

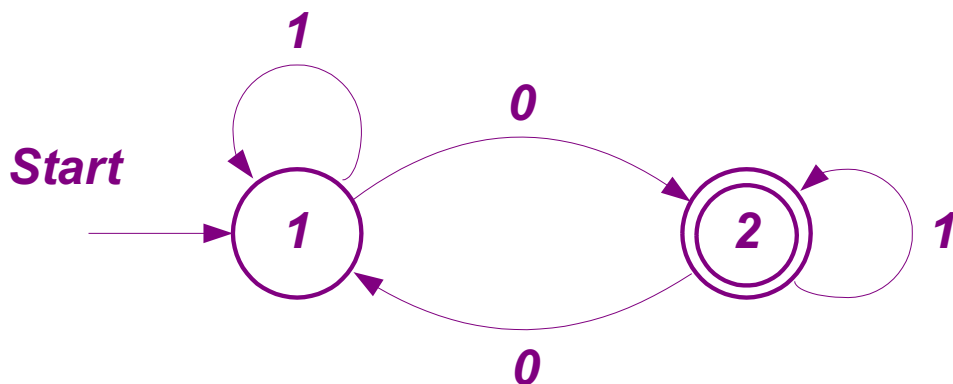
复习

■ 上一课的内容

- 形式语言 自动机

■ 例题

- 枚举出该自动机能够识别的字符串，并指出这些字符串的特点。



- 110, 11011, 1101010, 101010100
- $L = \{w \mid w \text{ 含有奇数个 } 0\}$



第二章 语言及文法

■ 主要内容：

- 定义形式语言的术语
- 给出文法的定义和文法的分类

■ 要求掌握：

- 语言和文法的形式定义
- **CHOMSKY**文法体系的分类。



第一节 语言的定义与运算

一、语言的一些术语：

- 字母表：字符的有限集合，记为 T 。
- 字符串：由字母表 T 中的字符构成的序列称字母表 T 上的字符串（句子）。
 - 常记为 u, v, w, x, y, z ；
 - 常用 a, b, c, d 标识单个字符。



字母表 (*Alphabet*)

✧ 概念 形式符号的集合

✧ 记号 常用 T 、 Σ 表示

✧ 举例

- 英文字母表 $\{ a, b, \dots, z, A, B, \dots, Z \}$
- 英文标点符号表 $\{ , ; : . ? ! ' ' " " () \dots \}$
- 汉字表 $\{ \dots, \text{自}, \dots, \text{动}, \dots, \text{机}, \dots \}$
- 化学元素表 $\{ H, He, Li, \dots, \}$
- $T = \{ a, n, y, \text{任,意} \}$



字符串 (*string*)

◇ 概念 字母表 T 上的一个字符串（简称串），或称为字 (*word*)，为 T 中字符构成的一个有限序列。空串 (*empty string*)，用 ε 表示，不包含任何字符。

举例 设 $T = \{ a, b \}$ ，则 $\varepsilon, a, ba, bbaba$ 等都是串

◇ 字符串 w 的长度，记为 $|w|$ ，是包含在 w 中字符的个数

举例 $|\varepsilon| = 0, |bbaba| = 5$

a^i 表示含有 i 个 a 的字符串



关于字符串的运算

✧ 连接 (*concatenation*)

设 x, y 为串, 且 $x = a_1 a_2 \dots a_m$, $y = b_1 b_2 \dots b_n$,
则 x 与 y 的连接

$$x y = a_1 a_2 \dots a_m b_1 b_2 \dots b_n$$

✧ 连接运算的性质

- $(x y) z = x (y z)$
- $\varepsilon x = x \varepsilon = x$
- $|x y| = |x| + |y|$

关于字符串的运算

✧ 其它 如 取头字符, 取尾部, 子串匹配 等

- 设 $\omega_1, \omega_2, \omega_3$ 是字母表 T 上的字符串, 称 ω_1 是字符串 $\omega_1\omega_2$ 的前缀, ω_2 是字符串 $\omega_1\omega_2$ 的后缀, 且 ω_2 是字符串 $\omega_1\omega_2\omega_3$ 的子串。

- 空串是任何字符串的前缀, 后缀及子串。

- 例:

abc的前缀 **a ab abc ϵ .**

后缀 **c bc abc ϵ .**

子串 **a b c ab bc abc ϵ ,**

即一个字符串可以看作是多个字符串的连接。

- 
- 字符串 ω 的逆用 $\tilde{\omega}$ 表示。是字符串 ω 的倒置。

$$\omega = b_1 b_2 \dots b_n$$

$$\tilde{\omega} = b_n b_{n-1} \dots b_2 b_1$$

- 空串 ε 的逆还是 ε

字母表的幂运算

✧ 幂运算 设 T 为字母表, n 为任意自然数,

定义 (1) $T^0 = \{ \varepsilon \}$

(2) 设 $x \in T^{n-1}$, $a \in T$, 则 $ax \in T^n$

(3) T^n 中的元素只能由 (1) 和 (2) 生成

✧ * 闭包 $T^* = T^0 \cup T^1 \cup T^2 \cup \dots$

✧ + 闭包 $T^+ = T^1 \cup T^2 \cup T^3 \cup \dots$

✧ $T^* = T^+ \cup \{ \varepsilon \}$, $T^+ = T^* - \{ \varepsilon \}$



闭包的物理意义

✧ **T**的星号闭包**T***：字母表**T**上的所有字符串和空串的集合。

✧ **T**的正闭包**T+**：字母表**T**上的所有字符串构成的集合。

$$T^* = T^+ \cup \{\varepsilon\}$$

✧ **举例** 设 $T = \{0, 1\}$ ，则

$$T^0 = \{\varepsilon\}, \quad T^1 = \{0, 1\},$$

$$T^2 = \{00, 01, 10, 11\}, \quad \dots$$

$$T^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, \dots\}$$

$$T^+ = \{0, 1, 00, 01, 10, 11, \dots\}$$



语言 (*LANGUAGES*)

✧ 概念 设 T 为字母表, 则任何集合 $L \subseteq T^*$ 是字母表 T 上的一个语言 (language)

✧ 举例

- 英文单词集 $\{..., \text{English}, ..., \text{words}, ...\}$
- C 语言程序集 $\{...\}$ 字母表?
- 汉语成语集 $\{..., \text{马到成功}, ...\}$
- 化学分子式集 $\{..., \text{H}_2\text{O}, ..., \text{NaCl}, ...\}$
- $\{ \text{any}, \text{任意} \}$

语言 (*LANGUAGES*)

✧ 举例：设 $T = \{a, b\}$

则 $L_1 = \{a^n b^n \mid n \geq 1\}$

$L_3 = \{b^k \mid k \text{ 是质数}\}$

$L_2 = \{\epsilon\}$ 只有一个空句子的语言

$L_4 = \{\} = \Phi$ 空语言

均为字母表 T 上的语言。

✧ 由语言的定义知语言是集合，对于集合的运算可应用于对于语言的计算。如并，交，补，差。

语言的基本运算

✧ 语言的积:

两个语言 L_1 和 L_2 的积 $L_1 L_2$ 是由 L_1 和 L_2 中的字符串连接所构成的字符串的集合。即 L_1 中所有字符串分别与 L_2 中的字符串连接得到的集合。

设 $T = \{a, b\}$, L_1 和 L_2 是 T 上的语言。

$L_1 = \{ab, ba\}$ $L_2 = \{aa, bb\}$

则 $L_1 L_2 = \{abaa, abbb, baaa, babb\}$

$L_2 L_1 = \{aaab, aaba, bbab, bbba\}$

■ $L_1 L_2 \neq L_2 L_1$ 语言的积不可交换。

语言的基本运算

✧ 语言的幂:

语言的幂可归纳定义如下:

$$L^0 = \{\epsilon\}$$

$$L^n = L \cdot L^{n-1} = L^{n-1} \cdot L \quad n \geq 1$$

上例中, $L_1 = \{ab, ba\}$ $L_2 = \{aa, bb\}$

$$L_1^2 = \{abab, abba, baab, baba\}$$

$$L_2^2 = \{aaaa, aabb, bbaa, bbbb\}$$



第二节 文法

- **定义**：所谓文法是用来定义语言的一个数学模型
- **表示语言的方法**：
 - 若语言 L 是有限集合，可用**列举法**
 - 若 L 是无限集合（集合中的每个元素有限长度），用其他方法。
 - 方法一：文法产生系统，由定义的文法规则产生出语言的每个句子
 - 方法二：机器识别系统：当一个字符串能被一个语言的识别系统接受，则这个字符串是该语言的一个句子，否则不属于该语言。



元语言

- 定义：描述语言的语言


例如：各种各样的程序设计语言

- 当人们要解释或讨论程序设计语言本身时，又需要一种语言，被讨论的语言叫做对象语言，即某种程序设计语言，讨论对象语言的语言称为元语言。

BNF（巴科斯范式）

BNF范式通常被作为讨论某种程序设计语言语法的元语言

- $\langle \text{数字} \rangle ::= 0|1|2|\dots|9$ $::=$ “定义为”
- $\langle \text{字母} \rangle ::= A|B|C|\dots|Z|a|b|\dots|z$
- $\langle \text{标识符} \rangle ::= \langle \text{字母} \rangle | \langle \text{标识符} \rangle \langle \text{字母} \rangle | \langle \text{标识符} \rangle \langle \text{数字} \rangle$
-
- 通过上述定义可知，所有以字母开头的，由字母和数字组成的字符串都是标识符。
- BNF定义了一种语言，其中标识符如上定义。
- BNF描述它所定义的语言，为元语言。

- 
- 例如：汉语语法中定义了句子的结构由主语、谓语、宾语组成。这里主谓宾只是描述了句子的结构，并不是句子。而按照这种结构组成的建立在汉字上的字符串就是句子。如他是学生。
 - 文法是一种元语言，一种方法，根据文法产生出语言的句子。

三、Chomsky 文法体系

■ 例如：

BNF $\langle \text{标识符} \rangle ::= \langle \text{字母} \rangle$

$\langle \text{标识符} \rangle ::= \langle \text{标识符} \rangle \langle \text{字母} \rangle$

$\langle \text{标识符} \rangle ::= \langle \text{标识符} \rangle \langle \text{数字} \rangle$

$\langle \text{字母} \rangle ::= a|b|\dots|z|A|B|\dots|Z$

$\langle \text{数字} \rangle ::= 0|1|\dots|9$

将 $::=$ 改为 \rightarrow 表示可被代替


用 I, L, D 分别表示标识符、字母、数字;



则上述表达式可以表示为

$$I \rightarrow L$$
$$I \rightarrow IL$$
$$I \rightarrow ID$$
$$L \rightarrow a|b|...|z$$
$$D \rightarrow 0|1|...9$$

这就是一个文法的生成式集合。

- 
- Chomsky 文法体系中，任何一种文法必须包含有两个不同的有限符号的集合，即非终结符集合 N 和终结符集合 T 。一个形式规则的有限集合 P （生成式集合），一个起始符 S 。
 - P 中的生成式是用来产生语言句子的规则，而句子则是仅由终结符组成的字符串。这些字符串必须从一个起始符 S 开始，不断使用 P 中的生成式而导出来。
 - 可见文法的核心是生成式的集合，它决定了语言中句子的产生。



文法的形式定义

- 文法 G 是一个四元组 $G=(N, T, P, S)$, 其中
 - N 非终结符的有限集合
 - T 终结符的有限集合 $N \cap T = \emptyset$
 - P 形式为 $\alpha \rightarrow \beta$ 的生成式的有限集合。
且 $\alpha \in (N \cup T)^* N^+ (N \cup T)^*$ $\beta \in (N \cup T)^*$
 - S 起始符 且 $S \in N$ 。

- 
- 将上例用文法表示

$$G=(N, T, P, S)$$

$$N = \{ I, L, D \}$$

$$T = \{ a, b, c, \dots z, 0, 1, \dots 9 \}$$

$$P = \{ I, L_a, \dots, D_0, \dots, D_9 \}$$

$$S = I$$

- 文法是语言的产生系统，研究怎样构造文法能产生出符合要求的句子。



课堂练习

- 字符串012的逆, 前缀, 后缀和子串
- 字母表 $T=\{0, 1\}$, $L1=\{00, 11\}$, $L2=\{01, 10\}$ 求
 - T^* 和 T^+
 - $L1L2$, $L2L1$, $L1^2$



四. 推导与句型

1、直接推导

设 $G = (N, T, P, S)$ 是文法，若 $A \rightarrow \beta$ 是 P 中的生成式， α 和 γ 是 $(N \cup T)^*$ 中的字符串，则有 $\alpha A \gamma \Rightarrow \alpha \beta \gamma$ 称 $\alpha A \gamma$ 直接推导出 $\alpha \beta \gamma$ ，或说 $\alpha \beta \gamma$ 是 $\alpha A \gamma$ 的直接推导。



2、推导序列

- 设 $G = (N, T, P, S)$ 是文法， α 、 α_0 、 $\alpha_1 \dots \alpha_n$ 、 α' 都是 $(N \cup T)^*$ 中的字符串，且 $\alpha = \alpha_0$ 、 $\alpha' = \alpha_n$ ，其中 α_i 直接推导出 α_{i+1} ($0 \leq i \leq n$)，则称序列 $\alpha_0 \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \dots \Rightarrow \alpha_n$ 是长度为 n 的推导序列，而 $\alpha = \alpha_0$ 是长度为 0 的推导序列。
- 对 α 推导出 α' 记为 $\alpha \xrightarrow{*}_G \alpha'$ ，若推导序列长度大于 0，则记为 $\alpha \xrightarrow{+}_G \alpha'$ 。
- 推导序列的每一步，都产生一个字符串，这些字符串一般称为句型。

3、句型 and 句子

■ 句型

字符串 α 是文法 G 的句型，当且仅当

$$S \xrightarrow{*}_G \alpha, \text{ 且 } \alpha \in (N \cup T)^*.$$

■ 句子

ω 是 G 的句子，当且仅当 $S \xrightarrow{*}_G \omega$, 且 $\omega \in T^*$ 。
(ω 是由终结符组成的字符串)

例： $I \Rightarrow L \Rightarrow a$

$I \Rightarrow IL \Rightarrow LL \Rightarrow zL \Rightarrow zb$

■ 句型包含句子



4. 文法产生的语言

由文法G产生的语言记为L(G)。

$$L(G) = \{\omega | \omega \in T^* \text{ 且 } S \xrightarrow[G]{*} \omega\}$$

或：

L(G)中的一个字符串，必是由终结符组成的，并且是从起始符S推导出来的。

第三节 Chomsky 文法体系分类

- 文法 $G = (N, T, P, S)$; $P: \alpha \rightarrow \beta$
其中 $\alpha \in (N \cup T)^* N^+ (N \cup T)^*$
 $\beta \in (N \cup T)^*$ 属于 Chomsky 文法体系
- 该体系对生成式的形式做了一些规定，分为四类，即 0 型、1 型、2 型、3 型文法
- 0 型文法：无限制文法
对应的语言：递归可枚举语言，与图灵机等价。

1型文法

- 也称上下文有关文法 (CSG : Context-sensitive Grammar)

生成式的形式为 $\alpha \rightarrow \beta$,

其中 $|\alpha| \leq |\beta|$, $\beta \in (N \cup T)^+$,

$\alpha \in (N \cup T)^* N^+ (N \cup T)^*$

- 对应的语言：上下文有关语言 (CSL : Context-sensitive Language)
- 若不考虑 ε , 与线性有界自动机 (LBA, Linear Bounded Automaton) 等价。



2型文法

- 也称上下文无关文法 (CFG : Context-free Grammar)

$A \rightarrow \beta,$

$A \in N, \text{ 且 } \beta \in (N \cup T)^*$

- 对应的语言：上下文无关语言 (CFL : Context-free Language)。
- 对应的自动机：下推自动机 (PDA : Pushdown Automaton)。



3型文法

也称正则文法

- 右线性文法（Right-linear Grammar）：
 $A \rightarrow \omega B$ 或 $A \rightarrow \omega$
 $A, B \in N, \omega \in T^*$ 。
- 左线性文法（Left-linear Grammar）：
 $A \rightarrow B\omega$ 或 $A \rightarrow \omega$
 $A, B \in N, \omega \in T^*$ 。
- 对应的语言：正则语言
- 对应的自动机：有限自动机（Finite Automaton）。



例1 :

$G = (\{A, B, C\}, \{a, b, c\}, P, A)$

$P: A \rightarrow abc \quad A \rightarrow aBbc \quad Bb \rightarrow bB \quad Bc \rightarrow Cbcc$
 $bC \rightarrow Cb \quad aC \rightarrow aaB \quad aC \rightarrow aa。$

1型文法, 其定义的 $L = \{a^n b^n c^n \mid n \geq 1\}$

- $A \Rightarrow abc$
- $A \Rightarrow aBbc \Rightarrow abBc \Rightarrow abCbcc \Rightarrow aCbbcc$
 $\Rightarrow aabbcc$
 $\Rightarrow aaBbbcc$



例2 :

$G = (\{S, B, C\}, \{a, b\}, P, S)$

$P: S \rightarrow aC ; S \rightarrow bB ; B \rightarrow aS ; B \rightarrow bBB \quad B \rightarrow a ;$
 $C \rightarrow bS ; C \rightarrow aCC ; C \rightarrow b$

是2型文法

- $S \Rightarrow aC \Rightarrow ab$
- $S \Rightarrow aC \Rightarrow aaCC$
- $S \Rightarrow aC \Rightarrow abS \Rightarrow abaC \Rightarrow ababS \Rightarrow ababaC \Rightarrow ababab$
- $S \Rightarrow bB \Rightarrow bbBB \Rightarrow bbaSB \Rightarrow bbaaCB \Rightarrow bbaabB \Rightarrow bbaaba$



例3：

$G = (\{A, B, C\}, \{a, b, c\}, P, A)$

$P: A \rightarrow Ba ; A \rightarrow c ; B \rightarrow Cb ; C \rightarrow c$

- 左线性文法
- $L = \{c, cba\}$ 正则语言
- 注意：已知语言求文法，文法不是唯一的，即可以有不同的表达方法。



四类文法之间的关系

- 只是对生成式形式加以限制
- 0型 无限制
- 1型 不允许 $A \rightarrow \epsilon$ 形式
- 2型
- 3型 属于2型
- 不含 $A \rightarrow \epsilon$ 的2型、3型属于1型，1型、2型、3型均属于0型。



课堂作业

例 1 构造右线性文法, 识别语言 $L = \{a^{3n+1} \mid n \geq 0\}$ 。

例 2 构造上下文无关文法, 能够产生 $L = \{\omega \mid \omega \in \{a, b\}^* \text{ 且 } \omega \text{ 中 } a \text{ 的个数是 } b \text{ 的两倍}\}$ 。

例 4 找出由下列各组生成式产生的语言(起始符为 S):

(1) $S \rightarrow SaS, S \rightarrow b$;



■ 作业: P37 4, 5, 6, 7 题