语法分析

语法分析：概览

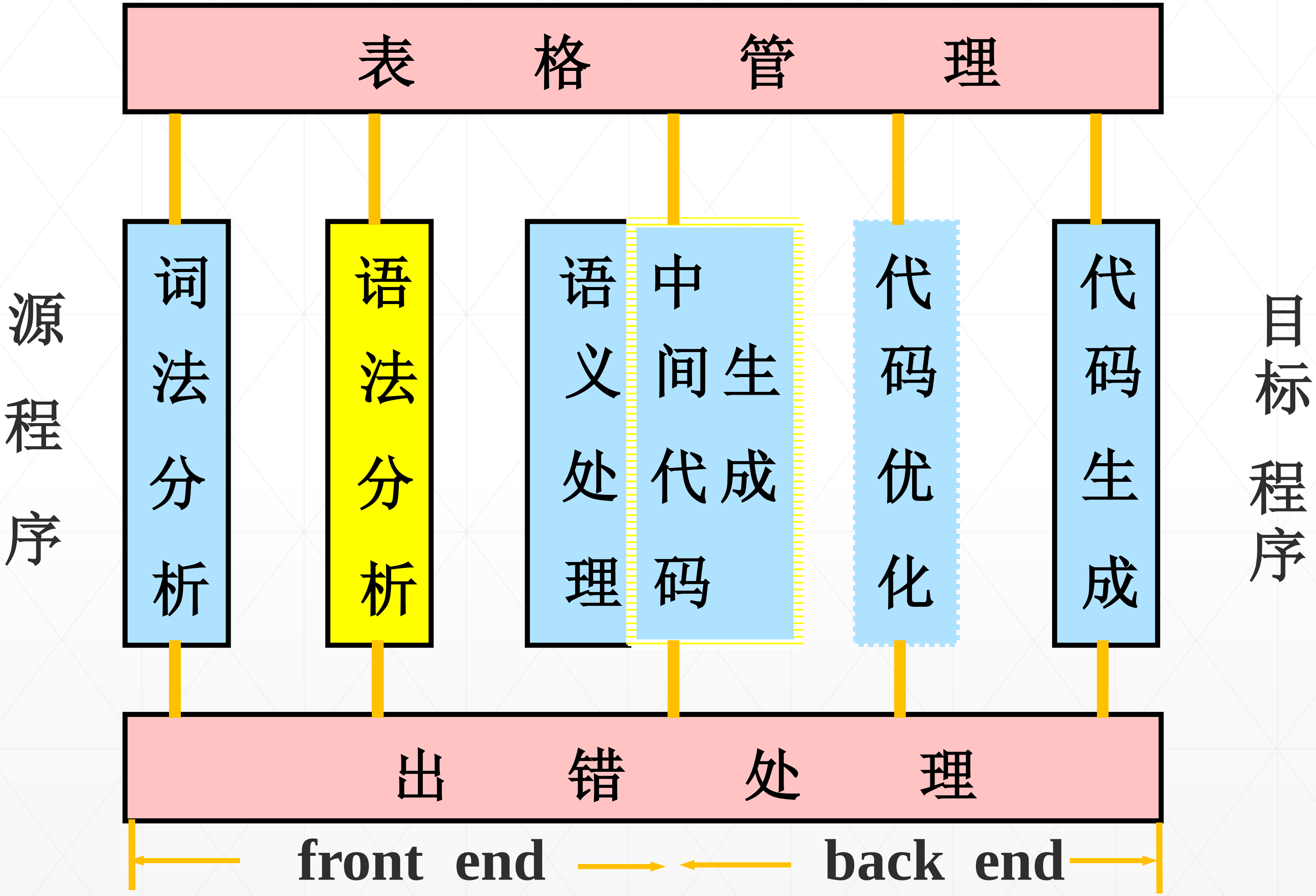
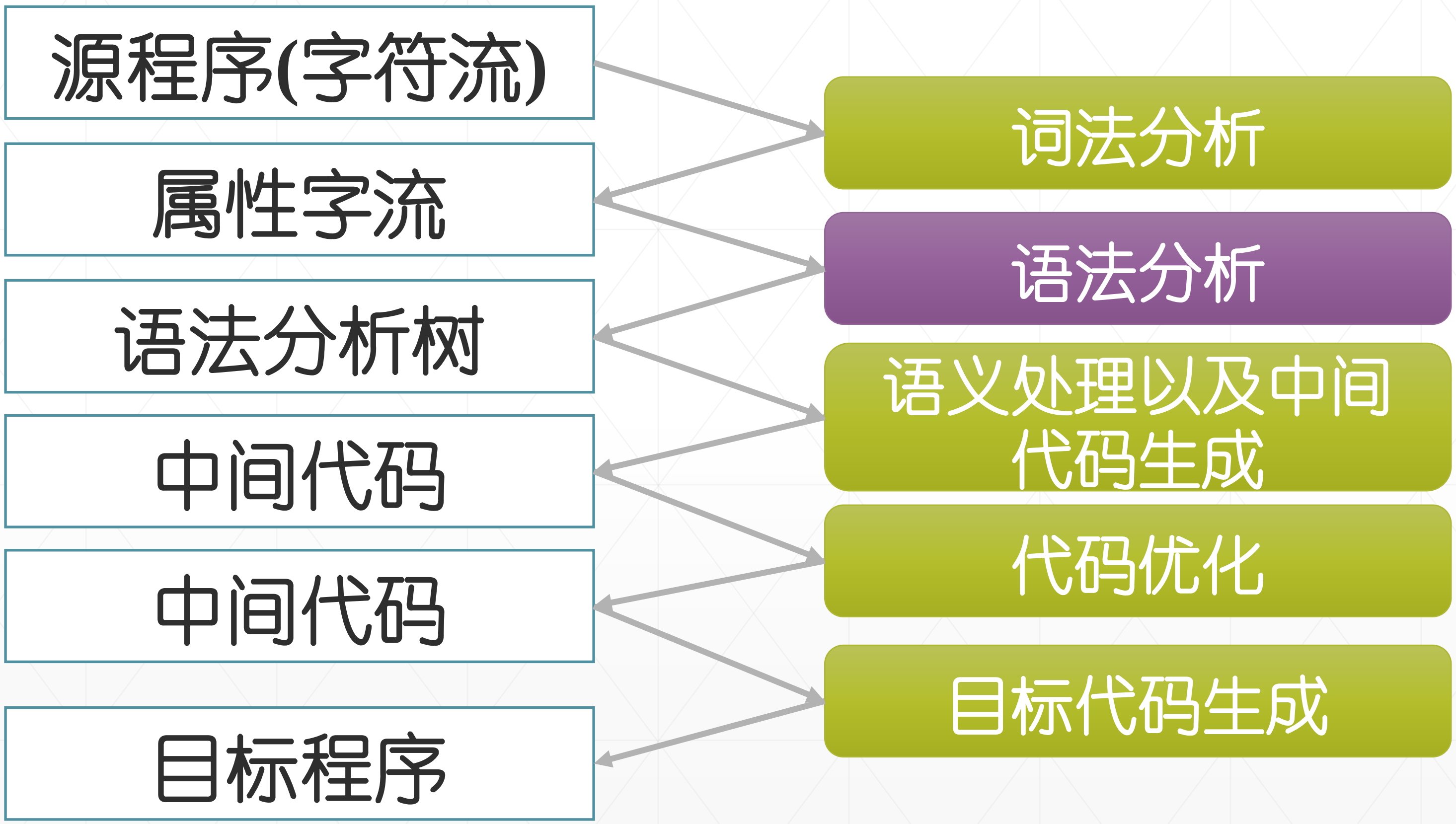


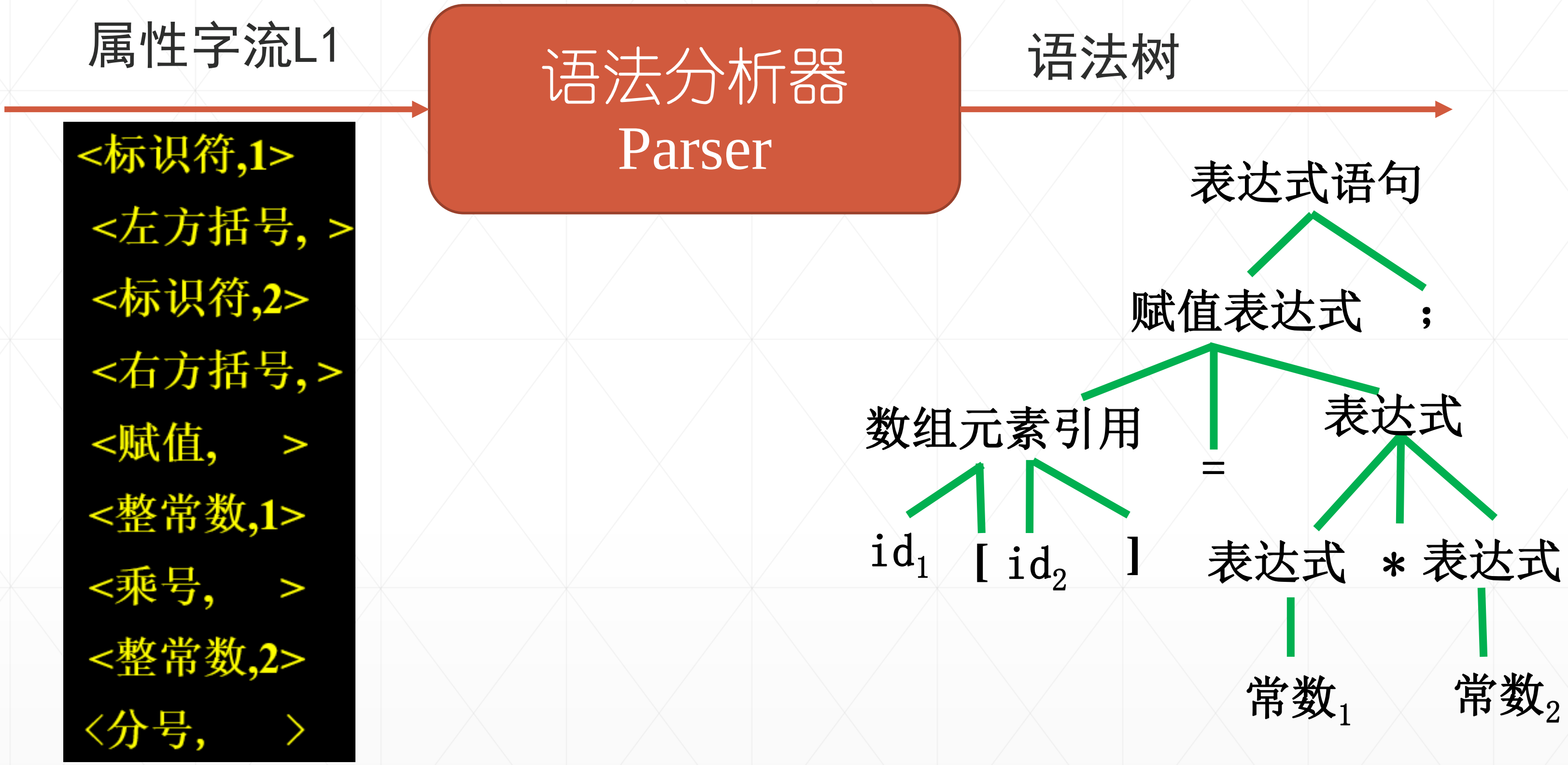
表 格 管 理
源程序
词法分析
语法分析
语义处理
中间代码生成
代码优化
代码生成
目标程序
出 错 处 理
front end
back end

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。

词法分析：概览

关键问题

- 如何定义语言的语法规则
- 如何识别属性字流的语法结构

语法结构的识别

语法结构的定义

上下文无关文法

分析器构造

递归下降分析器

LL分析器

确定的下推自动机

语言设计和定义

语言分析

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)

2.1 文法和语言

- 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

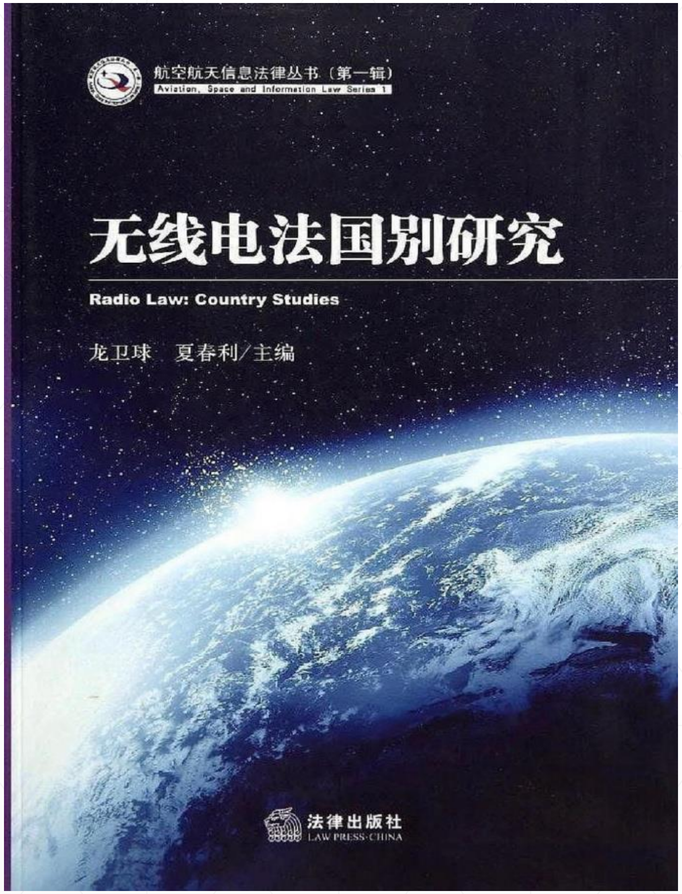
2.1 文法和语言

- 2.1.1 语言的语法和语义
- 2.1.2 文法和语言的定义
- 2.1.3 文法的表示方法
- 2.1.4 语法树与二义性
- 2.1.5 文法和语言的类型

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

语言的语法和语义

- 语言要素
  - 语法：语言的描述规则
  - 语义：语言传递的信息

语法是一种媒介，**语义**以语法为媒介来表述。

语言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〉

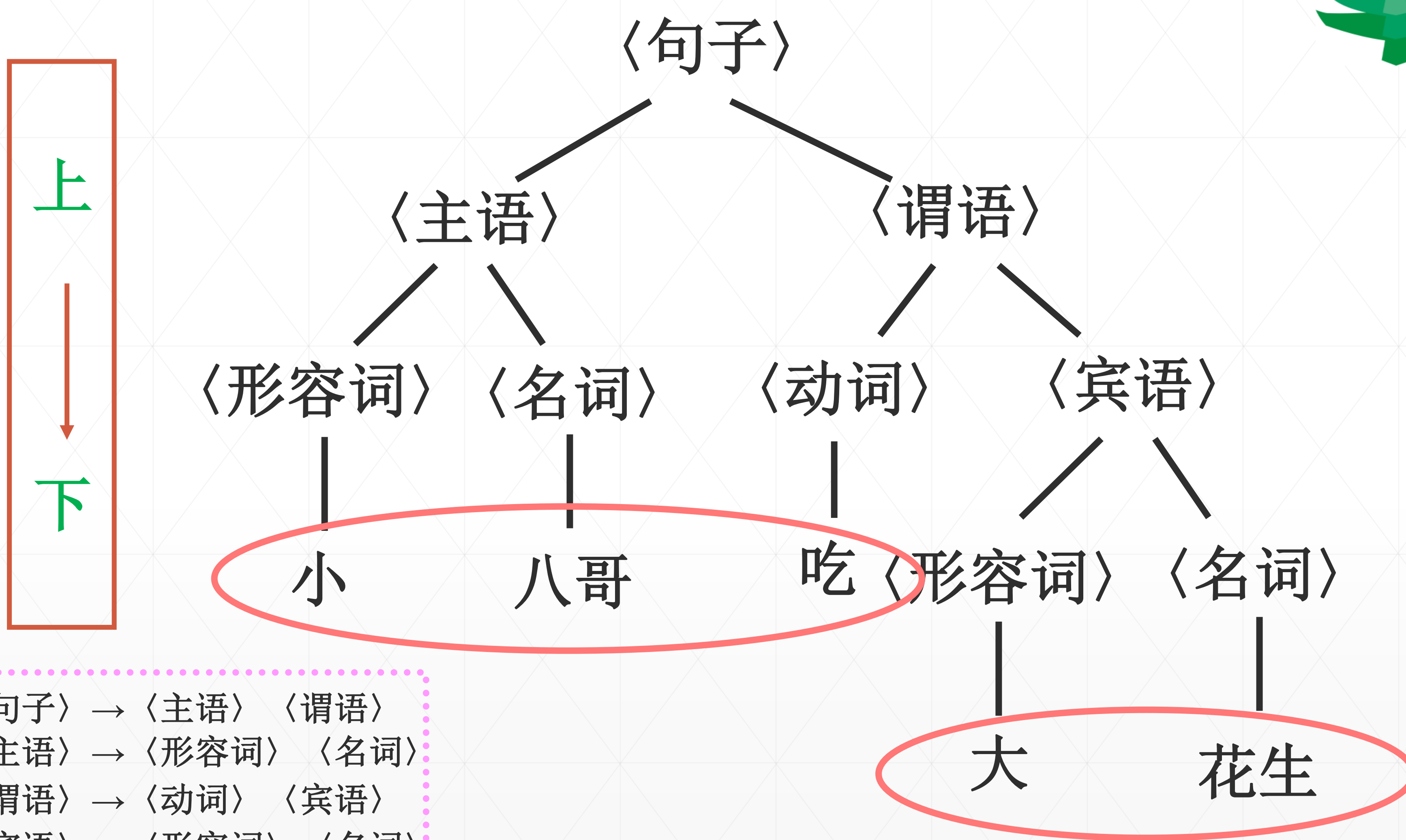
〈宾语〉→〈形容词〉〈名词〉

〈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>

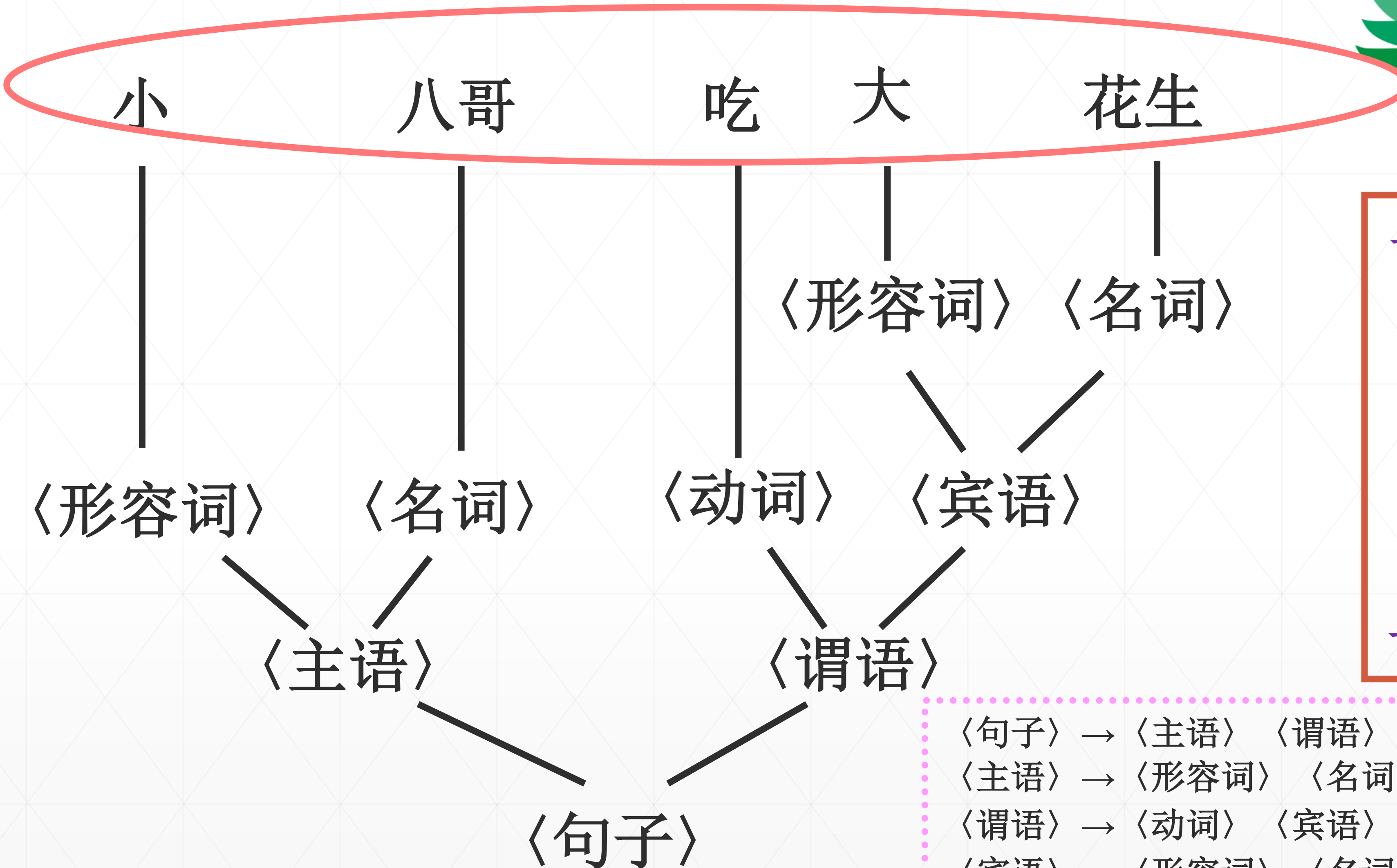
⇒ 小八哥 吃<形容词> <名词>

⇒ 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

2.1.1　语言的语法和语义

2.1.2　文法和语言的定义

2.1.3　文法的表示方法

2.1.4　语法树与二义性

2.1.5　文法和语言的类型

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：非终结符号集(非空有限的)。

元素称为非终结符号，或语法变量。

代表了一个语法范畴，表示某种语法结构。

V_T：终结符号集(非空有限的)。

元素称为终结符号。

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规则，文法的实体。

文法：文法形式定义

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，$\beta \in V^*$。

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

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

文法的简化表示：

例：数字文法G_2

$$<\text{NUMBER}>\ \rightarrow 0\ |\ 1\ |\ 2\ |\ 3\ |\ \ldots|\ 9$$

其中：$V_N=\{\text{NUMBER}\}$，$V_T=\{0,1,2,\ldots,9\}$，

$S=\text{NUMBER}$，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_N

V_T

文法： 语言的定义

1. 语言的非形式化定义

文法G描述的语言$L(G)$:

从G的开始符号S出发，

反复使用产生式的右部**替换**左部，

最后所能得到的终结符号串的全体。

替换

- 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： $\lambda => \mu$ 。

- 直接推导序列

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

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

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

文法：　语言的定义

■句型

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

■句子

对文法$G[S]$，若$S \overset{*}{\Rightarrow} \alpha(\alpha \in V_T^*)$，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

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

句子

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

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 $\}$

例： 设有文法G

$$S \rightarrow P \mid aPb$$

$$P \rightarrow ba \mid bQa$$

$$Q \rightarrow ab$$

写出该文法所描述的语言。

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。

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：　$L(G)=\{ba, baba, abab, ababab\}$

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，$A \to \gamma$ 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。

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
- 直接递归
- 间接递归
- 左（右）递归

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

其中有$E \rightarrow E$ …这样的产生式，所以文法G_1是直接左递归文法。

文法中存在递归，推导中就会出现无止境的替换，从而有穷规则描述无限集合成为可能

例： 设有文法G_2：

$$T \rightarrow Qc \mid c$$

$$Q \rightarrow Rb \mid b$$

$$R \rightarrow Ta \mid a$$

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

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： $\boxed{S \rightarrow S0}\ |\ 0$ 求 $L(G)$？

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

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

S后面为0^n（$n≥1$）

G'： $S \rightarrow 0S \mid 0$

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

S替换为空串

例： 设有文法 G：

$S \rightarrow 0S1 \mid \varepsilon$ 求L(G)？

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

S前面为0^n,后面为$1^n(n\geq1)$

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

例： 设有语言 $L(G_1)= \{ a\, b^n\, a \mid n>=0 \}$，

给出文法G_1？

使用递归定义一个语法成分

$G_1(S)$ ： $S \rightarrow aRa$ $R \rightarrow Rb \mid \varepsilon$

$G_1(S)$ ： $S \rightarrow aT$ $T \rightarrow Ra$ $R \rightarrow Rb \mid \varepsilon$

$G_1(S)$ ： $S \rightarrow aa \mid aRa$ $R \rightarrow b \mid Rb$

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

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

文法和语言之间的相互转换举例

例：写出文法 G 　$\boxed{S \to Sb} \mid R$

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

S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

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

写出该语言的文法。

一个语法成分 (非终结符）描述出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

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

写出该语言的文法。

语言集合{*ab*, *abab*, *ababab*……}

G： $S \rightarrow abS \mid ab$

定义在字母表{*a*,*b*}上的语言

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

语言集合{*ab*, *aabb*, *aaabbb*……}

G： $S \rightarrow aSb \mid ab$

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

一个语法成分 (非终结符）描述

$S \rightarrow 1S0 \mid B$

$B \rightarrow 0B1 \mid \varepsilon$

$G(S)$：　$S \rightarrow 1S0 \mid B$

$B \rightarrow 0B1 \mid \varepsilon$

文法和语言之间的相互转换举例

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

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

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

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

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

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

文法和语言之间的相互转换举例

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

$T\to aTb|aTbb|\varepsilon$

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

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

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

$a^ib^j=a^{m+n}b^{2m+n}=a^ma^nb^nb^{2m}$

令：$j=i+m+n$

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

设语言$S=\{a^ib^j|0\le i\le j\le 3i\}$，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

$S \rightarrow E|-E$

$E \rightarrow A|BA|BDA$

$D \rightarrow CD|C$

$C \rightarrow 0|B$

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

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

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

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

$P \to 0PP \mid 1S$

$Q \to 0S \mid 1QQ$

$S \to \varepsilon \mid 0S1S \mid 1S0S$

2.1 文法和语言

2.1.1 语言的语法和语义

2.1.2 文法和语言的定义

2.1.3 文法的表示方法

2.1.4 语法树与二义性

2.1.5 文法和语言的类型

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

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

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

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>

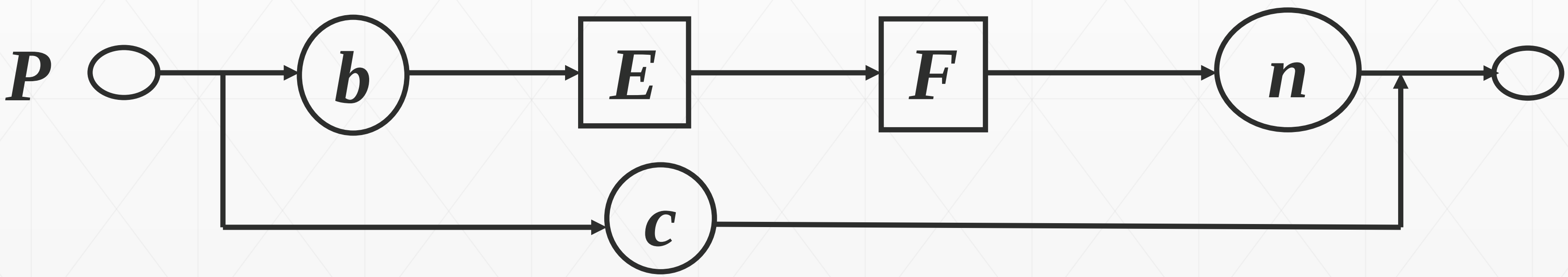
<条件语句> → <如果子句> [else<语句>]

文法的表示方法

3. 语法图(上下文无关文法）

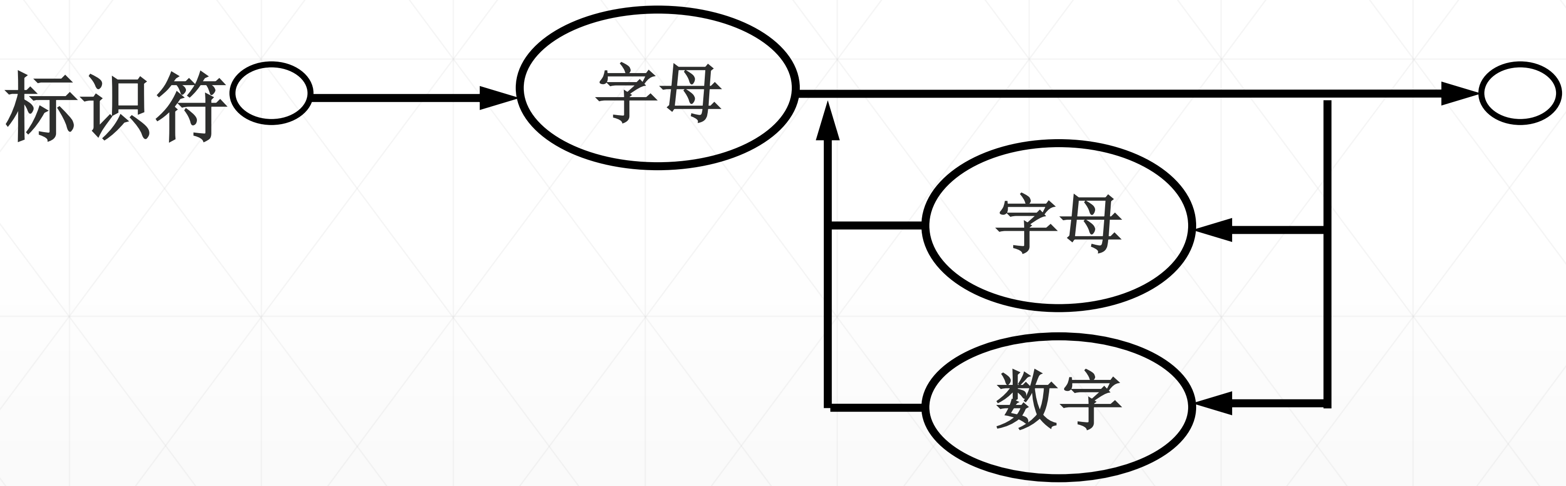
- 用图描述产生式规则
- 每个图都有一个起始结点和一个终止结点，其他的结点标记为文法符号。
- 终结符结点用圆形表示。
- 非终结符结点用方形表示。
- 从起始结点到终止结点的路径(标记为结点序列)定义候选式。

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

推导过程的形象化表示形式

给出了语法结构的图形表示，便于理解语法结构层次。

推导过程对应为树的生长过程

- 表示：

分析树的根结点	⟷	*G*的*S*
父子结点关系	⟷	产生式规则
父结点	⟷	产生式左部
子结点(个数)	⟷	侯选式(侯选式长度)
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

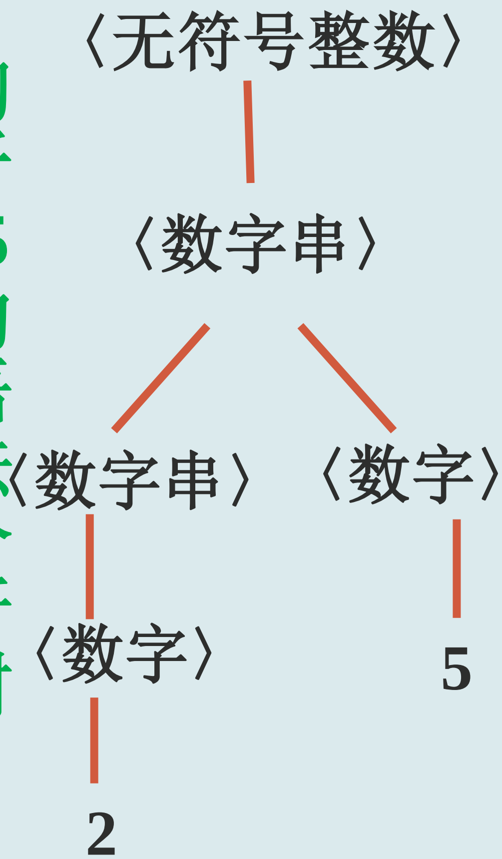
句子25的最右推导：

<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。

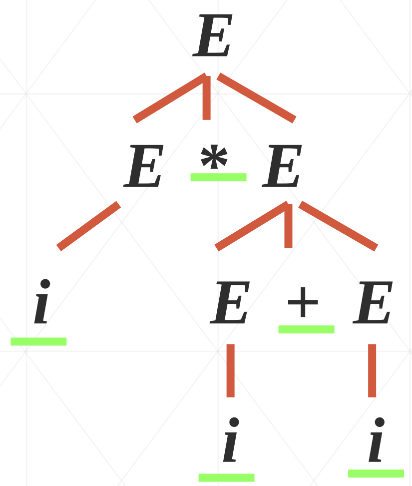
包含有二义性句子（或句型）的文法称为二义文法。

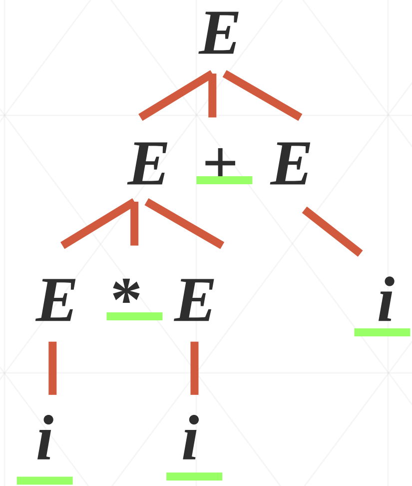
否则，该文法是无二义性的。

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$i*i+i$存在两棵不同的分析树，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

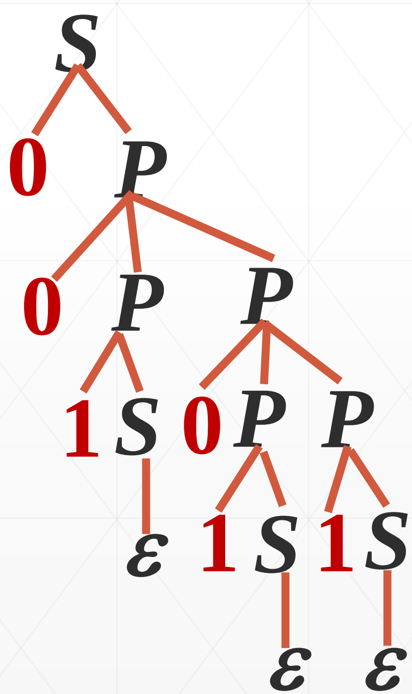
例: 设集合A为字母表Σ={0,1}上的有相同个数的0和1组成的符号串集合，给出正确描述集合A的文法。

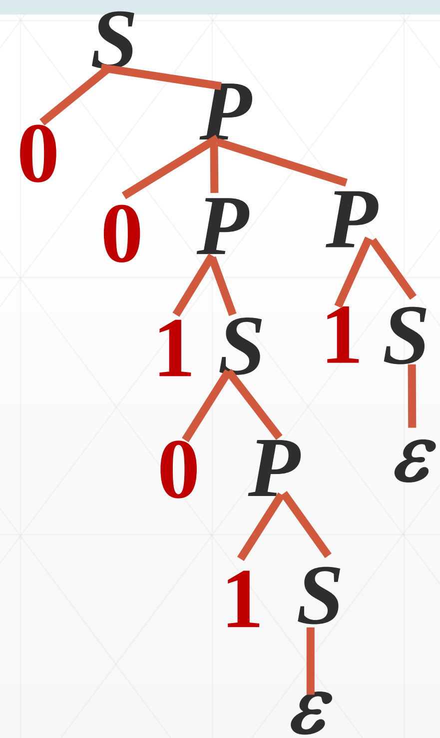
$S \to \varepsilon \mid 0P \mid 1Q$

$P \to 0PP \mid 1S$

$Q \to 0S \mid 1QQ$

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. 既有语法又有语义的二义性

i+i+i 有语法的二义性但无语义的二义性

你真可以。 有语义的二义性但无语法的二义性

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

$$G_1': \quad E \rightarrow T \mid E+T$$

$$T \rightarrow F \mid T*F$$

$$F \rightarrow (E) \mid i$$

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$\alpha\rightarrow\beta$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

$S\rightarrow 0|AC0B$

$\boxed{C0\rightarrow 00C}$

$CB\rightarrow DB|E$

$0D\rightarrow D0$

$AD\rightarrow AC$

$0E\rightarrow E0$

$AE\rightarrow \varepsilon$

G是0型文法

$L(G)=\{0^n\,|\,n$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

$S \rightarrow aSBC \mid abC$

$CB \rightarrow CD$

$CD \rightarrow BD$

$BD \rightarrow BC$

$bB \rightarrow bb$

$bC \rightarrow bc$

$cC \rightarrow cc$

$CB \Rightarrow BC$

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

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

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

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

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

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

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

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

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：

$$A \to \alpha$$

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。

文法和语言的类型

例： 设有文法G_2

$$S \to Ac \mid Sc$$

$$A \to ab \mid aAb$$

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

$$A \to \alpha B \qquad \text{或} A \to \alpha$$

左线性文法：每个产生式形如：

$$A \to B\alpha \qquad \text{或}\ A \to \alpha$$

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

$$S \rightarrow Sc \mid Bc$$

$$B \rightarrow Bb \mid Ab$$

$$A \rightarrow Aa \mid a$$

G_3是左线性文法

G_3是3型文法

G_3产生的语言：

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

例：设有文法G_3

$$S \to aB \mid c$$

$$B \to Sb \mid b$$

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

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

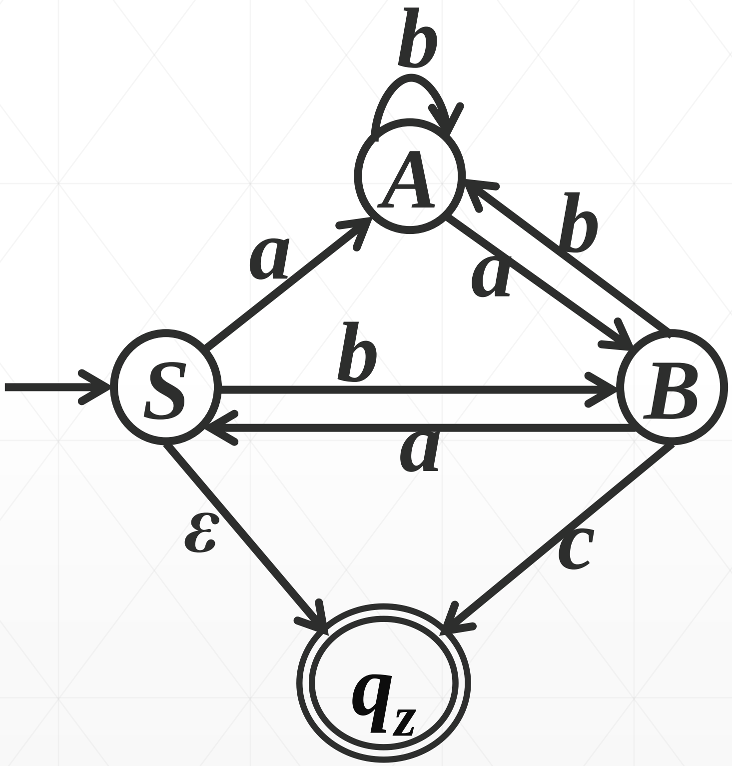
右线性文法⇒FA

- **例：** 写出与下述文法等价的有限自动机M

$S \rightarrow aA|bB|\varepsilon$

$A \rightarrow aB|bA$

$B \rightarrow aS|bA|c$



3型文法与FA的转换

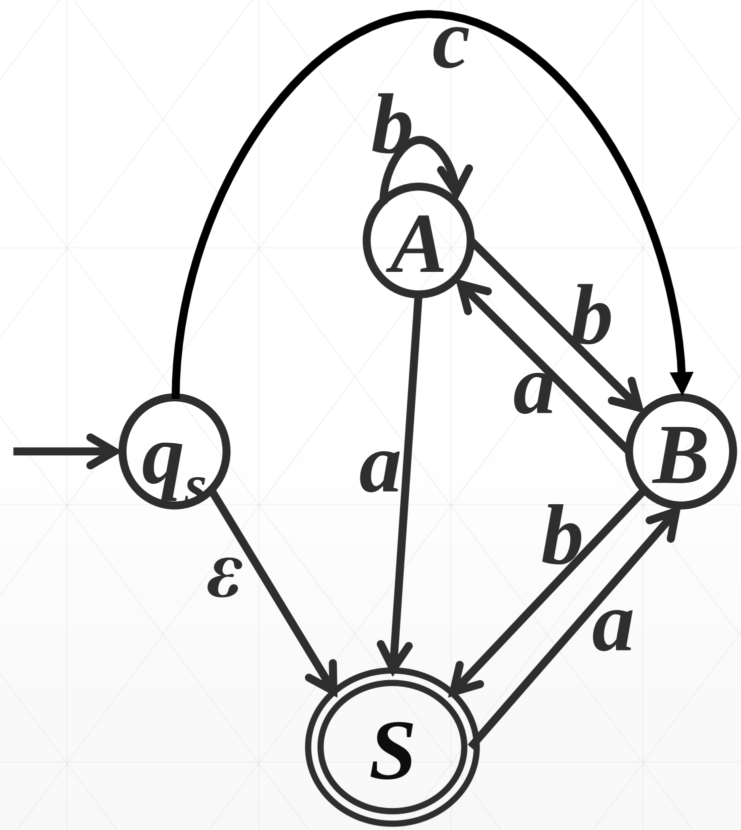
左线性文法⇒FA

- **例：** 写出与下述文法等价的有限自动机M

$S \rightarrow Aa|Bb|\varepsilon$

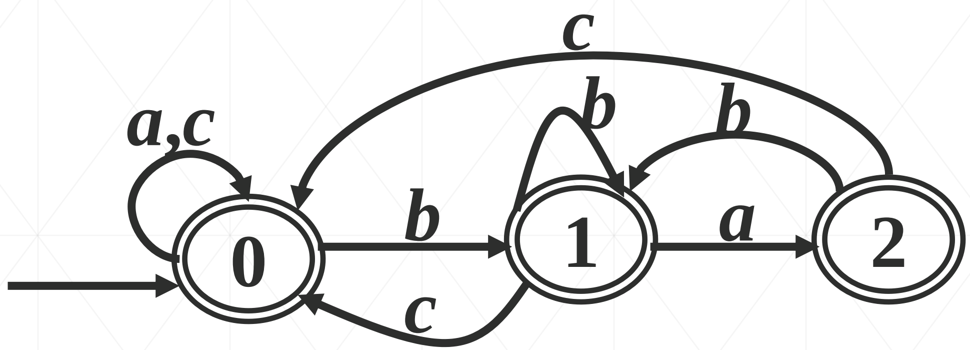
$A \rightarrow Ba|Ab$

$B \rightarrow Sa|Ab|c$



FA⇒线性文法

- **例：** 写出与下述有限自动机M等价的文法



右线性：

$S \rightarrow aS|cS|bA|\varepsilon$

$A \rightarrow aB|bA|cS|\varepsilon$

$B \rightarrow cS|bA|\varepsilon$

左线性：

$S \rightarrow A|B|C$

$A \rightarrow Aa|Ac|Bc|Cc|\ \varepsilon$

$B \rightarrow Ab|Bb|Cb$

$C \rightarrow Ba$