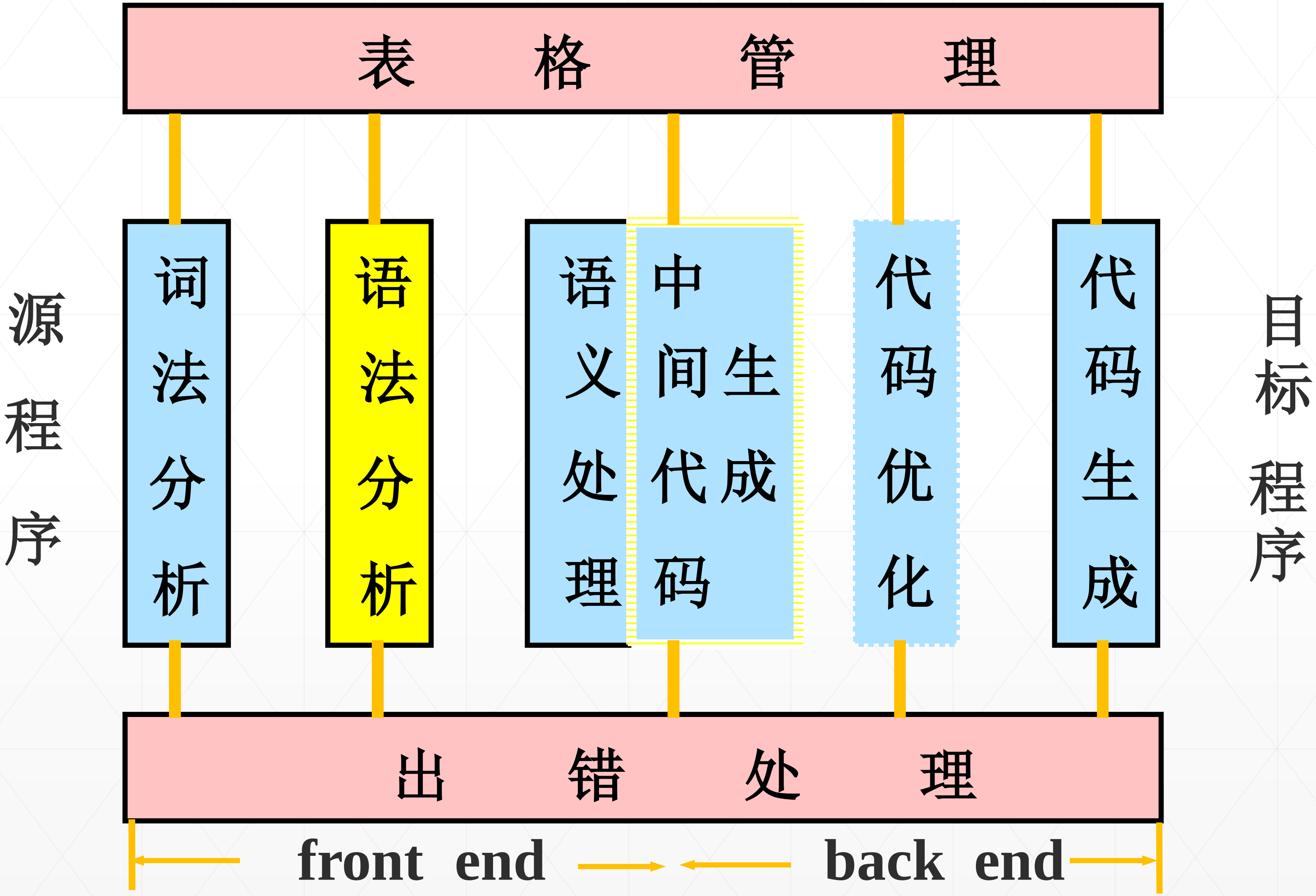
语法分析

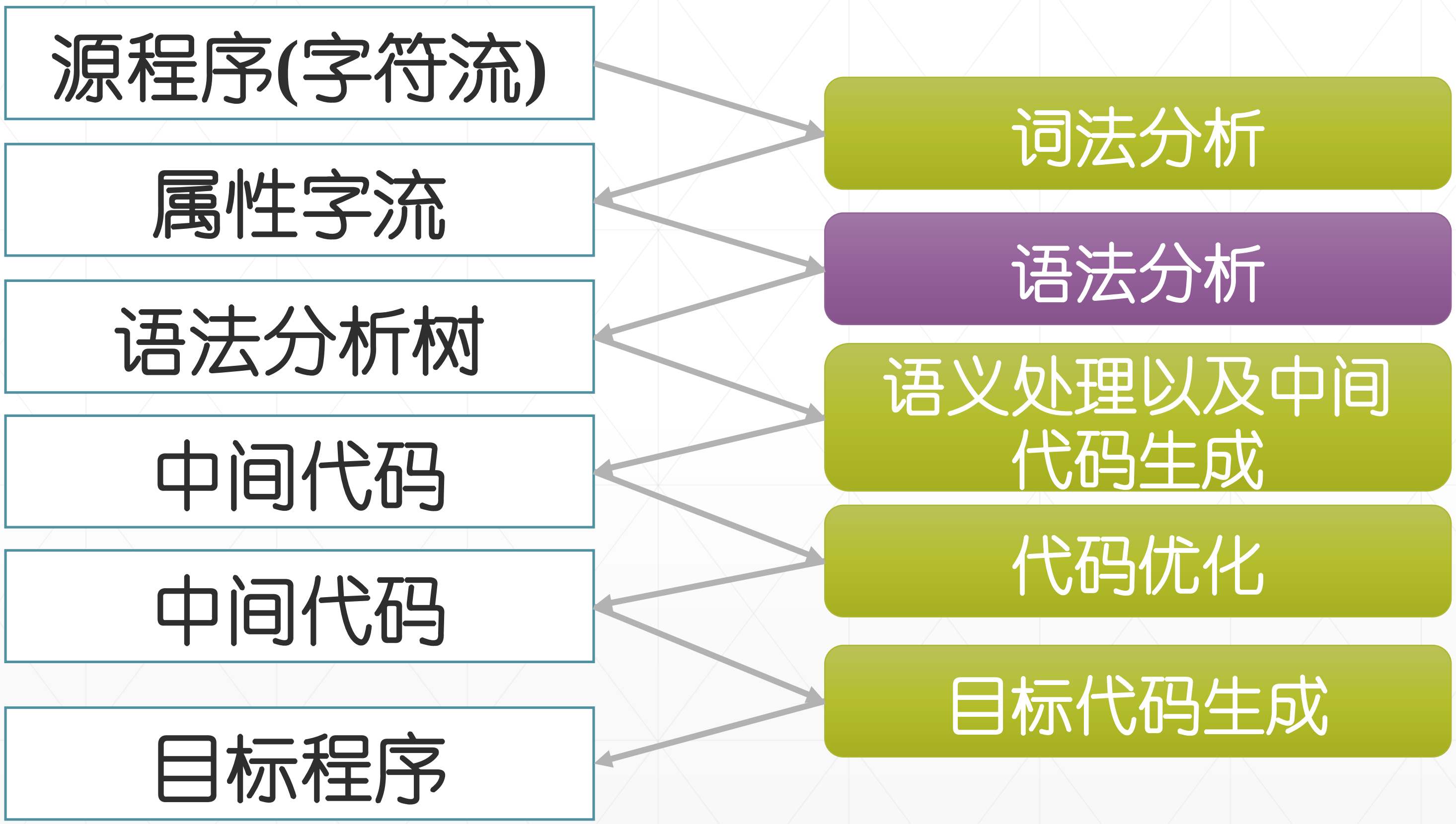
语法分析：概览



表格管理
源程序
词法分析
语法分析
语义处理
中间代码生成
代码优化
代码生成
目标程序
出错处理
front end
back end

语法分析：概览

- 基本功能



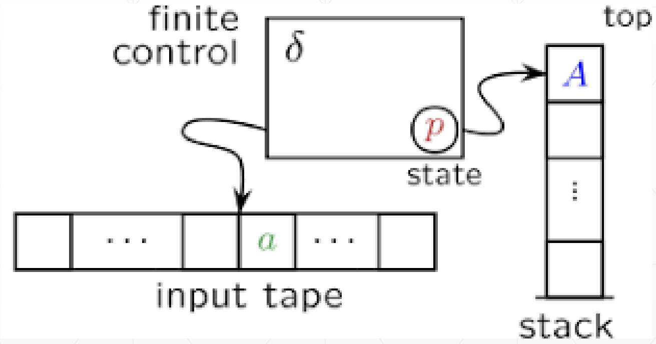
源程序(字符流)
属性字流
语法分析树
中间代码
中间代码
目标程序
词法分析
语法分析
语义处理以及中间代码生成
代码优化
目标代码生成

语法分析：概览

从左至右地扫描Token流，按照语言的语法规则识别语句结构，输出语法树或者语法错误信息。

语法分析：概览

- 自动生成工具
  - Bison/Yacc,CUP, ANTLR, SableCC, Beaver, JavaCC,...
- 内部对应一个确定性的下推自动机



语法分析：概览

语法说明

如何简洁地描述合法程序的结构

语法识别

编译器如何判断输入程序是否符合说明给出的结构

上下文无法文法

LL、LR 分析器

语法分析：概览

- 讲授内容
  - 文法介绍
  - 自顶向下的分析方法
  - 自底向上的分析方法
  - 二义文法分析与错误处理
  - 自动生成工具简介(自学)

- 关于语言：

  自然语言 —— 人与人交互的工具；

  程序设计语言—— 人机交互的工具。

本节展开思路

从文法和语言的直观概念

文法、语言的形式定义

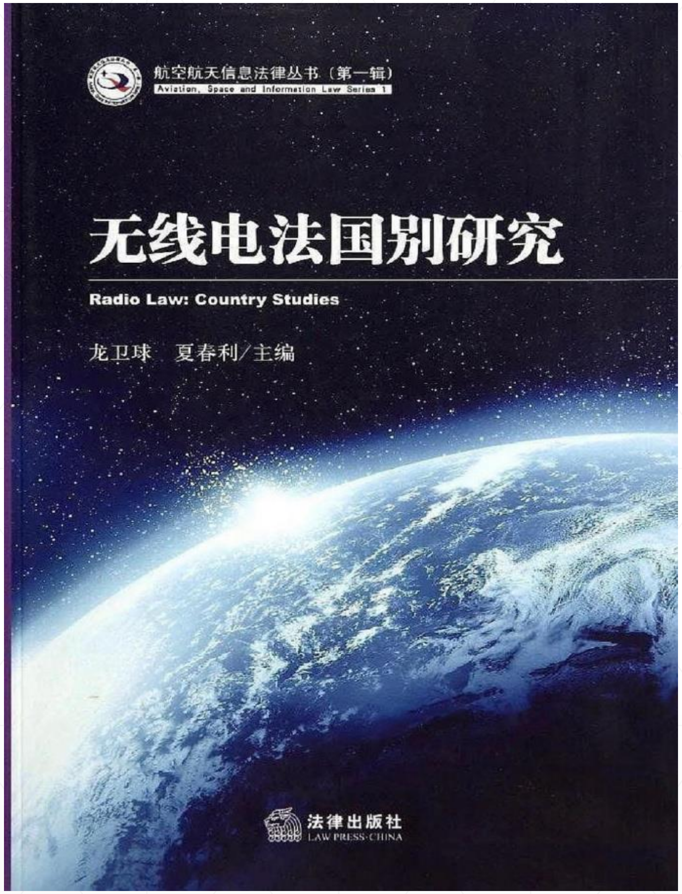
表示方法　类型　二义性问题

2.1 文法和语言

文法：自然语言的问题

2333 ………

文法：自然语言的问题



中文词性标注 无线电法国别研究

Jieba: 无线电/b 法国/ns 别/r 研究/vn
SnowNLP: 无线电/n 法国/ns 别/d 研究/v
PKUSeg: 无线电/n 法国/ns 别/d 研究/v
Thulac: 无线电/n 法国/ns 别/d 研究/v
HanLP: 无线电/n 法国/nsf 别/d 研究/vn
FoolNLTK: 无线/b 电法/n 国别/n 研究/n
LTP: 无线电/n 法国/ns 别/d 研

- 开会的时候，有人抽烟，老板咬牙道，“抽烟的都掐死。”
- 过几天天天天天气不好。
- 我背有点驼，麻麻说“你的背得背背背背佳”
- 六十老儿生一子人言非是我子也家产田园尽付与女婿外人不得争执
- 下雨天留客天留我不留

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.
- …

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

<u>**语法**</u>是一种媒介，<u>**语义**</u>以语法为媒介来表述。

<u>语言</u>是由单词按一定规则（文法）组成来表达特定意思的句子的集合。

对语言的分析集中于对句子的分析。

句子的分析依据：语言的文法规则。

- **例：设有语句“小八哥吃大花生”。**

汉语语法规则中的其中七条规则：

〈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

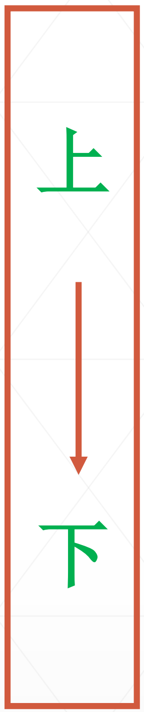
元语言:描述另一个语言的语言。

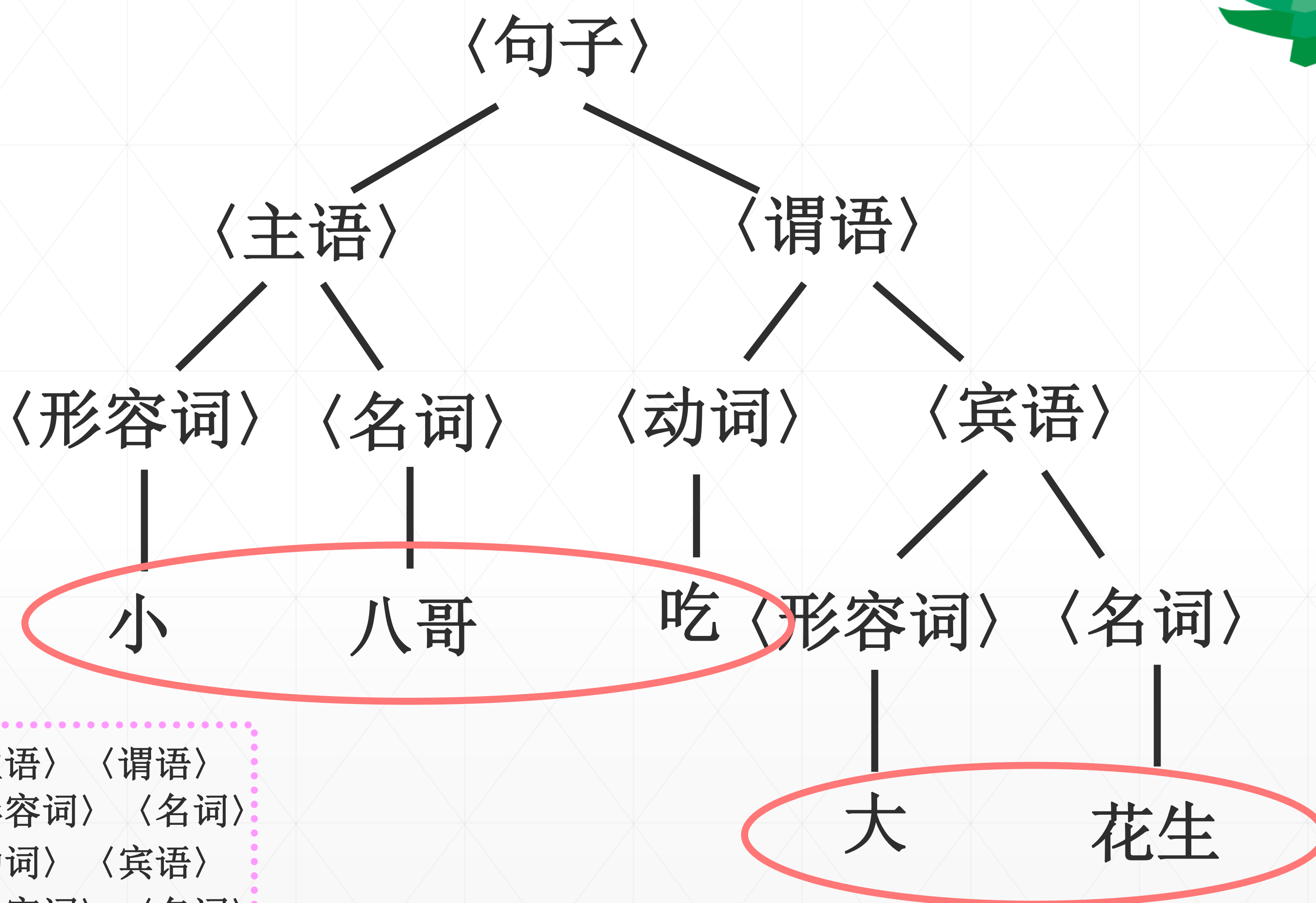
- **例：设有语句“小八哥吃大花生”。**

汉语语法规则中的其中七条规则：

〈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

语句“小八哥吃大花生”的语法分析树





〈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

- 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 <动词> <宾语>

⇒ 小八哥 吃<宾语>

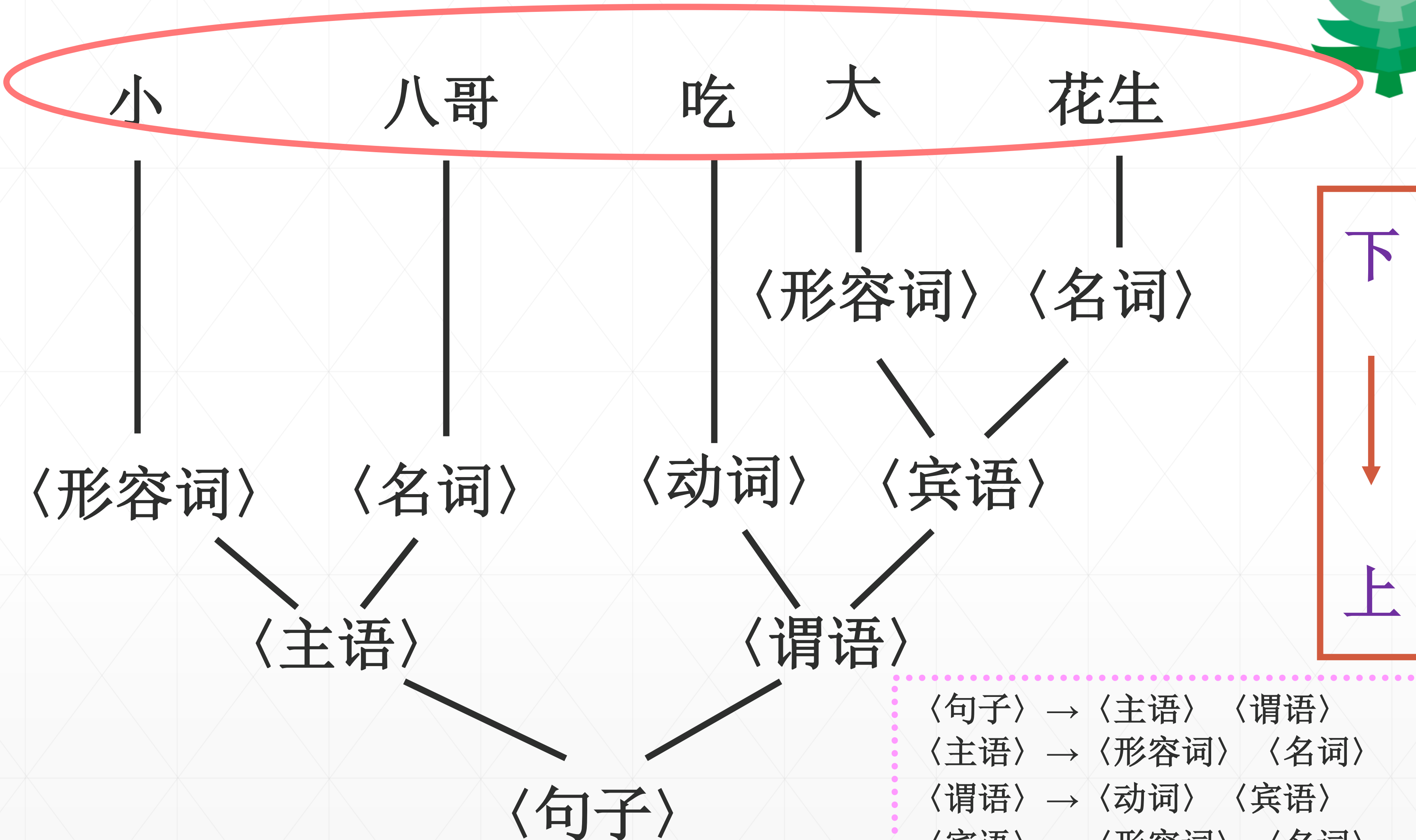
⇒ 小八哥 吃<形容词> <名词>

⇒ 小八哥吃大<名词>

⇒ 小八哥吃大花生

〈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

句子的归约



〈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

2.1　文法和语言

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
  - 用集合的列举法表示枚举出字母表中的符号
- 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∑={0,1}
  - **C语言**：字母表是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

已知∑是字母表，∑上的字符串的集合∑*可递归定义如下：

i) ε(ε是由∑中的0个字符组成的符号串,称为空串)∈∑*;

ii) 如果α∈∑*且a∈∑,那么αa∈∑*;

iii) α∈∑*当且仅当α由有限步i)和ii)产生。

∑*中的元素称为字符串。

📖通常使用小写希腊字母或小写字母表示符号串，

如α=*STR*表示 α是由符号*S*、*T*和*R*，并按此顺序组成的符号串。

- **符号串长度**

如果某符号串α中有m个**符号**，则称其长度为m，记为$|\alpha| = m$。

空串ε的长度定义为0，记为$|\varepsilon| = 0$。

例：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

$$|x{=}y{+}{+};\ z{=}0;| = 9$$

文法：字符和字符串

- 符号串的前缀、后缀和子符号串

设α是一个符号串，

从α的尾部删去0个或若干个符号之后剩余的部分称为α的前缀。

从α的首部删去0个或若干个符号之后剩余的部分称为α的后缀。

若α的前缀(后缀)不是α自身，则将其称为α的**真前缀(真后缀)**。

从一个符号串中删去它的一个前缀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α=abc
  - ε,a,ab,abc都是α的前缀，且除abc外都为真前缀
  - ε,c,bc,abc都是α的后缀，且除abc外都为真后缀
  - abc是α的前缀或后缀，但既不是真前缀也不是真后缀。
- 例：设α =abcd
  - ε,a,b,c,ab,bc,cd,abc,bcd及abcd都是α的子串
  - ac,ad,cb,bd,ba等都不是α的子串

符号串的连接

设α和β是两个符号串，如果将符号串β直接拼接在符号串α之后，则称此操作为符号串α和β的连接，记作$\alpha\beta$。

例如， 设有字符串 α=abc， β=xyz

则$\alpha\beta$=abcxyz， $\beta\alpha$=xyzabc。

连接运算是有序的。

一般$\alpha\beta \neq \beta\alpha$。

符号串的方幂

设α是某字母表上符号串，把α自身连接 **n** 次得到符号串β，即 $\beta = \alpha\,\alpha \ldots\ \alpha$ (n个α)，称β是符号串 α的n次幂，记作$\beta = \alpha^n$。

设α是符号串，则有

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

注意

$$\alpha^1 = \alpha$$

$$\alpha^2 = \alpha\,\alpha$$

$$\alpha^3 = \alpha^2\alpha = \alpha\,\alpha^2 = \alpha\,\alpha\,\alpha$$

$$\ldots\ldots$$

$$\alpha^n = \alpha^{n-1}\alpha = \alpha\,\alpha^{n-1} = \underbrace{\alpha\,\alpha \ldots \alpha}_{n个}$$

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，定义

$$AB=\{\alpha\beta \mid (\alpha \in A) \wedge (\beta \in B)\}$$

例如，设$A=\{ab,c\}$, $B=\{d,ef\}$,

则 $AB=\{abd, abef, cd, cef\}$

注意：

有$\{\varepsilon\}A=A\{\varepsilon\}=A$, $\varnothing A= A\varnothing =\varnothing$，其中$\varnothing$为空集。

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

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。

$\boxed{A^0=\{\varepsilon\}}$ 注意

$A^1=A$

$A^2=AA$

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

……

$A^n=A^{n-1}A=AA\ldots A$

显然有：

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

例：$A=\{ab, x, aby\}$，

则$A^2=AA$

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

符号串集合的并

设P、Q为字符串集，集合$P \cup Q$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 $P \cup Q = \{0, 1, 01, 10, 11, 00\}$

文法：符号串 集合运算

符号串集合的闭包

设A为符号串集，

A的正闭包记作A^+，定义为，

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

A的自反闭包记作A^*,定义为，

$$A^* = A^0 \cup A^1 \cup A^2 \cup \ldots \cup A^n \cup \ldots$$

由定义知，

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

例如， 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：字符和字符串

- 串集合运算应用
  - 设有$L=\{A..Z,a..z\}$， $D=\{0,1,2,\ldots,9\}$
    - $L \cup D$ ={ 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且包含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ε的集合 }

文法：文法形式定义

一部文法G是一个四元组$G=(V_N, V_T, S, P)$

其中：

V_N：非空有限的非终结符号集(一般用大写字母表示)。

其中的元素称为非终结符，或语法变量。

代表了一个语法范畴，表示某种语法结构。

V_T：非空有限的终结符号集（一般用小写字母表示）。

V_N、V_T合称为文法G的符号集V，$V= V_T \cup V_N$。$V_T \cap V_N =\varnothing$。

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。

按一定格式书写的定义语法范畴的文法规则，文法的实体。

产生式的形式(BNF):

$$\boxed{\alpha \rightarrow \beta \text{或} \ \alpha ::= \beta}$$

其中：α称为产生式的左部，

β称为产生式的右部或称为α的候选式。

$\alpha \in V^+$，且α中至少包含V_N中的一个元素，$\beta \in V^*$。

注意，开始符号S至少且必须在文法某个产生式的左部出现一次。

以S为开始符号的文法G可记为G (S)。

文法：文法形式定义

例：简单的算术表达式文法G_1定义为

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

文法的简化表示：

例：数字文法

$$< NUMBER > \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid \ldots \mid 9$$

其中： V_N ={ NUMBER }， V_T ={0,1,2, …,9}，

S= NUMBER， P为定义式本身。

文法：文法形式定义

汉语语法规则中的其中七条规则：

P

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_T

V_N

1. 语言的非形式化定义

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，最后所能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*

$$S \to P \mid aPb$$

$$P \to ba \mid bQa$$

$$Q \to ab$$

写出该文法所描述的语言。

文法： 语言的定义

▪直接推导 “ => ”

$\lambda=\alpha A\beta$， $\mu=\alpha\gamma\beta$ ， $(\alpha,\beta,\gamma\in V^*)$。

P中存在一条规则$A\rightarrow\gamma$,

称λ直接推导出μ (或μ直接归约到λ)

记作： $\lambda \Rightarrow \mu$ 。

▪直接推导序列

如果存在$\lambda = \alpha_0 \Rightarrow \alpha_1$, $\alpha_1 \Rightarrow \alpha_2$,…, $\alpha_{n-1} \Rightarrow \alpha_n = \mu$

或$\alpha_0 \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \alpha_3 \Rightarrow \ldots \Rightarrow \alpha_{n-1} \Rightarrow \alpha_n$,

则λ经过n步$(n>0)$可以推导出 μ， 记作: $\lambda \stackrel{+}{\Rightarrow} \mu$ 。 当 $\lambda \stackrel{+}{\Rightarrow} \mu$或$\lambda = \mu$， 记作： $\lambda \stackrel{*}{\Rightarrow} \mu$ 。

文法： 语言的定义

句型

对文法$G[S]$，若$S \stackrel{*}{\Rightarrow} \alpha (\alpha \in V^*)$，则称$\alpha$为$G[S]$的句型 。

句子

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

最左(右)推导

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。

规范推导/规范句型/ 规范归约

最右推导也称规范推导。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：语言的定义

例： 设有文法$G[E]$：$E \rightarrow E*E \mid E+E \mid (E) \mid i$

判断**$1**：$i*i+i$ 是该文法的句子

$$\underbrace{E \underset{L}{\Rightarrow} E*E \underset{L}{\Rightarrow} i*E \underset{L}{\Rightarrow} i*E+E \underset{L}{\Rightarrow} i*i+E \underset{L}{\Rightarrow} i*i+i}_{\text{最左推导序列}}$$

句子

$$\underbrace{E \underset{R}{\Rightarrow} E*E \underset{R}{\Rightarrow} E*E+E \underset{R}{\Rightarrow} E*E+i \underset{R}{\Rightarrow} E*i+i \underset{R}{\Rightarrow} i*i+i}_{\text{最右推导/规范推导序列}}$$

句型

规范句型

2. 语言的形式定义

语言

文法 G所产生(描述)的语言$L(G)$:

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

文法：　语言的定义

文法⇒语言

语言:句子的集合。

由给定文法构造句子的思想：

从文法的开始符号出发，

利用直接推导替换非终结符，

直至最终符号串全由终结符号组成。

例： 设有文法G

$$S \to P \mid aPb$$

$$P \to ba \mid bQa$$

$$Q \to ab$$

写出该文法所描述的语言。

文法：语言的定义

$$S \Rightarrow P \Rightarrow ba$$

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

$$S \Rightarrow aPb \Rightarrow abab$$

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

则： $L(G)=\{\underline{ba}, \underline{baba}, \underline{abab, ababab}\}$

文法G：

$S \rightarrow P \mid aPb$

$P \rightarrow ba \mid bQa$

$Q \rightarrow ab$

文法的递归

设有文法G，$A\rightarrow\gamma$ 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。

间接递归文法

直接递归文法

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 $\begin{cases}\text{直接递归}\\\text{间接递归}\\\text{左（右）递归}\end{cases}$

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

其中有$E \to E$ …这样的产生式，所以文法G_1是直接左递归文法。

例： 设有文法G_2：

$$T \to Qc \mid c$$
$$Q \to Rb \mid b$$
$$R \to Ta \mid a$$

其中有$T \to Qc$ $Qc \Rightarrow Rbc \Rightarrow Tabc$，

或 $T \overset{+}{\Rightarrow} Tabc$，则文法$G_2$是间接左递归文法。

文法和语言之间的相互转换举例

文法和语言之间的相互转换举例

例： 设有文法 G： $\boxed{\boxed{S \to S0} \mid 0}$ 求$L(G)$？

后面为0^n（$n \geq 1$)

$L(G) = \{ 00^n \mid n>=0 \}$

$= \{ 0^m \mid m>=1 \}$

G'： $S \to 0S \mid 0$

S替换为0

$L(G') = L(G)$

- 文法等价

若 $L(G_1)= L(G_2)$，则称文法G_1和G_2是等价的 。

文法和语言之间的相互转换举例

例： 设有文法 G：

$S \to 0S1 \mid \varepsilon$ 求$L(G)$？

S替换为空串

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

$= \{ 0^n 1^n \mid n>=0 \}$

前面为0^n,后面为$1^n(n\geq1)$

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

给出文法G1？

定义一个语法成分

$G1(S)$ ： $S \to aRa$ $R \to Rb \mid \varepsilon$

$G1(S)$ ： $S \to aT$ $T \to Ra$ $R \to Rb \mid \varepsilon$

$G1(S)$ ： $S \to aa \mid aRa$ $R \to b \mid Rb$

文法和语言之间的相互转换举例

例: 设有字母表{$a,b,(,)$}上的语言L :

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

写出描述语言L的文法。

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

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

文法和语言之间的相互转换举例

例：写出文法 G $\boxed{S \rightarrow Sb} \mid R$

$\boxed{R \rightarrow aRb \mid ab}$ 描述的语言 $L(G)$

后面为 b^m（$m \geq 1$)

产生串 $a^n b^n$（$n \geq 1$)

所以 $L(G) = \{a^n b^n b^m \mid n \geq 1, m \geq 0\}$

$= \{a^n b^{n+m} \mid n \geq 1, m \geq 0\}$

$= \{a^n b^j \mid j \geq n \geq 1\}$

文法和语言之间的相互转换举例

例： 设 定义在字母表{a,b,(,)}上的语言

$$L = \{ (a^n)(b^n) \mid n=1,2,3, \dots \}$$

写出该语言的文法。

一个语法成分 (非终结符）描述a、b个数相等的性质

G: $S \rightarrow (A)(B)$

$A \rightarrow aA \mid a$

$B \rightarrow bB \mid b$

×

a,b个数可以不相同

$B \rightarrow aBb \mid a)(b$

G: $S \rightarrow (\ B\ )$

$B \rightarrow a)(b \mid aBb$

文法和语言之间的相互转换举例

例： 设 定义在字母表$\{a,b\}$上的语言

$$L = \{ (ab)^n \mid n=1,2,3, \dots \}$$

写出该语言的文法。

$$G: \quad S \rightarrow abS \mid ab$$

定义在字母表$\{a,b\}$上的语言

$$L = \{ a^n b^n \mid n=1,2,3, \dots \}$$

$$G: \quad S \rightarrow aSb \mid ab$$

文法和语言之间的相互转换举例

例： 设 $L(G) = \{ 1^n 0^m 1^m 0^n \mid n,m \geq 0 \}$，

求$G(S)$?

一个语法成分 (非终结符）描述

$S \rightarrow 1S0 \mid B$

一个语法成分 (非终结符）描述

$B \rightarrow 0B1 \mid \varepsilon$

$G(S)$： $S \rightarrow 1S0 \mid B$

$B \rightarrow 0B1 \mid \varepsilon$

文法和语言之间的相互转换举例

例: 设语言 $S=\{a^ib^j \mid \boxed{0\le i\le j}\}$，满足 $L(G)=S$ 的文法 G 为【 $T\to AB \quad A\to aAb \mid \varepsilon \quad B\to Bb \mid \varepsilon$ 】。

$T\to Tb \mid A \qquad A\to aAb \mid \varepsilon$

$T\to aTb \mid B \qquad B\to Bb \mid \varepsilon$

$T\to aTb \mid Tb \mid \varepsilon$

$j=i+m \qquad m\ge 0$

$a^ib^j=a^ib^{i+m}=a^ib^i\,b^m=a^ib^mb^i$

文法和语言之间的相互转换举例

例: 设语言 $S=\{a^i b^j \mid \boxed{0 \le i \le j \le 2i}\}$，满足 $L(G)=S$ 的文法 G 为【 $T \to aTbb \mid A \quad A \to aAb \mid \varepsilon$ 】。

$T \to aTb \mid aTbb \mid \varepsilon$

$j=i+m \qquad 0 \le m \le i$

$i=m+n \qquad n \ge 0$

$j=2m+n \quad n \ge 0 \quad m \ge 0$

$a^i b^j = a^{m+n} b^{2m+n} = a^m a^n b^n b^{2m}$

令：$j=i+m+n$

$0 \le m \le i \quad 0 \le n \le m$

设语言 $S=\{a^i b^j \mid 0 \le i \le j \le 3i\}$，满足 $L(G)=S$ 的文法 G 为【____________________】。

文法和语言之间的相互转换举例

例:写一个文法，使其语言是奇整数的集合，每个奇整数不以0为前导。

解： 语言集合{1,-1,3,-3,5,-5,7,-7,9,-9,11,-11,…}

V_T={0,1,2,3,4,5,6,7,8,9,-}

V_N={S, E, D, C, B, A}

开始符号：S

$S \to E|-E$

$E \to A|BA|BDA$

$D \to CD|C$

$C \to 0|B$

$B \to 2|4|6|8|A$

$A \to 1|3|5|7|9$

文法和语言之间的相互转换举例

例: 设集合A为字母表Σ={0,1}上的有相同个数的0和1组成的符号串集合，给出正确描述集合A的文法。

解: 语言集合{ε,01,10,0011,0101,0110,1001,1010,1100…}

$V_T=\{0,1\}$

开始符号：S

$V_N=\{S, P, Q\}$

P：1的个数=0的个数+1，

Q：0的个数=1的个数+1，

$S \to \varepsilon \mid 0P \mid 1Q$

$P \to 0PP \mid 1S$

$Q \to 0S \mid 1QQ$

2.1　文法和语言

文法的表示方法

1. **BNF表示法**

   元语言符号集：{ →(::=), <　>, | }

2. **扩充BNF表示法（EBNF）**

   元语言符号集：{ → (::=), <　>, |,**{ },( ),[ ]** }

3. **语法图(上下文无关文法）**

文法的表示方法

2. 扩充BNF表示法 （EBNF）

1) 花括号 —$\{t\}_m^n$

串t重复的最大次数

串t重复的最小次数

省略m,n：t可重复0到任意多次。

2)圆括号 — “（”　“）”

提取产生式中的公共因子，简化产生式的表示。

3) 方括号——[t]

t 可有可无。

文法的表示方法

例：FORTRAN语言中标识符的定义：

长度≤8的字母开头后跟字母或数字的字符串

<FORTRAN标识符> → <字母>{<字母>|<数字>}$_0^7$

例：有文法规则

$$U \to xa \mid xb \mid \ \ldots \mid xz$$

等价于 $U \to x\,(a \mid b \mid \ldots \mid z)$

注意：

与终结符区分开。

例：设有条件语句的文法

<条件语句> → <如果子句>|<如果子句>else<语句>

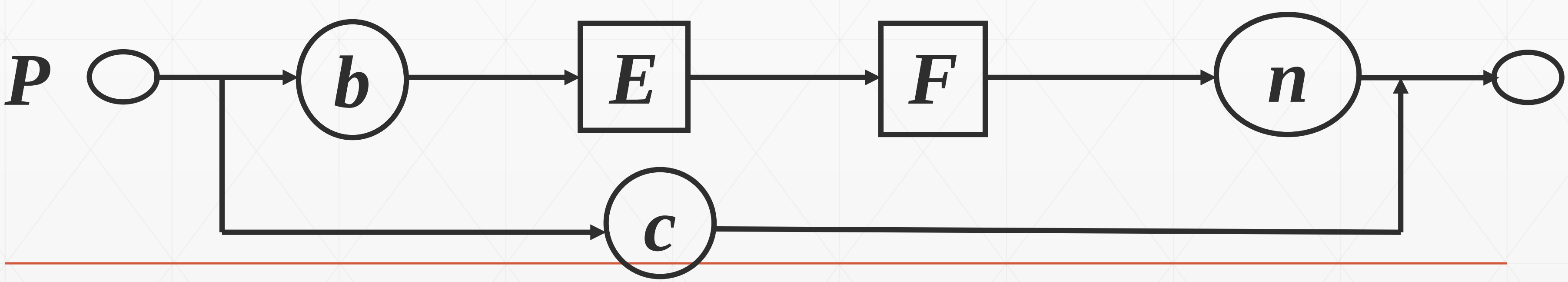
<如果子句> → if<布尔表达式> then<语句>

<条件语句> → <如果子句> [else<语句>]

3. 语法图(上下文无关文法）

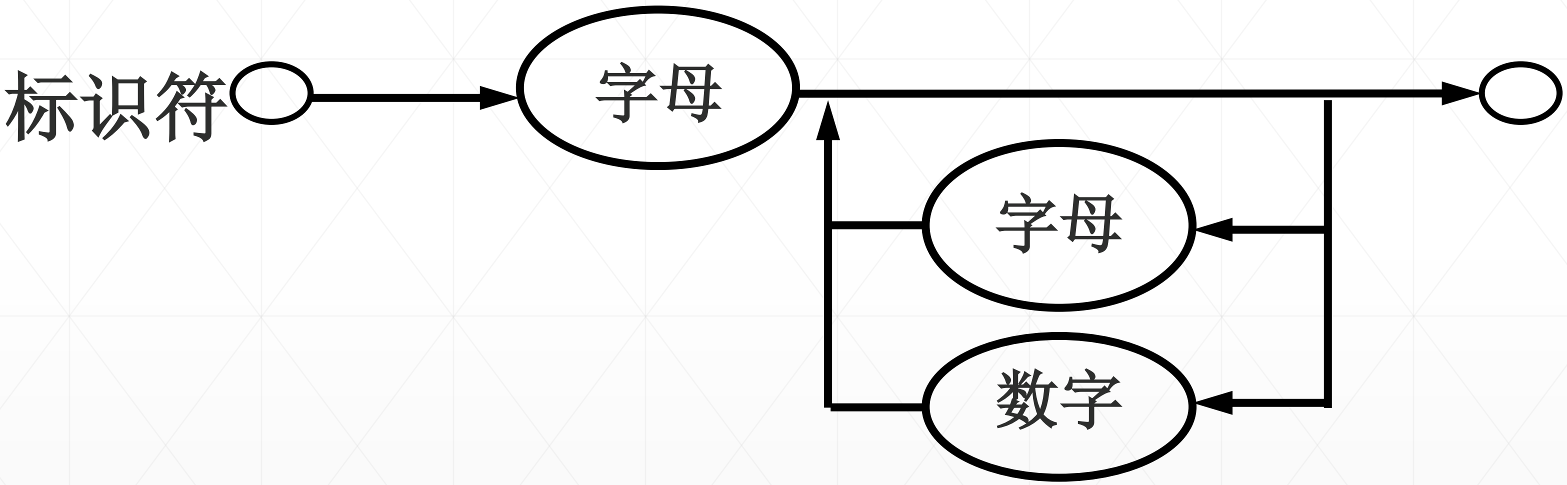
- 用图描述产生式规则
- 每个图都有一个起始结点和一个终止结点，其他的结点标记为文法符号。
- **终结符**结点用圆形表示。
- **非终结符**结点用方形表示。
- 从起始结点到终止结点的所有路径(标记为结点序列)定义**候选式**。

例： 产生式$P \rightarrow bEFn \mid c$ 。表示为



文法的表示方法

例： PASCAL语言中标识符的定义如下图。



2.1 文法和语言

语法分析树与二义性

语法分析树是句子结构的图形表示，展示出句子的推导过程，也利于理解句子语法结构的层次。

- 表示：

分析树的根结点	⟷	*G*的*S*
分析树的中间结点	⟷	*G*的产生式左部
父子结点间关系	⟷	*G*的产生式规则
叶结点从左到右连接的符号串	⟷	句型

语法分析树与二义性

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**

句子25的最左推导：

<无符号整数> => <数字串>

=> <数字串><数字>

=> <数字><数字> => 2<数字> => **25**

句子25的最右推导：

<无符号整数> => <数字串>

=> <数字串><数字>

=> <数字串>5 => <数字>5 => **25**

句子25的语法分析树

语法分析树与二义性

任何一个句型的一棵分析树包括了这个句型的所有可能的推导过程?

一个句型是否只对应一棵分析树?

或者是否只有惟一的最左(右)推导?

文法二义性问题

语法分析树与二义性

- 二义文法

一部文法*G*，如果至少存在一个句子（或句型），

有两棵(或两棵以上)不同的分析树（或最左推导或最右推导），

称该句子（或句型）是二义性的。

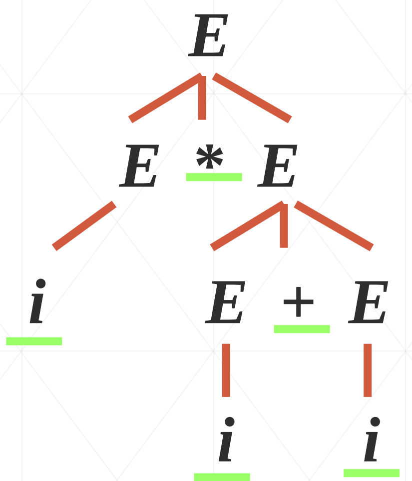
包含有二义性句子（或句型）的文法称为二义文法。

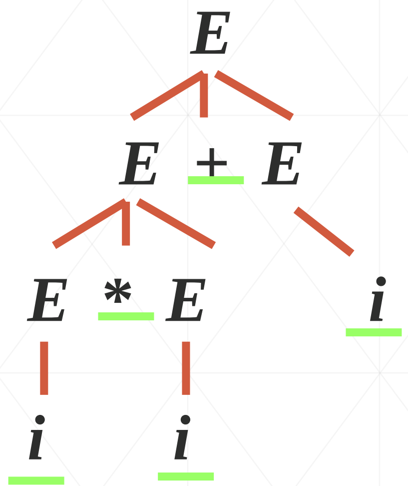
否则，该文法是无二义性的。

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。

语法分析树与二义性

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





文法G_1中的句子$i*i+i$存在两个不同的最左推导，所以文法G_1为二义文法。

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

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

文法和语言之间的相互转换举例

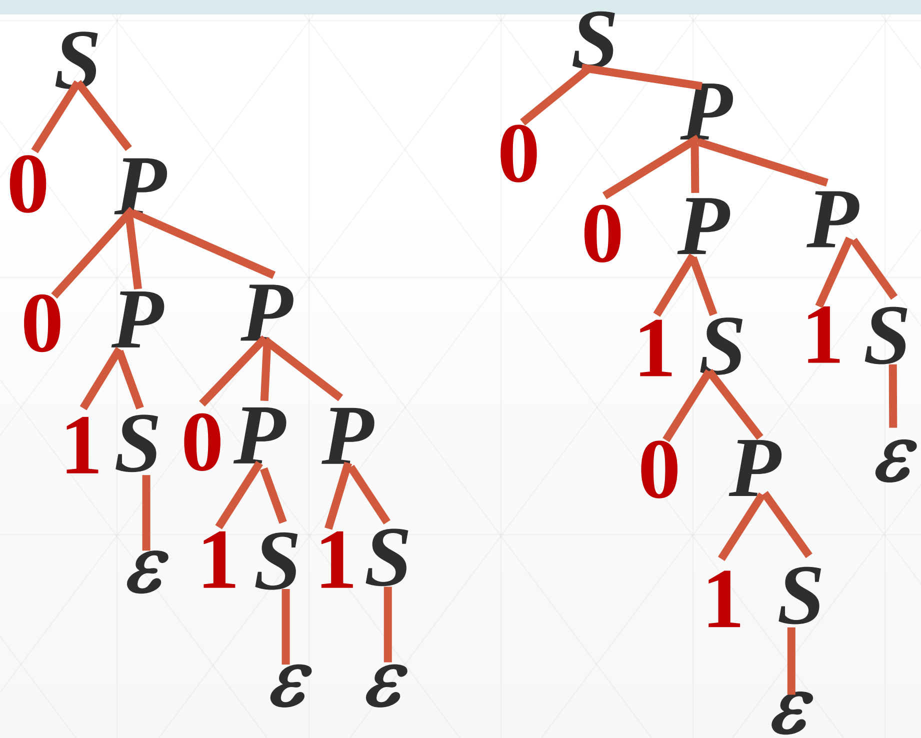
例: 设集合A为字母表$\Sigma=\{0,1\}$上的有相同个数的0和1组成的符号串集合，给出正确描述集合A的文法。

$S \to \varepsilon \mid 0P \mid 1Q$

$P \to 0PP \mid 1S$

$Q \to 0S \mid 1QQ$

句子**001011**存在两棵不同的分析树，所以文法为二义文法。



语法分析树与二义性

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，语义就二义。

Time Flies.	<u>既有语法又有语义的二义性</u>
i+i+i	<u>有语法的二义性但无语义的二义性</u>
你真可以。	<u>有语义的二义性但无语法的二义性</u>

语法分析树与二义性

文法二义性的消除

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$L(G_1)=L(G_1')$

$$
\begin{aligned}
G_1': \quad & E \rightarrow T \mid E+T \\
& T \rightarrow F \mid T*F \\
& F \rightarrow (E) \mid i
\end{aligned}
$$

2.1　文法和语言

文法和语言的类型

- **N.Chomsky 分类：**

四类文法:

0型文法（ 短语文法 ）

1型文法（上下文有关文法）

2型文法（上下文无关文法）

3型文法（线性文法、正则文法）

文法和语言的类型

1. 0型文法（短语文法）

文法G中的规则$\alpha \to \beta$不加任何限制。

可用0型文法描述的语言为0型语言L_0，

0型语言可由图灵机（Turing）来识别。

例：设有文法G

$S \to 0|AC0B$

$\boxed{C0 \to 00C}$

$CB \to DB|E$

$0D \to D0$

$AD \to AC$

$0E \to E0$

$AE \to \varepsilon$

G是0型文法

$L(G) = \{0^n \mid n$为2的非负整数次幂$\}$

2．1型文法（上下文有关文法）

- 每个产生式限制为：

$$\alpha A\beta \rightarrow \alpha\gamma\beta$$

其中，$A \in V_N$，α，β，$\gamma \in (V_T \cup V_N)^*$

可用1型文法描述的语言为1型语言L_1，

1型语言可由线性有界自动机来识别。

文法和语言的类型

例：设有文法G_1

$S \to aSBC \mid abC$

$CB \to CD$

$CD \to BD$

$BD \to BC$

$\Rightarrow CB \Rightarrow BC$

$bB \to bb$

$bC \to bc$

$cC \to cc$

$S \overset{*}{\Rightarrow} a^{n+1}bC(BC)^n$

$= a^{n+1}b(CB)^nC$

$\overset{*}{\Rightarrow} a^{n+1}b(BC)^nC$

$= a^{n+1}bB(CB)^{n-1}C^2$

$\Rightarrow a^{n+1}bb(CB)^{n-1}C^2$

$\overset{*}{\Rightarrow} a^{n+1}bb(BC)^{n-1}C^2$

$\overset{*}{\Rightarrow} a^{n+1}b^{n+1}C^{n+1}$

$\overset{*}{\Rightarrow} a^{n+1}b^{n+1}c^{n+1}$

G_1是1型文法，G_1产生的语言：

$L(G_1) = \{a^nb^nc^n \mid n \geq 1\}$

3．2型文法（上下文无关文法）

每个产生式限制形如：

$$A \rightarrow \alpha$$

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

能用2型文法描述的语言为2型语言L_2，2型语言可由非确定的下推自动机来识别。

例： 设有文法G_2

$$S \rightarrow Ac \mid Sc$$

$$A \rightarrow ab \mid aAb$$

G_2是2型文法

G_2产生的语言：

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

4．3型文法（正则文法、线性文法）

右线性文法：每个产生式形如

$$A \to \alpha B \qquad 或 A \to \alpha$$

左线性文法：每个产生式形如：

$$A \to B\alpha \qquad 或\ A \to \alpha$$

其中，A，$B \in V_N$，$\alpha \in V_T^*$

右线性文法和左线性文法统称为3型文法。

也叫正则文法或线性文法。

能用3型文法描述的语言为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。

文法和语言的类型

例：设有文法G_3

$$S \to Bc \mid Sc$$

$$B \to Ab \mid Bb$$

$$A \to Aa \mid a$$

G_3是左线性文法

G_3是3型文法

G_3产生的语言：

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

文法和语言的类型

例：设有文法G_3

$$S \rightarrow aB \mid c$$

$$B \rightarrow Sb \mid b$$

G_3不是右线性文法，也不是左线性文法

G_3不是3型文法

G_3是2型文法

G_3产生的语言：

$$L(G) = \{a^n c b^n \mid n \geq 0\} \cup \{a^n b^n \mid n \geq 1\}$$

5. 四类文法和语言小结及关系

描述功能越来越强 ↑ 产生式限制越来越严 ↓（弱 → 强）

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

$$\alpha \rightarrow \beta$$

0 → 2　　1 → 3

0型语言 ⊃ 1型语言 ⊃ 2型语言 ⊃ 3型语言

（$L_0 \supset L_1 \supset L_2 \supset L_3$）