编译原理与设计

北京理工大学 计算机学院

文法：本节展开思路

从文法和语言的直观概念

↓

文法、语言的运算和形式定义

表示方法

类型

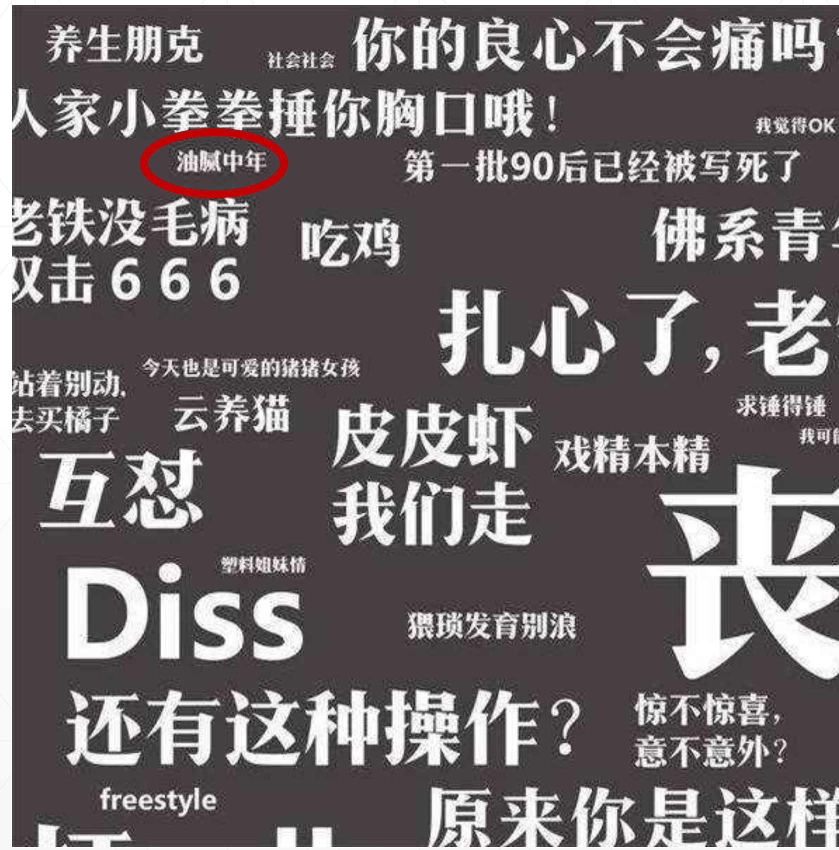
二义性问题

文法：自然语言的问题

确认过眼神~
巨婴~
锦鲤~
年度关键词
教科书式
书式~
你的关键词是什么?
吃鸡
官宣~
店小二~
杠精~
佛系人生~

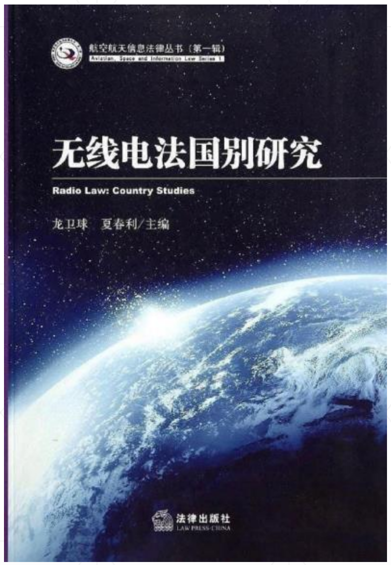
qiou
注释：
①穷到吃土
②不但丑，还穷到吃土
③穷丑土不至一点点

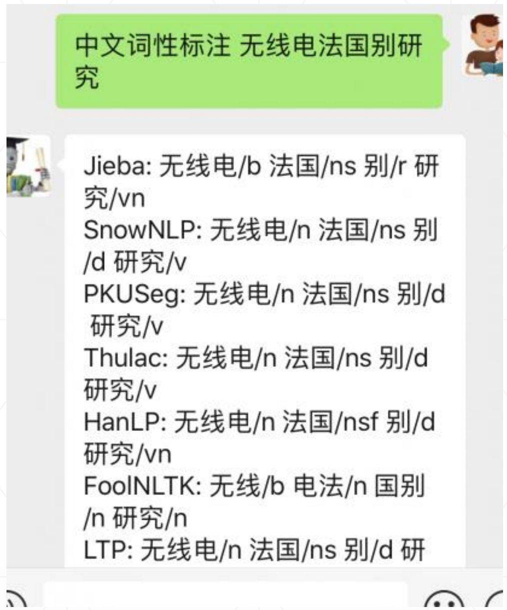
扎心了，老铁
我太难了
2017
2019
扎心了，老铁！
我太
了
"奔三"只是一个「扎心了」的话题



养生朋克
社会社会
你的良心不会痛吗？
人家小拳拳捶你胸口哦！
我觉得OK
油腻中年
第一批90后已经被写死了
老铁没毛病
吃鸡
佛系青
双击666
扎心了，老
今天也是可爱的猪猪女孩
站着别动，去买橘子
云养猫
皮皮虾我们走
求锤得锤
戏精本精
互怼
丧
塑料姐妹情
Diss
猥琐发育别浪
还有这种操作？
惊不惊喜，意不意外？
freestyle
原来你是这样

文法：自然语言的问题





- 开会的时候，有人抽烟，老板咬牙道，“抽烟的都掐死。”
- 我背有点驼，麻麻说“你的背得背背背背佳”

文法：自然语言的问题

- He is not a grave man until he is a grave man.
- Time flies like an arrow, fruit flies like a banana.
- Flying planes can be dangerous.
- I'm glad I'm a man, and so is Lola.
- John saw the man on the mountain with a telescope.
- …

文法：语言的语法和语义

- 语言要素
  - 语法：语言的描述规则
  - 语义：语言的含义

语法是一种媒介，语义以语法为媒介来予以说明。

语言是由单词按一定规则（文法）组成句子来表达特定意思。故对语言的分析集中于对句子的分析。而句子的分析依据语言的文法规则。

文法：语言的语法和语义

- 例：设有语句“小八哥吃大花生”。

  设有汉语语法规则的一个子集：

〈句子〉→〈主语〉〈谓语〉
〈主语〉→〈形容词〉〈名词〉
〈谓语〉→〈动词〉〈宾语〉
〈宾语〉→〈形容词〉〈名词〉
〈形容词〉→ 小 | 大
〈名词〉→ 八哥 | 花生
〈动词〉→ 吃

文法：语言的语法和语义

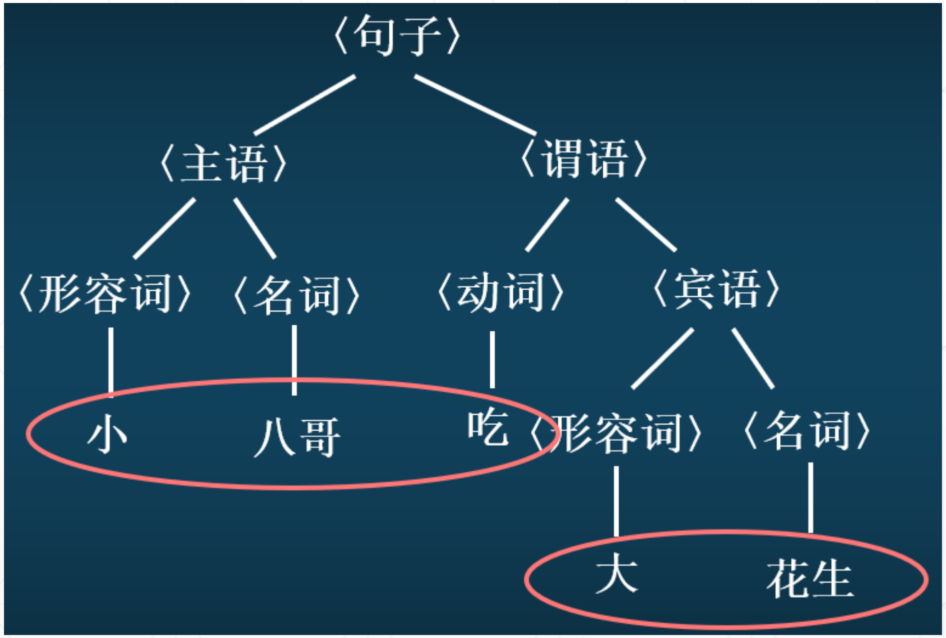
- 巴科斯-诺尔范式表示法，简称BNF。
  - 〈〉：表示语法成分；
  - → /::= ：表示“定义为”或“由…组合成”；
  - | ：“或”，具有相同左部的产生规则用|分开

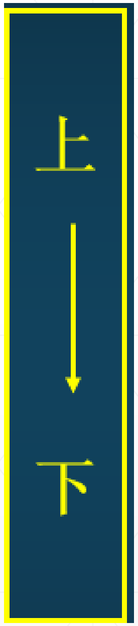
元语言符号

元语言:描述另一个语言的语言。

文法：语言的语法和语义

- 语句“小八哥吃大花生”分析语法树





文法：语言的语法和语义

- 句子的推导

<句子> ⇒ <主语> <谓语>

⇒ <形容词> <名词> <谓语>

⇒ <小> <名词> <谓语>

⇒ ……

⇒ 小八哥吃大<名词>

⇒ 小八哥吃大花生

文法：语言的语法和语义

- 句子的推导

<句子> ⇒ <主语> <谓语>

⇒ <形容词> <名词> <谓语>

⇒ <小> <名词> <谓语>

⇒ <小> <八哥> <谓语>

⇒ <小> <八哥> <动> <宾语>

⇒小八哥吃<宾语>

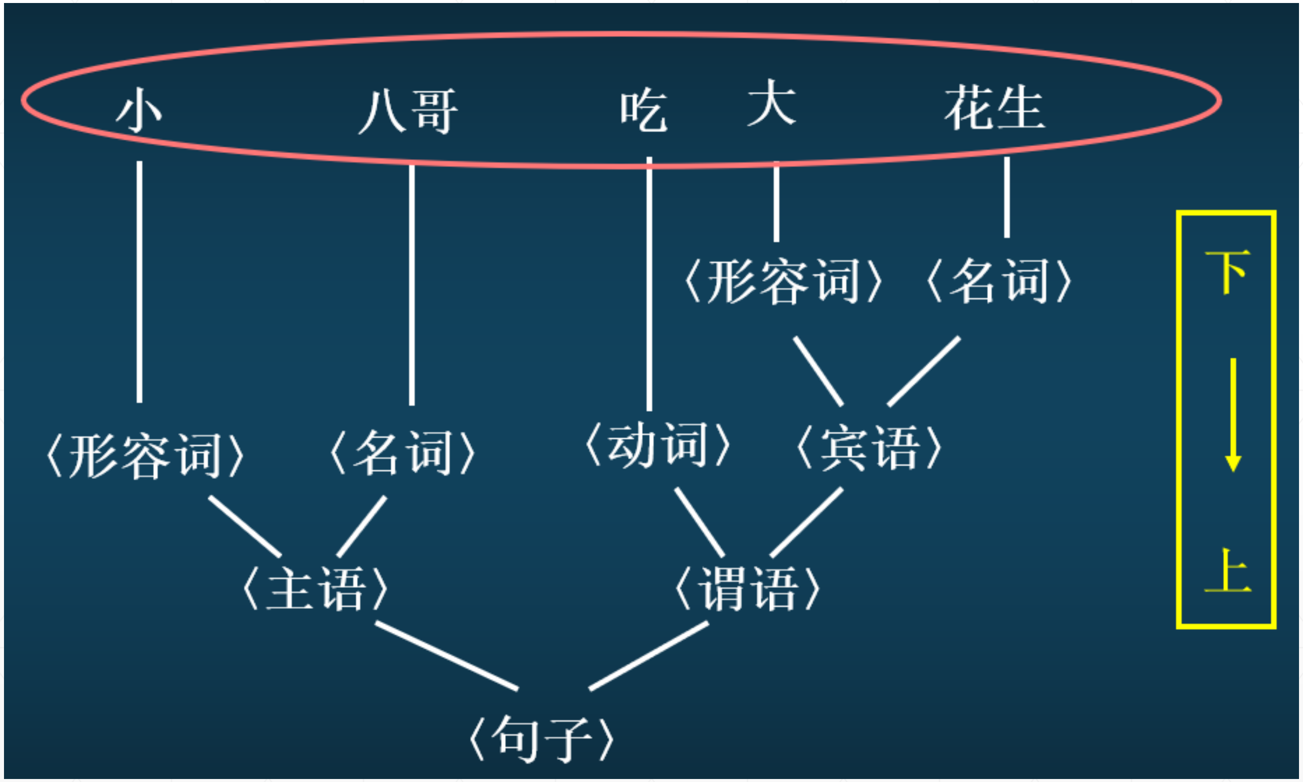
⇒ 小八哥吃<形容词> <名词>

⇒小八哥吃大<名词>

⇒小八哥吃大花生

文法：语言的语法和语义

- 句子的规约



文法：字符和字符串

- 字符、字符串

任何一种语言，都是由该语言的基本字符所组成的字符串的集合。

例如，程序设计语言的基本字符集是由字母、数字、运算符等其它符号组成，则任何程序都是由这些基本字符组成的序列。

- 字母表

字母表是元素的非空有穷集合。字母表中的元素称为符号，因此字母表也称为符号集。

文法：字符和字符串

- 字母表
  - 用希腊字母∑或大写英文字母等表示字母表
  - 用集合元素表示形式枚举字母表中的符号
- 例如
  - **汉语**：字母表中包括汉字、数字和标点符号等
  - **机器语言**：字母表是$\Sigma=\{0,1\}$
  - **C语言**：一切可打印字符的集合

文法：字符和字符串

- 字符串/符号串
  - 由字母表中的符号组成的任何有穷序列称为符号串
  - 通常使用小写字母表示符号串，如x=STR
  - 字=句子=串

例如:

(1) 001110是字母表Σ= {0,1}上的符号串；

(2) 字母表A= {a,b,c}上的一些符号串有：

a, b, c, ab, ba, aaca等。

文法：字符和字符串

- 符号串的递归定义
  - 字母表∑上的字符是∑上的符号串；
  - 若x是∑上的符号串，且$a∈∑$，则xa或ax是∑上的符号串；
  - y是∑上的符号串，当且仅当y可由(1)和(2)产生。

符号串长度

如果某符号串x中有m个符号，则称其长度为m，记为 | x | = m。

允许不包含任何符号的符号串，称其为空符号串或空串。空串用ε表示，其长度为0，记为 $|\varepsilon| = 0$。

文法：字符和字符串

- 例如

(1) 定义在字母表Σ={0,1}上的符号串

$$|001110| = 6$$

(2) 定义在字母表

Σ={ x,， y， z， =， 0， 1， +， ; }

上的符号串

$$|x{=}y{+}{+};z{=}0;| = 10$$

文法：字符和字符串

- 符号串的前缀
  - 设x是一个符号串，把从x的尾部删去0个或若干个符号之后剩余的部分称为x的前缀。
- 符号串的后缀
  - 从x首部删去0个或若干个符号之后剩余的部分称为x的后缀。
- 符号串的真前（后）缀
  - 若x前缀(后缀)不是x自身，则将其称为x的真前缀(真后缀)。
- 子符号串（子串）
  - 从一个符号串中删去它的一个前缀或(和)一个后缀之后剩余的部分称为该符号串的子符号串或子串。

文法：字符和字符串

- 例：设$x=abc$
  - ε, a, ab, abc都是x的**前缀**，且除abc外都为**真前缀**
  - ε, c, bc, abc都是x的**后缀**，且除abc外都为**真后缀**
  - abc是x的前缀或**后缀**，但既不是真前缀也不是真**后缀**。
- 例：设$x=abcd$
  - $\varepsilon, a, b, c, ab, bc, cd, abc, bcd$及$abcd$都是$x$的**子字符串**
  - Ac, ad, cb, bd, ba等都不是x的**子字符串**

文法：字符串运算

- 符号串的连接
  - 设*x*和*y*是两个符号串，如果将符号串*y*直接拼接在符号串*x*之后，则称此操作为符号串*x*和*y*的连接，记作*xy*。

设有字符串 j=abc，k=xyz

则jk=abcxyz，kj=xyzabc。

- 连接运算是有序的。一般来说xy≠yx，仅当x=y或其中之一为ε时,有xy=yx。

文法：字符串运算

- 符号串的方幂
  - 设x是某字母表上符号串，把x自身连接 n 次得到符号串z，即 $z = xx\ldots x$ （n个x），称z是符号串x的n次幂，记作$z=x^n$。

设x是符号串，则有定义

$$\boxed{x^0=\varepsilon}$$

注意

$$x^1=x$$

$$x^2=xx$$

$$x^3=x^2x=xx^2=xxx$$

……

$$x^n=x^{n-1}x=xx^{n-1}=\underbrace{xx\ldots x}_{n\text{个}}$$

文法：字符串运算

- 符号串集合的乘积

设A、B 是两个符号串集合，AB表示A与B的乘积，则有定义

$$AB=\{xy \mid (x\in A)\wedge(y\in B)\}$$

例如，设A={ab,c}, B={d,ef},

则 AB={abd, abef, cd, cef}

注意：

有 {ε}A= A{ε}=A，∅A= A∅ =∅ ，其中∅为空集。

$\varnothing\neq\{\varepsilon\}$

注意：

$$\varnothing=\{\ \}\neq\{\varepsilon\}$$

文法：字符串运算

- 符号串集合的方幂
  - 设A是符号串集合，A自身的乘积可以用方幂表示。则有定义

$A^0 = \{\varepsilon\}$

$A^1 = A$

$A^2 = AA$

$A^3 = A^2A = AAA$

……

$A^n = A^{n-1}A = AA\ldots A$ （n个A）

$A^{i+j} = A^i A^j$

$P = \{ab, x, aby\}$，

$P^2 = PP$

$= \{abab, abx, ababy, xab, xx, xaby, abyab, abyx, abyaby\}$

文法：字符串运算

- 符号串集合的并

设P、Q为字符串集，集合P ∪ Q为P和Q 的并，它的元素是P或Q中的元素。

例如，P={0, 1, 01}　　Q={0, 10, 11, 00}，

则 P ∪ Q = {0, 1, 01, 10, 11, 00}

文法：字符串运算

- 符号串集合的闭包

设A为符号串集，A的正闭包记作A^+，则有

$$A^+ = A^1 \cup A^2 \cup \ldots \cup A^n \cup \ldots$$

A的自反闭包记作A^*，则有

$$A^* = A^0 \cup A^+ = \{\varepsilon\} \cup A^+ = A^+ \cup \{\varepsilon\}$$

由定义知，

$$\begin{cases} A^+ = AA^* \\ A^* = A^0 \cup A^+ \end{cases}$$

设有 A={01,10}，则

$A^* = \{\varepsilon, 01, 10, 0101, 0110, 1001, 1010, 010101, 010110, \ldots\}$

$A^+ = \{01, 10, 0101, 0110, 1001, 1010, 010101, 010110, \ldots\}$

文法：字符串运算

- 串集合运算应用
  - 设有$L=\{A..Z,a..z\}$，$D=\{0,1,2,\ ...,9\}$
    - $L \cup D$ ={ 由字母和数字构成的集合 }
    - LD ={ 所有一个字母后跟随一个数字组成的字符串的集合 }
    - L^* ={ 由所有字母按任意顺序组成的字符串含（ε）所构成的集合 }
    - $L(L \cup D)^*$={ 由所有一个字母开头后跟随字母或数字组成的字符串或　的集合 }

文法：文法形式定义

一部文法G是一个四元组：$G=(V_N, V_T, S, P)$

- V_N：非空有限的非终结符号集(一般用大写字母表示)；
- V_T：非空有限的终结符号集（一般用小写字母表示）。
- S ：文法的开始符号或识别符号，亦称公理，$S \in V_N$。S代表语言最终要得到的语法范畴。
- P ：有限产生式集。

设V是文法G的符号集，则有$V=V_T \cup V_N$，$V_T \cap V_N=\varnothing$。

文法：文法形式定义

一部文法G是一个四元组：$G = (V_N, V_T, S, P)$

产生式就是按一定格式书写的定义语法范畴的文法规则，它是一部文法的实体。

产生式的形式:

$$\boxed{P \rightarrow \alpha \quad \text{或} \quad P::=\alpha}$$

其中：P称为产生式的左部，α为产生式的右部或称为P的候选式，有$P \in V^+$，$\alpha \in V^*$。

注意，公理S至少且必须在文法某个产生式的左部出现一次。

文法：文法形式定义

- 例

S

〈句子〉→〈主语〉〈谓语〉

〈主语〉→〈形容词〉〈名词〉

〈谓语〉→〈动词〉〈宾语〉

〈宾语〉→〈形容词〉〈名词〉

〈形容词〉→ 小 | 大

〈名词〉→ 八哥 | 花生

〈动词〉→ 吃

V_N　V_T

< NUMBER > → 0 | 1 | 2 | 3 | …| 9

其中:

V_N ={ NUMBER }

V_T ={0,1,2, …,9}

S= V_N

P为定义式本身。

文法：文法形式定义

- 例

简单的算术表达式文法G_1定义为

$$\underbrace{\{\ \{E\},\ \{i, +, *, (,)\},\ E,\ \{E \rightarrow i \mid i{+}i \mid i*i \mid (E)\}\ \}}_{\text{四元式形式}}$$

注意:

$$\left.\begin{array}{l} E \rightarrow i \\ E \rightarrow i{+}i \\ E \rightarrow i*i \\ E \rightarrow (E) \end{array}\right\} \xLeftrightarrow{\text{简写}} E \rightarrow i \mid i{+}i \mid i*i \mid (E)$$

文法：语言的形式化定义

给定一部文法*G*, 从*G*的开始符号*S*出发，反复使用产生式对非终结符进行替换，最后所得到的终结符号串的全体，即为文法*G*所描述的语言*L(G)*。

例： 设有文法G

$$S \rightarrow P \mid aPb$$
$$P \rightarrow ba \mid bQa$$
$$Q \rightarrow ab$$

则： L(G)={ba, abab, baba, ababab}

$S \Rightarrow P \Rightarrow ba$

$S \Rightarrow aPb \Rightarrow abab$

$S \Rightarrow P \Rightarrow bQa \Rightarrow baba$

$S \Rightarrow aPb \Rightarrow abQab \Rightarrow ababab$

文法：语言的形式化定义

- 直接推导“ => ”

有$V=\alpha A\beta=>\alpha\gamma\beta=W(\alpha,\beta,\gamma\in(V_N\cup V_T)^*)$,当且仅当P中存在一条规则$A\to\gamma$，称V直接推导出W（或W直接归约到V），记作：$V => W$。

- 直接推导序列

如果存在$V=\alpha_0=>\alpha_1,\alpha_1=>\alpha_2,\ldots,\alpha_{n-1}=>\alpha_n=W$

或$\alpha_1=>\alpha_2=>\alpha_3=>\ldots=>\alpha_{n-1}=>\alpha_n$，

则V经过n步(n>0)可以推导出W，记作: $V\overset{+}{=>}W$。当$V\overset{+}{=>}W$或$V=W$，记作：$V\overset{*}{=>}W$。

文法：语言的形式化定义

- 最左(右)推导

在推导过程中，总是对句型中的最左(右)边的非终结符进行替换，称为最左(右)推导。

- 句型

设有文法G[S]，若 $S \overset{*}{\Rightarrow} \alpha$（$\alpha \in (V_T \cup V_N)^*$），则称 α 为G[S]的句型 。

- 句子

设有文法G[S]，若 $S \overset{*}{\Rightarrow} \alpha$（$\alpha \in V_T^*$），则称 α 为G[S]的句子。

文法：语言的形式化定义

- 规范推导/规范句型/ 规范归约

最右推导也称为规范推导 。仅用规范推导得到的句型称为规范句型 。规范推导的逆序为规范归约。

文法：语言的形式化定义

例：设有文法G[E]:

$$E \rightarrow E * E \mid E + E \mid (E) \mid i$$

设有句子$1：i*i+i

$$E \underset{L}{\Rightarrow} E * E \underset{L}{\Rightarrow} i * E \underset{L}{\Rightarrow} i * E + E \underset{L}{\Rightarrow} i * i + E \underset{L}{\Rightarrow} i * i + i$$

最左推导

句子

$$E \underset{R}{\Rightarrow} E * E \underset{R}{\Rightarrow} E * E + E \underset{R}{\Rightarrow} E * E + i \underset{R}{\Rightarrow} E * i + i \underset{R}{\Rightarrow} i * i + i$$

句型

最右推导/规范推导

规范句型

文法：语言的形式化定义

- 文法的递归

设有文法G，$A \to \gamma$ 是G的产生式，若γ具有$\alpha A\beta$的形式，或$\gamma \overset{+}{\Rightarrow} \alpha A\beta$，则称G是递归文法。

若$\alpha = \varepsilon$，则G为左递归文法。若$\beta = \varepsilon$，则G为右递归文法。

- 递归文法
  - 直接递归
  - 间接递归

- 递归文法
  - 左递归
  - 右递归

文法：语言的形式化定义

例：　设有文法G_1：　$E \rightarrow E+E \mid E*E \mid (E) \mid i$

有E => E ...则文法G_1是直接递归文法。

例：　设有文法G_2：

$$T \rightarrow Qc \mid c$$

$$Q \rightarrow Rb \mid b$$

$$R \rightarrow Ta \mid a$$

有T=>Qc =>Rbc =>Tabc，即 $T \stackrel{+}{\Rightarrow} Tabc$，则文法$G_2$是间接递归文法。

文法：语言的形式化定义

- 语言

文法 G所产生的语言L(G)：

$L(G) = \{ \alpha \mid \alpha \in V_T^* \wedge S \overset{+}{\Rightarrow} \alpha$， S=开始符号 $\}$

例： 设有语言 $L(G1)= \{ ab^n a \mid n>=0 \}$, 求G1？

G1 ： $S \rightarrow aa \mid aRa$ $R \rightarrow b \mid Rb$

例： 设有文法 G：

$S \rightarrow 0S1 \mid 01$ 求L(G)？

$L(G) = \{ 0^n 1^n \mid n>=1 \}$

文法：语言的形式化定义

- 语言

文法 G所产生的语言L(G):

$$L(G) = \{ \alpha \mid \alpha \in V_T^* \wedge S \overset{+}{\Rightarrow} \alpha, \text{ S=开始符号} \}$$

例: 设有文法 G:

$S \rightarrow S0 \mid 0$ 求L (G) ?

$L(G) = \{ 0^n \mid n>=1 \}$

G’: $S \rightarrow 0S \mid 0$ $L(G') = L(G)$

文法：文法等价

若 $L(G_1)=L(G_2)$，则称文法G_1和G_2是等价的 。

例： 设有语言L(G)：

$$L(G)=\{\, a(b^n)\,a \mid n>=0 \,\}$$

$$=\{a()a,\ a(b)a,\ a(bb)a,\ \ldots\}$$

G: $S \rightarrow a(B)a$ $B \rightarrow Bb \mid b \mid \varepsilon$

G': $S \rightarrow a()a \mid a(B)a$ $B \rightarrow Bb \mid b$

文法：文法表示

- BNF表示法，元语言符号集： {→, <   >, | }
- 扩充BNF表示法 n(EBNF)
  - <FORTRAN标识符> → <字母>{<字母>|<数字>}$_0^7$
  - 引入圆括号：“(” “)”

    U → xa | xb | … | xz

    U → x（a | b | … | z）
  - 引入方括号：[t]

    <条件语句> → <如果子句>|<如果子句>else<语句>

    <如果子句> → IF<布尔表达式> THEN<语句>

    <条件语句> → <如果子句> [else<语句>]

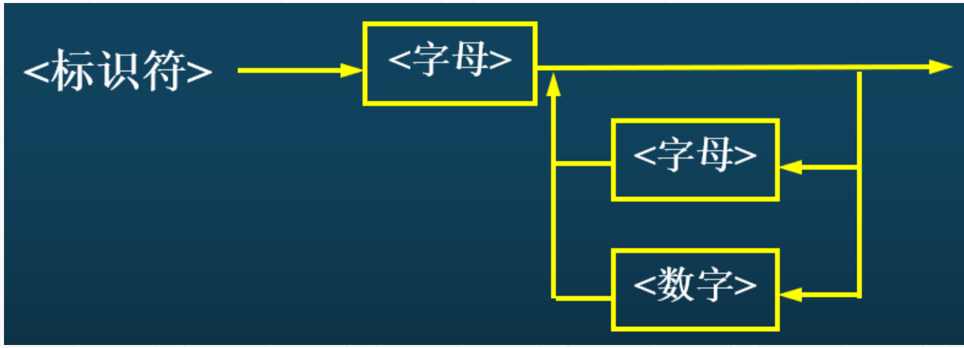
    <如果子句> → IF<布尔表达式> THEN<语句>

文法：文法表示

- 语法图

> 用椭圆和椭圆中的字符表示终结符；
>
> 用矩形内的字符表示非终结符；
>
> 第一个图的左部为文法的开始符号。

产生式P→ α　或　P::=α , α=bEFn。表示为:



文法：语法树与二义性

语法树是句子结构的图形表示，它代表了句子的推导结果，有利于理解句子语法结构的层次。

语法树的根结点	G的S
语法树的中间结点	G的V_N
语法树的叶结点	G的V_T
结点间关系	G的产生式规则

文法：语法树与二义性

例： 设有无符号整数的文法

<无符号整数> → <数字串>

<数字串> → <数字串><数字>|<数字>

<数字> → 0 | 1 | 2 | … | 9

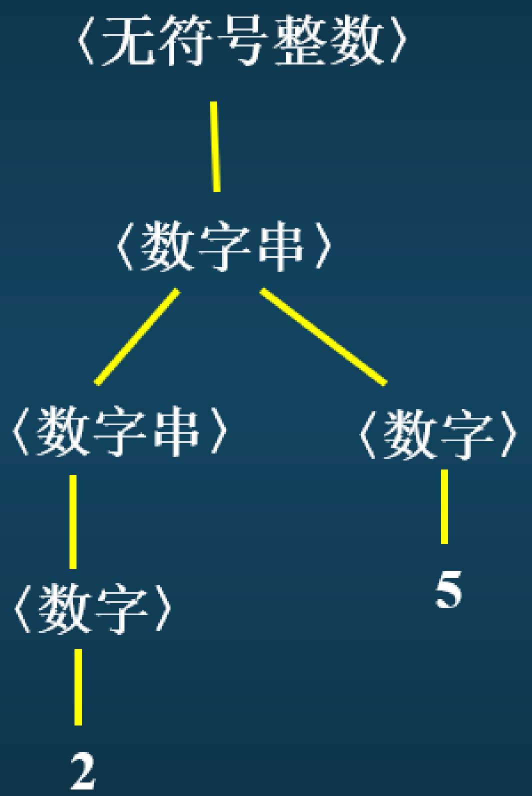
<u>对句子25的最左推导过程是：</u>

<无符号整数> $\underset{L}{\Rightarrow}$ <数字串> $\underset{L}{\Rightarrow}$ <数字串><数字> $\underset{L}{\Rightarrow}$ <数字><数字> $\underset{L}{\Rightarrow}$ 2<数字> $\underset{L}{\Rightarrow}$ 25

<u>对句子25的最右推导过程是：</u>

<无符号整数> $\underset{R}{\Rightarrow}$ <数字串> $\underset{R}{\Rightarrow}$ <数字串><数字> $\underset{R}{\Rightarrow}$ <数字串>5 $\underset{R}{\Rightarrow}$ <数字>5 $\underset{R}{\Rightarrow}$ 25

文法：语法树与二义性



注意：

一棵语法树包括了一个句型的所有可能的推导过程。

一个句型是否只对应一棵语法树？是否只有惟一的最左（右）推导？

文法二义性问题

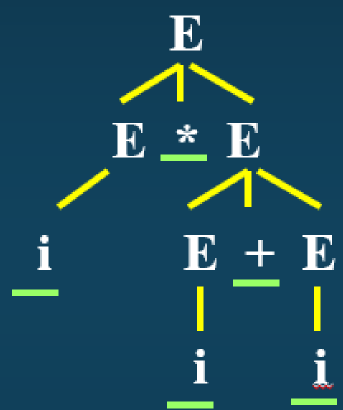
文法：语法树与二义性

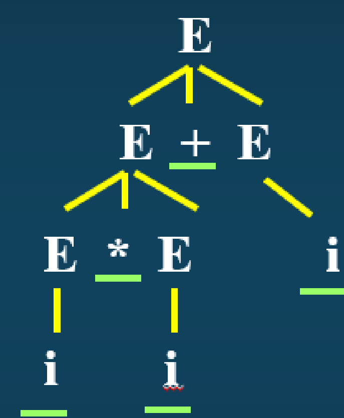
对一部文法G，如果至少存在一个句子,有两棵(或两棵以上)不同的语法树，则称该句子是二义性的。包含有二义性句子的文法称为二义文法。否则，该文法是无二义性的。

定义提供了对给定文法在某一范围内判定是否是二义性文法的充分条件。

文法：语法树与二义性

例： 设有文法G_1： $E \rightarrow E+E \mid E*E \mid (E) \mid i$





$$E \underset{L}{\Rightarrow} E*E \underset{L}{\Rightarrow} i*E \underset{L}{\Rightarrow} i*E+E \underset{L}{\Rightarrow} i*i+E \underset{L}{\Rightarrow} i*i+i$$

$$E \underset{L}{\Rightarrow} E+E \underset{L}{\Rightarrow} E*E+E \underset{L}{\Rightarrow} i*E+E \underset{L}{\Rightarrow} i*i+E \underset{L}{\Rightarrow} i*i+i$$

文法：语法树与二义性

- Time Flies
  - 既有语法又有语义的二义性
- 你真可以。
  - 有语义的二义性但无语法的二义性

注意：

文法的二义性与语义的二义性是完全不同的概念。并非文法是二义的，语言就二义。

文法：语法树与二义性

- 二义性消除

例如，对有文法G_1： $E \rightarrow E+E \mid E*E \mid (E) \mid i$

(1) 分析二义性原因

a) 运算符“+”和“*”未体现优先级；

b) “+”和“*”自身结合规则不明确；

(2) 构造G_1'，使$L(G_1)=L(G_1')$

$$
\begin{aligned}
E &\rightarrow T \mid E+T \\
T &\rightarrow F \mid T*F \\
F &\rightarrow (E) \mid i
\end{aligned}
$$

文法：分类

- Chomsky 分类
  - 0型文法（短语文法 ）

如果对文法G中的规则$\alpha \rightarrow \beta$不加任何限制，则称G为0型文法或短语文法。

其中，$\alpha, \beta \in (V_T \cup V_N)^*$ 且$\alpha \neq \varepsilon$。

0型文法相应的语言为0型语言L_0，0型语言可由图灵机（Turing）来识别。

$S \rightarrow aQb$

$aQb \rightarrow caRbc$

$aRb \rightarrow caba$

$L(G) = \{ccabac\}$

文法：分类

- Chomsky 分类
  - 1型文法（上下文有关文法 ）

设文法$G=(V_N, V_T, S, P)$，对P中的每个产生式($S\to \varepsilon$)限制为形如：

$$\alpha A\beta\to\alpha\gamma\beta$$

其中，$A\in V_N$，$\alpha, \beta\in(V_T\cup V_N)^*$，$\gamma\in(V_T\cup V_N)^+$，则称文法G为1型文法或上下文有关文法。

1型文法相应的语言为1型语言L_1，1型语言可由线性有界自动机来识别。

$S \to aSBC \mid abC$

$CB \to CD$

$CD \to BD$

$BD \to BC$

$bB \to bb$

$bC \to bc$

$cC \to cc$

$L(G_1) =\{a^nb^nc^n \mid n\geq1\}$

文法：分类

- 2型文法（上下文无关文法）

设文法G＝（V_N,V_T, S , P），对P中的每个产生式限制形如：

$A\rightarrow\alpha$ 　　其中，$A\in V_N$，$\alpha\in(V_T\cup V_N)^*$

则称文法G为2型文法。

2型文法也称为上下文无关文法。2型文法相应的语言为2型语言L_2，2型语言可由非确定的下推自动机来识别。

$S \rightarrow Ac \mid Sc$

$A \rightarrow ab \mid aAb$

$L(G_2) =\{a^n b^n c^m \mid n,m\geq 1\}$

文法：分类

- 3型文法（正则文法、线性文法）

设文法G=（V_N,V_T, S, P），对P中的每个产生式形如：

$A \to aB$　　或$A \to a$

或（$A \to Ba$　或 $A \to a$）

其中，A, B∈ V_N，a∈V_T，则称文法G为3型文法（正则文法或线性文法）。

3型文法相应的语言为3型语言L_3，3型语言可由确定的有限状态自动机来识别。

$S \to Bc \mid Sc$

$B \to Ab \mid Bb$

$A \to Aa \mid a$

$L(G_3) = \{a^n b^m c^k \mid n,m,k \geq 1\}$

文法：分类

0型文法 L_0	⟷	图灵机
1型文法 L_1	⟷	线性有界自动机
2型文法 L_2	⟷	非确定下推自动机
3型文法 L_3	⟷	确定有限自动机