第二章 - 语言及其文法

字母表

字母表是一个有穷符号集合(符号：字母、数字、标点符号等)

字母表运算

乘积：将两个字母表的元素进行依次结合

例如：A={0, 1} B={a, b} ，AB={0a, 0b, 1a, 1b}

n次幂：字母表对自身的n次乘积后的集合

幂次为0：表示空集，使用符号ε表示

幂次为n：表示长度为n的符号串集合

例如：{0, 1} ^ 3 ={0, 1} {0, 1} {0, 1}

正闭包：长度为正数的符号串集合，从1次幂到n次幂运算结果的并集

例如：A ^ + = A ∪ A ^ 2 ∪ A ^ 3 ∪ …

克林闭包：在正闭包的基础上加入一个ε，即构成

例如：A ^ * = A ^ 0 ∪ A ∪ A ^ 2 ∪ A ^ 3 ∪ …

串

串是字母表中符号的一个有穷序列。

长度：表示符号符号个数(记作 |串|)，如：|atao| = 4；空串表示长度为0 的串(也是使用 ε 表示)

串运算

连接：将两个串拼在一起形式一个新串

例如：x = atao，y = Firebasky，则 z = xy = ataoFirebasky

> x是z的前缀，y是z的后缀

空串在连接运算中可以表示为单位元(有点像线代中矩阵相乘中的单位矩阵)，即εs = sε = s

幂：将幂次个串连接起来

例如：s ^ 1 = (s ^ 0)s = εs = s，或者s = a, s ^ 2 = aa, s ^ 3 = aaa ....

文法

在学习英文时，我们需要将一个句子转换成为主语、名词短语、动词短语等(如果有考研学习长难句应该深有体会)，接着名词短语可以继续转换为形容词和名词，动词短语也是如此。单词就是语言的基本符号(句子的最小单位)，而前面所提到的动名词短语是语法成分。

文法形式化定义

G = (VT, VN, P, S) 【这里的T和N是写在左下角的】

VT：终极符集合，例如：VT = {atao, Firebasky, m3w}

> 终极符是文法所定义的语言的基本符号，也可以称为token

表示的符号

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

VN：非终极符集合，例如：VN = {<句子>, <动词短语>, <名词短语>, ...}

> 非终极符是用来表示语法成分的符号，也可以称为语法变量

表示的符号

1. 字母表中排在前面的大写字母，如A、B、 C
2. 字母S。通常表示开始符号
3. 小写、斜体的名字，如 expr、stmt等

4. 代表程序构造的大写字母。如E(表达式)、T(项)和F(因子)

> ## VT ∩ VN = 空集；VT ∪ VN ：文法符号集

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）
字母表中排在后面的小写字母（主要是u、v、. . . 、z）,表示终结符号串（包括空串）
小写希腊字母，如α、β、γ，表示文法符号串（包括空串）

P：产生式集合，例如：P = { <句子> → <主语> <动词短词> <名词短语> }

> 产生式描述将终结符和非终极符组合成串的方法，一般形式：α→β (α定义为β)
> α ∈ (VT ∪ VN) ^ +，且α中至少包含VN中的一个元素：称为产生式的头或左部
> β ∈ (VT ∪ VN)：称为产生式的体或右部

S：开始符号，例如：S = <句子>

> 开始符号表示该文法中最大的语法成分，除非特别说明，第一个产生式的左部就是开始符号

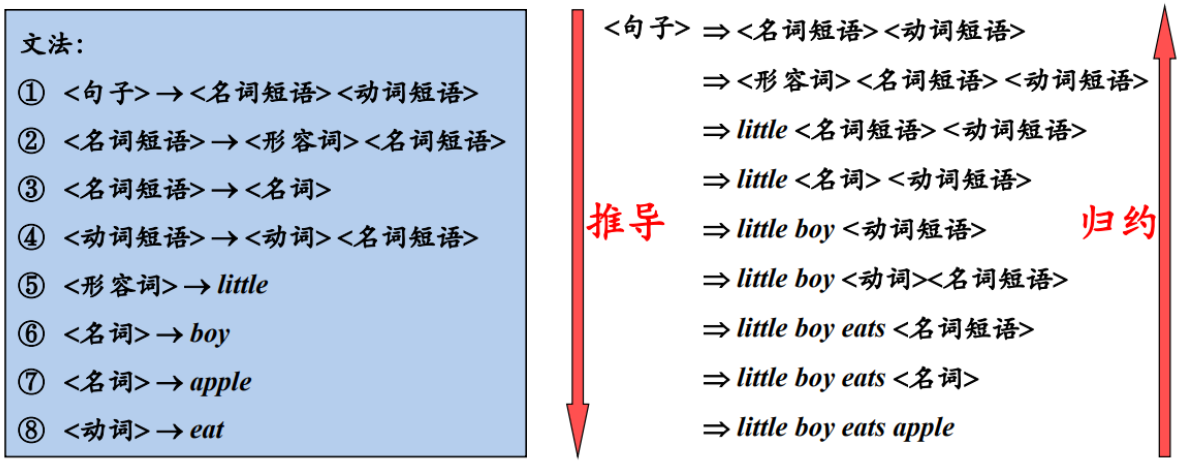
表达式：

```
G =( { id, +, *, (, ) }, {E}, P, E )
P ={ E → E + E, E → E * E, E → ( E ), E → id }

约定：不引起歧义的前提下，可以只写产生式
简写为
G : E → E + E, E → E * E, E → ( E ), E → id

当一组相同左部的E产生式时，简写为
E → E + E | E * E | ( E ) | id (读作:E定义为E + E,或者E * E,或者( E ),或者id
E + E, E * E, ( E ), id称为E的候选式
```

推导和归约

给定文法G=(VT , VN , P , S)，如果 α→β ∈ P，那么可以将符号串γαδ中的α替换为β，也就是说，将γαδ重写为γβδ，记作 γαδ ➡️ γβδ。此时，称文法中的符号串 γαδ 直接推导出 γβδ。简而言之，就是用产生式的右部替换产生式的左部。

例



➡️+表示经过正数步推导
➡️* 表示经过若干(可以是0)步推导

句型和句子

如果 S ➡️* α，α ∈ (VT∪VN)*，则称α是G的一个句型

> 一个句型中既可以包含终结符，又可以包含非终结符，也可能是空串

如果 S ➡️* w，w ∈ VT*，则称w是G的一个句子

> 句子是不包含非终结符的句型

例

<句子>⇒ <名词短语> <动词短语>
⇒ <形容词> <名词短语> <动词短语>
⇒ *little* <名词短语> <动词短语>
⇒ *little* <名词> <动词短语>
⇒ *little boy* <动词短语>
⇒ *little boy* <动词><名词短语>
⇒ *little boy eats* <名词短语>
⇒ *little boy eats* <名词>
句子 ⇒ *little boy eats apple*

句型

语言的形式化定义

由文法G的开始符号S推导出的所有句子构成的集合称为文法G生成的语言，记为L(G)。即 L(G) = { w | S ➡️* w, w ∈ VT }

文法的分类

文法分类体系

0型文法：无限制文法/短语结构文法

∀α → β∈P， α中至少包含1个非终结符

0型语言：由0型文法G生成的语言L(G)

1型文法：上下文有关文法(CSG)

∀α → β∈P，｜α｜≤｜β｜

产生式的一般形式： α1Aα2 → α1βα2 (β ≠ ε)

> CSG中不包含ε-产生式：原因是如果β为ε，则｜β｜=0，但是α至少要包含一个非终结符，所以｜α｜≥ 1，前后矛盾不满足规则

1型语言(上下文有关语言)：由上下文有关文法(1型文法)G生成的语言L(G)

2型文法：上下文无关文法(CFG)

∀α → β∈P，α ∈ VN

产生式的一般形式：A→β

2型语言(上下文无关语言)：由上下文无关文法(2型文法)G生成的语言L(G)

3型文法：正则文法(RG)

右线性文法： A→wB 或 A→w (w是终结符，B是非终结符)

左线性文法： A→Bw 或 A→w (w是终结符，B是非终结符)

3型语言(正则语言)：由正则文法(3型文法)G生成的语言L(G)

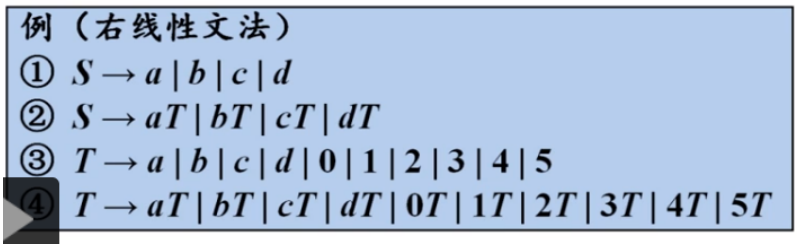
> 四种文法之间的关系是从上往下逐级包含的

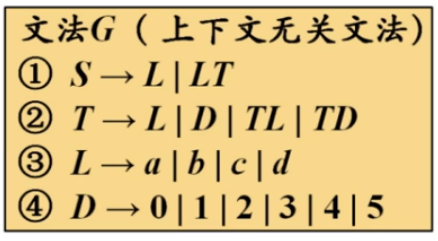
课后作业

1.请写出无符号整数和浮点数的文法

```
无符号整数文法
S → C|AB
A → 1|2|3|4|5|6|7|8|9
B → AB|BC|A|C
C → 0

浮点数文法
S → B.C|AC.C
A → 1|2|3|4|5|6|7|8|9
B → 0
C → AC|BC|A|B
```

2.写出于下列等价的左线性文法





```
S → a|b|c|d
S → Sa|Sb|Sc|Sd|S0|S1|S2|S3|S4|S5
```