法来提出学习问题,这一过程有时称为文法推论。直接从样本模式串学习有穷自动机是我们关注

的焦点。这一章结尾提供了学习树文法及其自动机和其他句法识别方法的指导。

假如一个类的所有模式由一个未知文法 $G$ 产生，并假设具有下列性质的有限样本集合 $R^+$

$$R^+ \subseteq \{v \mid v \text{ in } L(G)\} \tag{12.3.10}$$

是可用的。集合 $R^+$ 称为正样本集。它仅是来自与文法 $G$ 有关的类训练模式的一个集合。如果 $G$ 中每个产生式都被应用于生成至少一个 $R^+$ 中的元素，则就说这个样本集合在结构上是完善的。我们希望学习(组成)一个有限自动机 $A_f$，它将接受来自 $R^+$ 的字串并可能接受类似 $R^+$ 的字串。

基于有限自动机的定义和 $G$ 与 $A_f$ 的对应性，允许 $R^+ \subseteq \sum^*$，这里 $\sum^*$ 是由来自 $\sum$ 的元素组成的所有串的集合。令 $\sum^*$ 中的 $z$ 是一个串，对于 $\sum^*$ 中的某些 $w$，有 $zw$ 仍属于 $R^+$。对于一个正整数 $k$，将关于 $R^+$ 的 $z$ 的 $k$ 尾定义为集合 $h(z, R^+, k)$，其中：

$$h(z, R^+, k) = \{w \mid zw \text{ in } R^+, |w| \leqslant k\} \tag{12.3.11}$$

换句话说，$z$ 的 $k$ 尾是带有以下性质的串 $w$ 的集合：(1)$zw$ 属于 $R^+$ 和(2)$w$ 的长度小于或等于 $k$。

从一个样本集合 $R^+$ 和一个 $k$ 的特殊值学习自动机 $A_f(R^+, k) = (Q, \sum, \delta, q_0, F)$ 的过程，由令：

$$Q = \{q \mid q = h(z, R^+, k),\ z \text{ in } \sum^*\} \tag{12.3.12}$$

和对 $\sum$ 中的每个 $a$，

$$\delta(q, a) = \{q' \text{in } Q \mid q' = h(za, R^+, k),\ q = h(z, R^+, k)\} \tag{12.3.13}$$

组成。

另外，令：

$$q_0 = h(\lambda, R^+, k) \tag{12.3.14}$$

且

$$F = \{q \mid q \text{ in } Q, \lambda \text{ in } q\} \tag{12.3.15}$$

这里，$\lambda$ 是空串(没有字符的串)。注意到自动机 $A_f(R^+, k)$ 具有从 $R^+$ 构造的所有 $k$ 尾集合的状态子集。

**例 12.14　推导一个来自样本模式的有限自动机**

假如 $R^+ = \{a, ab, abb\}$ 且 $k = 1$。则从前边的讨论有：

$$\begin{aligned}
z = \lambda, \quad & h(\lambda, R^+, 1) = \{w \mid \lambda w \text{ in } R^+, |w| \leqslant 1\} \\
& = \{a\} \\
& = q_0 \\
z = a, \quad & h(a, R^+, 1) = \{w \mid aw \text{ in } R^+, |w| \leqslant 1\} \\
& = \{\lambda, b\} \\
& = q_1 \\
z = ab, \quad & h(ab, R^+, 1) = \{\lambda, b\} \\
& = q_1
\end{aligned}$$

$$z = abb, \qquad h(abb, R^+, 1) = \{\lambda\} = q_2$$

在这种情况下,$\sum^*$ 中的其他串 $z$ 生成不属于 $R^+$ 的串 $zw$,使自动机到达第四个状态,由 $q_\varnothing$ 定义,这与 $h$ 为空集的状态相对应。由此,状态为:$q_0 = \{a\}, q_1 = \{\lambda, a\}, q_2 = \{\lambda\}$ 和 $q_\varnothing$,它们给出集合 $Q = \{q_0, q_1, q_2, q_\varnothing\}$。尽管状态由字符集合($k$ 尾数)得到,状态符号 $q_0, q_1, \cdots$ 仅仅被用于形成集合 $Q$。下一步是得到转移函数。由于 $q_0 = h(\lambda, R^+, 1)$,得出:

$$\delta(q_0, a) = h(\lambda a, R^+, 1) = h(a, R^+, 1) = q_1$$

和

$$\delta(q_0, b) = h(\lambda b, R^+, 1) = h(b, R^+, 1) = q_\varnothing$$

类似地,

$$q_1 = h(a, R^+, 1) = h(ab, R^+, 1)$$

得出:

$$\delta(q_1, a) = h(aa, R^+, 1) = h(aba, R^+, 1) = q_\varnothing$$

还有,$\delta(q_1, b) \supseteq h(ab, R^+, 1) = q_1$ 且 $\delta(q_1, b) \supseteq h(abb, R^+, 1) = q_2$,即 $\delta(q_1, b) = \{q_1, q_2\}$。按照刚才的描述,得出 $\delta(q_2, a) = \delta(q_2, b) = \delta(q_\varnothing, a) = \delta(q_\varnothing, b) = q_\varnothing$。终结状态集合包含那些在 $k$ 尾中有空字符串 $\lambda$ 的状态。此时,$q_1 = \{\lambda, b\}$ 且 $q_2 = \{\lambda\}$,所以 $F = \{q_1, q_2\}$。

基于这些结果,推导的自动机给出如下:

$$A_f(R^+, 1) = (Q, \sum, \delta, q_0, F)$$

这里,$Q = \{q_0, q_1, q_2, q_\varnothing\}$,$\sum = \{a, b\}$,$F = \{q_1, q_2\}$,转移函数如上面给出的。图 12.32 显示了状态图。自动机接受形如 $a, ab, abb, \cdots, ab^n$ 的串,这些串和给定的样本集一致。

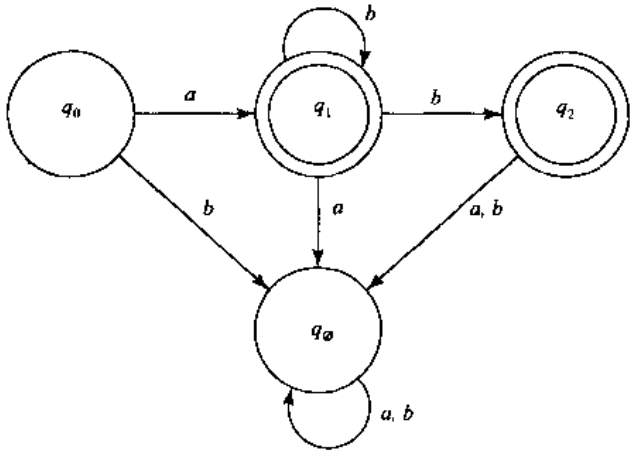



图 12.32 由样本集合 $R^+ = \{a, ab, abb\}$ 推出的有限自动机状态图

前述例子显示出 $k$ 值控制着结果自动机的性质。下面的性质说明 $A_f(R^+,k)$对这一参数的依赖性。

性质 1:对于所有的 $k\geqslant 0$, $R^+\subseteq L[A_f(R^+,k)]$,这里 $L[A_f(R^+,k)]$是 $A_f(R^+,k)$接受的语言。

性质 2:如果 $k$ 大于或等于 $R^+$ 中最长的串长度,则 $L[A_f(R^+,k)]=R^+$;如果 $k=0$,则 $L[A_f(R^+,k)]=\sum^*$。

性质 3:$L[A_f(R^+,k+1)]\subseteq L[A_f(R^+,k)]$

性质 1 保证 $A_f(R^+,k)$作为最小值,将接受样本集合 $R^+$ 中的串。如果 $k$ 大于或等于 $R^+$ 中最长串的长度,则由性质 2,自动机将仅接受 $R^+$ 中的串。如果 $k=0$, $A_f(R^+,0)$将由一个状态 $q_0=\{\lambda\}$构成,它既是初始状态又是终结状态。对$\sum$中的 $a$,转移函数将具有形式 $\delta(q_0,a)=q_0$。所以, $L[A_f(R^+,0)]=\sum^*$,并且自动机将接受空串 $\lambda$ 和所有由来自$\sum$的字符组成的串。最后,性质 3 指出 $A_f(R^+,k)$接受的语言规模在 $k$ 增大时减小。

这三条性质允许仅仅通过改变参数 $k$ 来控制 $A_f(R^+,k)$的性质。如果 $L[A_f(R^+,k)]$是来自被选中样本 $R^+$ 的语言 $L_0$ 的一种猜测,并且 $k$ 值非常小,那么这个语言 $L_0$ 的猜测会构成一个也许包含$\sum^*$ 中大部分或全部串的自由推论。然而,如果 $k$ 等于 $R^+$ 中最长串的长度, $A_f(R^+,k)$将仅接受包含在 $R^+$ 中串的推论会比较保守。图 12.33 用图形显示了这些概念。

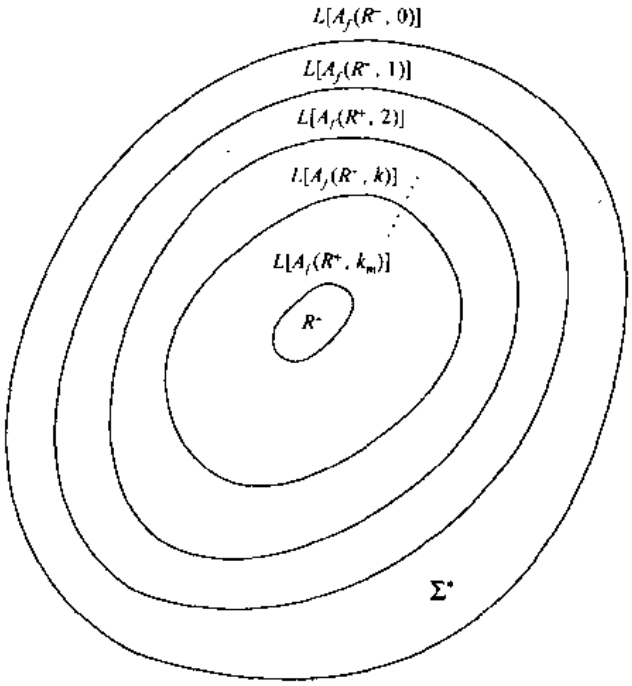



图 12.33　$L[A_f(R^+,k)]$和 $k$ 的关系,值 $k_m$ 大于或等于 $R^+$ 中最长串的长度

**例 12.15**　从给定模式集合中推导自动机的另一个例子

考虑集合 $R^+=\{caaab, bbaab, caab, bbab, cab, bbb, cb\}$。对 $k=1$,使用前述方法得出下列相同的过程: