



链滴

[编译原理] 学习笔记（二）——文法和语言

作者: [openshell](#)

原文链接: <https://ld246.com/article/1479712779009>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

一、对程序设计语言的描述从语法、语义和语用三个因素考虑:

- a) 语法: 对语言结构的定义;
- b) 语义: 语言的含义;
- c) 语用: 从使用的角度描述语言。

形式语言理论是编译的理论基础。

二、字母表: 元素的非空有穷集合;

符号/字符: 字母表中的元素;

符号串: 符号的有穷序列。

三、符号串运算:

- a) 符号串的连接: $\varepsilon x = x\varepsilon = x$;
- b) 集合的乘积: $AB = \{xy | x \in A, y \in B\}$; $\{\varepsilon\}A = A\{\varepsilon\} = A$;
- c) 符号串的幂运算: $x = abc, x^2 = abcabc$;
- d) 集合的幂运算
- e) 正闭包 A^+ 与闭包 A^* : $A^* = \{\varepsilon\} \cup A^+$

四、形式语言: 字母表上按照某种规则构成的所有符号串的集合, 其不考虑语义。描述形式语言的式有两种:

- a) 枚举——当语言为有穷集合时;
- b) 文法——描述了无穷集合的语言。

五、文法: $G = (V_n, V_t, P, S)$

a) 规则 P : 也称为产生式, 是一个符号与一个符号串的有序对 (A, β)

$A \rightarrow \beta$

- i. 一组规则定义了一个语言的语法结构;
- ii. 规则中出现的符号分为终结符号和非终结符号

- b) V_n 为非终结符(non-terminate);
- c) V_t 为终结符(terminate);
- d) S 为非终结符号, 称为文法的开始符号/识别符号, 至少要在一条规则的左部出现。

六、推导: 推导的依据是规则

- a) 直接推导: 仅使用一次规则;
- b) 推导: 至少使用一次规则;
- c) 广义推导: 经过0步或若干步的推导。
- d) 最右推导又称规范推导, 推导出的句型为规范举行; 与之对应的最左规约为规范规约。

七、句型、句子和语言:

- a) 句型: $S \Rightarrow^* x, x \in (V_n \cup V_t)^*$, 其中 $S \Rightarrow^* x$ 为广义推导。
- b) 句子: $S \Rightarrow^* x, x \in V_t^*$, 其中 $S \Rightarrow^* x$ 为广义推导, x 必须是终结符的闭包(可为 ε)。
- c) 语言: $L(G[S]) = \{x | S \Rightarrow^+ x \text{ 且 } x \text{ 属于 } V_t^*\}$, 其中 $S \Rightarrow^+ x$ 为推导, 至少使用一次规则。

八、递归:

- a) 递归规则: 在规则的左部和右部具有相同非终结符的规则;

- i. 规则左递归: $A \rightarrow A\dots$;
- ii. 规则右递归: $A \rightarrow \dots A$;
- iii. 规则递归: $A \rightarrow \dots A\dots$;

b) 文法递归: 对文法中的任一非终结符, 若能建立一个推导过程使得右部再次出现该非终结符, 则法是递归的。如: $U \rightarrow Vx$, $V \rightarrow Uy|z$, 虽然这两个规则都不是递归规则, 但组成的文法是递归文法 $U \rightarrow x \rightarrow Uyx$ 。所以含有递归规则的文法一定是递归文法, 而递归文法不一定含有递归规则。

九、 短语、直接短语和句柄: 都是针对某一句型的

a) 短语: $S \Rightarrow^* \alpha A \delta$ 且 $A \Rightarrow^+ \beta$, 则称 β 是相对于非终结符 A 的句型 $\alpha A \delta$ 的短语; 对应语法树中的子树念。

b) 直接短语: 其中 $A \Rightarrow \beta$ 为直接推导; 对应语法树中的简单子树。每个直接短语都是某规则的右部。

c) 句柄: 是直接短语 (即某规则的右部), 且具有最左性; 对应简单子树中最左的一棵。

十、 文法的二义性: 如果一个文法存在某个句子对应两棵不同的语法树 (包含两个或两个以上的最右最左) 推导 (规约), 则该文法是二义性的, 可以利用文法之间的等价性来消除二义性。

a) 不改变文法中原有的语法规则, 进增加一些语法的非形式定义, 如优先级;

b) 构造一个等价的无二义性文法。

十一、 文法的分类:

a) 0型文法/无限制文法: $\alpha \rightarrow \beta$, 其中 $\alpha \in (V_n \cup V_t)^*$ 且至少含有一个非终结符, $\beta \in (V_n \cup V_t)^*$ 。

b) 1型文法/上下文有关文法: $\alpha A \beta \rightarrow \alpha u \beta$, 其中 $A \in V_n$, $\alpha, \beta \in (V_n \cup V_t)^*$, $u \in (V_n \cup V_t)^+$ 。

c) 2型文法/上下文无关文法: $A \rightarrow \beta$, 其中 $A \in V_n$, $\beta \in (V_n \cup V_t)^*$ 。常用于句法分析。

d) 3型文法/正规文法: 常用于词法分析

i. 右线性文法: 只能对推出式的右边展开, $A \rightarrow \alpha B | \alpha$, $A, B \in V_n$, $\alpha \in V_t^*$ 。

ii. 左线性文法: 只能对推出式的左边展开, $A \rightarrow B \alpha | \alpha$, $A, B \in V_n$, $\alpha \in V_t^*$ 。