

第二章 程序设计语言

学习内容

语言及其文法

字母表

- ▶ 字母表 Σ 是一个非空有穷符号集合
符号：字母、数、标点符号 ...

- ▶ 以下是不同的字母表：

$\{a, b, c, d\}$

$\{a, b, c, \dots, z\}$

$\{0, 1\}$

ASCII字母表

字母表上的运算

- 字母表 Σ_1 与 Σ_2 的乘积:

$$\Sigma_1\Sigma_2=\{ab|a\in\Sigma_1, b\in\Sigma_2\}$$

例: $\Sigma_1=\{0,1\}$, $\Sigma_2=\{a,b\}$, $\Sigma_1\Sigma_2=\{0a,0b,1a,1b\}$

- 字母表 Σ 的 n 次幂递归地定义为:

(1) $\Sigma^0=\{\epsilon\}$

(2) $\Sigma^n=\Sigma^{n-1}\Sigma \quad n\geq 1$

例: $\Sigma_1^3=\{000,001,010,011,100,101,110,111\}$

字母表上的运算

- 字母表 Σ 的**正闭包**(Positive Closure):

$$\Sigma^+ = \Sigma \cup \Sigma^2 \cup \Sigma^3 \cup \Sigma^4 \cup \dots$$

例: $\{0,1\}^+ = \{0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, \dots\}$

- 字母表 Σ 的**克林闭包**(Kleene Closure)为:

$$\Sigma^* = \Sigma^0 \cup \Sigma^+ = \Sigma^0 \cup \Sigma \cup \Sigma^2 \cup \Sigma^3 \cup \dots$$

例: $\{0,1\}^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, \dots\}$

串

- **串：** 是字母表中符号的一个有穷序列
 Σ 是一个字母表, $\forall x \in \Sigma^*$, x 是 Σ 上的一个串。
- 串 s 的长度: s 中符号的个数, 通常记作 $|s|$, 例: $|aab|=3$
- 空串是长度为0的串, 用 ε (*epsilon*) 表示, $|\varepsilon|=0$

串上的运算

- 连接

- ▶ 串 x 和 y 的连接：把 y 附加到 x 后面而形成的串，记作 xy
- ▶ 例如，如果 $x=dog$ 且 $y=house$ ， $xy=doghouse$
- ▶ 空串是连接运算的单位元，对于任何串 s 都有， $\varepsilon s = s\varepsilon = s$

设 x, y, z 是三个字符串，如果 $x=yz$ ，则称 y 是 x 的前缀， z 是 x 的后缀

串上的运算

- 幂

- 串 s 的 n 次幂:

- (1) $s^0 = \varepsilon$;

- (2) $s^n = s^{n-1}s$ 。

设 x, y, w 是三个字符串，如果 $w = xy$ ，则称 y 是 w 的子串

语言

设 Σ 是一个字母表, $\forall L \subseteq \Sigma^*$, L 称为字母表 Σ 上的一个语言

$\forall x \in L$, x 叫做 L 的一个句子

- 例: 字母表 $\{0, 1\}$ 上的语言

$\{0, 1\}$

$\{00, 11\}$

$\{0, 1, 00, 11\}$

$\{0, 1, 00, 11, 01, 10\}$

$\{00, 11\}^*$

$\{01, 10\}^*$

语言的运算

- Σ_1 和 Σ_2 是字母表, $L_1 \subseteq \Sigma_1^*$, $L_2 \subseteq \Sigma_2^*$, 语言 L_1 与 L_2 的乘积是字母表 $\Sigma_1 \cup \Sigma_2$ 上的一个语言, 该语言定义为:

$$L_1 L_2 = \{xy \mid x \in L_1, y \in L_2\}$$

- 字母表 Σ , $\forall L \in \Sigma^*$, L 的 n 次幂是一个语言, 该语言定义为:
 - (1) 当 $n=0$ 时, $L^n = \{\epsilon\}$;
 - (2) 当 $n \geq 1$ 时, $L^n = L^{n-1}L$ 。

语言的运算

- ▶ L 的正闭包 L^+ 是一个语言，该语言定义为：
$$L^+ = L \cup L^2 \cup L^3 \cup L^4 \cup \dots$$
- ▶ L 的克林闭包 L^* 是一个语言，该语言定义为：
$$L^* = L^0 \cup L \cup L^2 \cup L^3 \cup L^4 \cup \dots$$

文法定义

文法是用于描述语言的语法结构的形式规则。

任何一种语言都有它自己的文法，不管它是机器语言还是自然语言。

自然语言的文法：主 谓 宾

机器语言也有描述它语言构成的特定文法

文法可以定义为一个四元组。

文法定义

四元组 (V_T, V_N, S, P)

1. 一个终结符号集合 V_T
2. 一个非终结符号集合 V_N
3. 一个产生式集合 P ，定义语法范畴
产生式: $\alpha \rightarrow \beta$ (α 定义为 β)
4. 一个特定的非终结符——开始符号 S

$$V_T \cap V_N = \Phi$$

$V_T \cup V_N$: 文法符号集

产生式: 描述了将终结符和非终结符组合成串的方法

$\alpha \in (V_T \cup V_N)^+$: 称为产生式的头或左部, α 中至少包含 V_N 中的一个元素

$\beta \in (V_T \cup V_N)^*$: 称为产生式的体或右部

符号约定

终结符:

- 字母表中排在前面的小写字母, 如 a 、 b 、 c
- 运算符, 如 $+$ 、 $*$ 等
- 标点符号, 如括号、逗号等
- 数字0、1、...、9
- 粗体字符串, 如 id 、 if 等

非终结符:

- 字母表中排在前面的大写字母, 如 A 、 B 、 C
- 字母 S 。通常表示开始符号
- 小写、斜体的名字, 如 expr 、 stmt 等
- 代表程序构造的大写字母。如 E (表达式)、 T (项)和 F (因子)

符号约定

文法符号:

- 字母表中排在后面的大写字母（如 **X** 、 **Y** 、 **Z** ）表示文法符号（即终结符或非终结符）

终结符号串:

- 字母表中排在后面的小写字母（主要是 **u** 、 **v** 、 **\dots** 、 **z** ）表示终结符号串（包括空串）

文法符号串:

- 小写希腊字母，如 **α** 、 **β** 、 **γ** ，表示文法符号串（包括空串）

除非特别说明，第一个产生式的左部就是开始符号

产生式的简写

对一组有相同左部的 α 产生式 $\alpha \rightarrow \beta_1, \alpha \rightarrow \beta_2, \dots, \alpha \rightarrow \beta_n$

可以简记为:

$$\alpha \rightarrow \beta_1 \mid \beta_2 \mid \dots \mid \beta_n$$

读作:

α 定义为 β_1 , 或者 β_2 , ..., 或者 β_n

$\beta_1, \beta_2, \dots, \beta_n$ 称为 α 的候选式

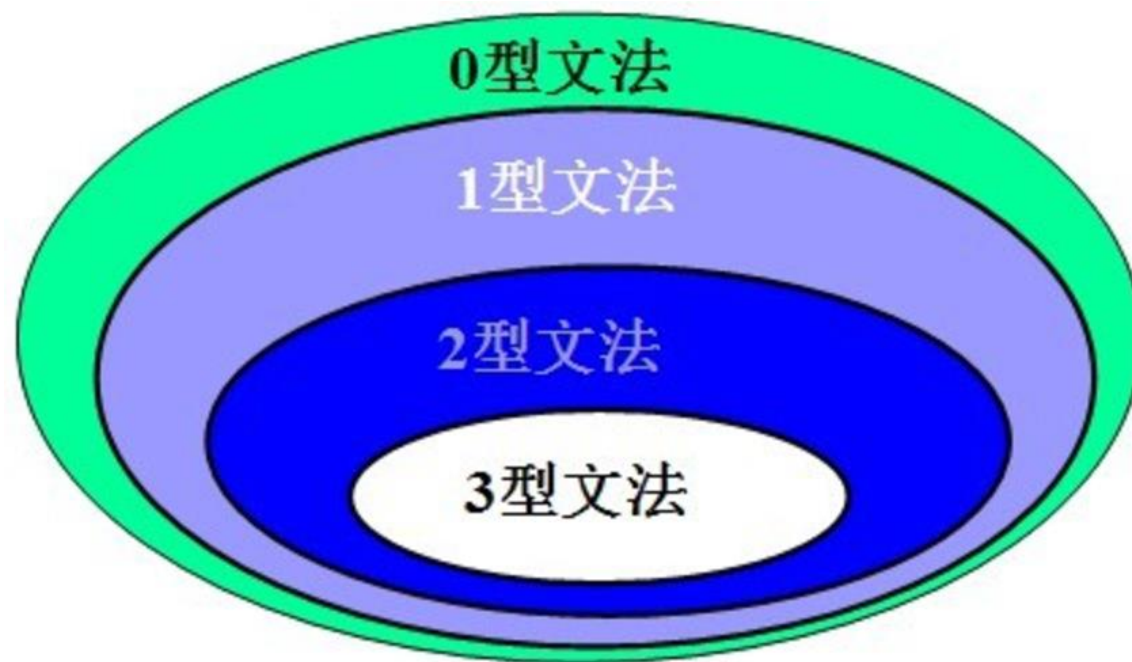
例:

$\begin{aligned} E &\rightarrow E + E \\ E &\rightarrow E * E \\ E &\rightarrow (E) \\ E &\rightarrow \text{id} \end{aligned}$	\Longrightarrow	$E \rightarrow E + E \mid E * E \mid (E) \mid \text{id}$
--	-------------------	--

文法的 4 种类型

1956年，Chomsky建立形式语言的描述。

通过对产生式的施加不同的限制，Chomsky把文法分为4种类型



文法的 4 种类型

- 2型文法(上下文无关文法, context-free grammar. CFG)
 - $\forall \alpha \rightarrow \beta$, 都有 $\alpha \in V_N$, $\beta \in (V_N \cup V_T)^*$
 - 产生式的形式描述: $A \rightarrow \beta$ ($A \in V_N$)

即: β 取代 A 时, 与 A 所处的上下文无关。
- 产生的语言称“上下文无关语言”
- 例如: $S \rightarrow 01$ $S \rightarrow 0S1$

文法的 4 种类型

- 3型文法(regular grammar, RG): 也称正则文法
- 每个产生式均为 “ $A \rightarrow aB$ ”或 “ $A \rightarrow a$ ” —— 右线性
or “ $A \rightarrow Ba$ ”或 “ $A \rightarrow a$ ” —— 左线性,
($A、B \in V_N, a \in V_T^*$)
- 产生的语言称 “正则语言”
- 例如: $S \rightarrow 0A \mid 0$

$$A \rightarrow 1B \mid B$$

$$B \rightarrow 1 \mid 0$$

产生式设计练习

- 首先是“人会做”

- 我们自己先把语法概念“什么模样”搞清楚

- 然后是“让计算机做”——符号化

- 为这个语法概念起个名字，“模样”中的其他语法概念、单词也都有相应的名字
 - 将语法概念放在产生式左部
“模样”放在产生式右部
都是用名字替换掉语法概念和单词——

产生式设计练习

- **while** (expression) statement

对应的产生式

$stmt \rightarrow \mathbf{while} (expr) stmt$

- **for** (expression₁; expression₂; expression₃)
statement

对应的产生式

$stmt \rightarrow \mathbf{for} (expr ; expr ; expr) stmt$

- **int(float, double, char)** id₁, id₂, ...;

$type \rightarrow \mathbf{int} \mid \mathbf{float} \mid \mathbf{double} \mid \mathbf{char}$

$idlist \rightarrow idlist , \mathbf{id} \mid \mathbf{id}$

$decl \rightarrow type idlist ;$

学习内容

语言及其文法