第2章

形式语言概论

本课程：文法；自动机，正规表达式

作业

- 理论：Cconline.sysu.edu.cn
- 编译器构造实验：

- 转文法
- QQ:**597371232**
- 实验报告提交：1967074105@qq.com
- isscjh@mail.sysu.edu.cn

2.1 语言成分

一个语言的成分包括字母表（Alphabet），文法(Grammar)以及它的语义。

本章将主要讨论字母表、符号串、产生式文法系统以及句子分析等方面的内容。

2.1 语言成分

- 字母表与符号 ：字母表是元素的非空有穷集合。字母表中的元素称为符号。
- 符号串及其运算 ：

① 符号串。
② 符号串的长度。
③ 符号串的连接。
④ 符号串集合的乘积。
⑤ 符号串的方幂。
⑥ 符号串集合的方幂。
⑦ 符号串集合的正闭包。
⑧ 符号串集合的自反闭包。

字母表与符号串

- **字母表**：是符号的非空有穷集合，用Σ表示
- **符号串**：由字母表中的符号所组成的任何有穷序列被称之为该字母表上的符号串

  设有字母表Σ={a，b，c}，那么：

  序列ab是Σ上的一个符号串；

  同样序列ba，序列abc；

  序列bcca等都是Σ上的符号串

符号串的长度 | s |

- 在语言的理论中，术语“句子”和“字”常常用作术语“符号串”的同义语。
- 符号串s的长度记作| s |，是组成该符号串的符号的个数。
- 例如，上述Σ上的符号串ab的长度是2，记作 | ab |=2。
- 符号串abc的长度是3，记作| abc |＝3。
- 空符号串记作 ε，它由零个符号组成，于是 | ε |＝0。

与符号串有关的几个术语(1)

- 符号串s 的前缀：移走符号串s的尾部的零个或多于零个符号所得到的一个符号串。

  例如ban是符号串banana的一个前缀。
- 符号串 s 的后缀：删去符号串s的头部的零个或多于零个符号所得到的一个符号串。

  例如nana是符号串banana的一个后缀。
- 符号串s的子串：从 s 中删去一个前缀和一个后缀而得到的符号串。

与符号串有关的几个术语(2)

- 符号串s的真前缀、真后缀、真子串：任何非空符号串 x ，相应地，是s的前缀、后缀或子串，并且s≠x。
- 符号串 s 的子序列：从符号串 s 中删去零个或多于零个符号(这些符号不要求是连续的)而得到的符号串。
- 术语“语言”表示某个确定的字母表上的符号串的任何集合。

符号串的运算 (1)

- 符号串的连接：设x , y 是符号串，则 xy 称为 x 与 y 的连接
- 符号串集合的乘积：设 A，B是符号串集合，AB表示A 与B的乘积，则定义 $AB=\{xy|\ x\in A, y\in B\}$
- 符号串的方幂。同一符号串的连接可写成方幂形式．设x是一符号串，则定义

  $x^0=\varepsilon$

  $x^1=x$

  $x^2=xx$

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

  例如，x=abc，x^2=abcabc，x^3=abcabcabc

符号串的运算 (2)

- 符号串集合的方幂。同一符号串集合的乘积也可以写成方幂形式设符号串集合A，则定义

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

$A^1=A$

$A^2=AA$

$A^3=A^2A=AA^2$

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

例如，A={a，bc}，A^2=AA={aa，abc，bca，bcbc}，$A^3=A^2A$={aaa，abca，bcaa，bcbca，aabc，abcbc，bcabc，bcbcbc}

符号串的运算 (3)

- 符号串集合的正闭包：设符号串集合A，A的正闭包记为A^+，则有

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

  即A^+为集合A上所有符号串的集合
- 符号串集合的自反闭包：符号串集合A的自反闭包记为A^*，则有

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

语言之上的几个重要运算

假设L和M表示两个语言

- 语言L和M的合并(union)，记作$L \cup M$，定义为：
  $L \cup M=\{S|S \text{ is in L or S is in M}\}$
- 语言L和M的连接(concatenation),记作LM,定义为：
  $LM=\{st|s \text{ is in L and t is in M}\}$
- 语言L的Kleene闭包，记作L^*，定义为：
  $L^*=\cup L^i=L^0 \cup L^1 \cup L^2 \cup L^3$.

文法与语言的形式定义

- 定义 文法G是一个四元组：

  $G=(V_T，V_N，P，S)$

  V_T：终结符集；

  V_N：非终结符集；

  P：产生式集；

  S：开始符号；

算术表达式

- 表达式是由运算符连接运算量的式子
- 表达式→表达式+项|表达式-项|项

  项→项×因子|项÷因子|因子

  因子→运算量|(表达式)
- G[E]=(V_N,V_T,P,E)

  V_N={E,T,F}

  V_T={+,-,*,/,i,(,)}

  P={E→ E+T|E-T|T,

  T→ T*F|T/F|F, F→ i|(E)}

文法及其分类 文法：$G(V_N, V_T, P, S)$

- 0型文法：P: $\alpha \rightarrow \beta$

  其中：$\alpha \in (V_N \cup V_T)^+$, $\beta \in (V_N \cup V_T)^*$
- 1型文法（上下文有关文法）：

  P: $\alpha \rightarrow \beta$ 其中：$|\alpha| \leq |\beta|$

  或$\gamma_1 A \gamma_2 \rightarrow \gamma_1 \delta \gamma_2$ ，γ_1，$\gamma \in (V_N \cup V_T)^*$

  $A \in V_N$ $\delta \in (V_N \cup V_T)^+$
- 2 型文法（上下文无关文法）：

  P: $A \rightarrow \delta$ $A \in V_N$ $\delta \in (V_N \cup V_T)^+$
- 3型文法（右线性文法、正规文法）：

  P: $A \rightarrow a|aB$ $A,B \in V_N$ $a \in V_T$

0型文法

在2.2.1节中定义的文法即为0型文法(无限制的文法)。其产生式具有以下形式：

$$\alpha \rightarrow \beta$$

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

1型文法（上下文有关文法）

1型文法G的产生式具有以下形式：　　$\alpha \rightarrow \beta$

其中：　$\alpha=\gamma_1 A\gamma_2$；

$\beta=\gamma_1\delta\gamma_2$；

γ_1，$\gamma_2 \in (V_N \cup V_T)^*$；

$A \in V_N$；$\delta \in (V_N \cup V_T)^+$。

1型文法（上下文有关文法）举例(1)

- 条件 P: $\alpha \rightarrow \beta$ 其中：$|\alpha| \leqslant |\beta|$
- $G[S]=(V_N,V_T,P,S)$

  $V_N=\{S,A,B,C\}$

  $V_T=\{a,b,c\}$

  $P=\{S \rightarrow aSBC,\ S \rightarrow aBC,\ aB \rightarrow ab,$

  $bB \rightarrow bb,\ bC \rightarrow bc,\ CB \rightarrow BC,\ cC \rightarrow cc\}$
- 或条件 P: $\gamma_1 A \gamma_2 \rightarrow \gamma_1 \delta \gamma_2$

  $CB \rightarrow BC$

  改为 $CB \rightarrow CD$，$CD \rightarrow BD$，$BD \rightarrow BC$

1型文法（上下文有关文法）举例(2)

- $G[S]=(V_N,V_T,P,S)$

  $V_N=\{S,X,Y,Z\}$

  $V_T=\{x,y,z\}$

  $P=\{S\rightarrow xSYZ|xYZ,$

  $xY\rightarrow xy,$

  $yY\rightarrow yy,$

  $yZ\rightarrow yz,$

  $ZY\rightarrow YZ,$

  $zZ\rightarrow zz\}$

2型文法（上下文无关文法）

在1型文法的产生式中上下文γ_1和γ_2用空符号串ε代替，则有以下形式的产生式称为2型文法：

$A \rightarrow \delta$

其中，$A \in V_N$，$\delta \in (V_N \cup V_T)^+$。

2型文法（上下文无关文法）举例(1)

- 条件 $P:A \rightarrow \delta \quad A \in V_N,\ \delta \in (V_N \cup V_T)^+$
- $G[E]=(V_N,V_T,P,E)$

  $V_N=\{E,T,F\}$

  $V_T=\{+,-,*,/,i,(,)\}$

  $P=\{E \rightarrow E+T|E-T|T,$

  $T \rightarrow T*F|T/F|F,\ F \rightarrow i|(E)\}$
- 表达式是由运算符连接运算量的式子
- 表达式→表达式+项|表达式-项|项

  项→项×因子|项÷因子|因子

  因子→运算量|(表达式)

2型文法（上下文无关文法）举例(2)

- G4[N]

  P: N→ ND |D;

  D→ 0|1|2|3|4|5|6|7|8|9;

- G7[S]

  P: S→ aTd;

  T→ bT|cT|b|c;

- G9[S]

  P: S→ 0S1|01 ;

3型文法（右线性文法和正规文法）

在正规文法中，P中的每个产生式(S→ε例外，S为文法的开始符号)只有两种形式：A→a ，A→aB 。

其中 A，B∈V_N ，a∈V_T。此外，如果S→ε是P中的一个产生式，那么S不能出现在任何产生式的右边。

例 正规文法G_5(S)

S→dB|+A|-A| G

A→dB| G

B→dB| H|d

G→dH

H→dH|d 其中d代表十进制数字。

3型文法（右线性文法、正规文法）举例

- 条件 P: $A \rightarrow a|aB$ $A,B \in V_N$, $a \in V_T$
- G5[S]

  P: S→ 0 | 1 | 1A | 0B

  A→1A|0B

  B→0 | 1 | 0B
- G4[N]

  P: N→ 0|1|2|3|4|5|6|7|8|9 |

  0N|1N|2N|3N|4N|5N|6N|7N|8N|9N;

推导和规范推导(1)

- **定义2.3：** 文法 $G=(V_N, V_T, P, S)$，$\alpha\to\beta\in P$，$\gamma,\delta\in(V_N\cup V_T)^*$，则称符号串$\gamma\beta\delta$为$\gamma\alpha\delta$用产生式$\alpha\to\beta$的直接推导，记为

$$\gamma\alpha\delta \Rightarrow \gamma\beta\delta$$

$\overset{+}{\Rightarrow}$ 多步推导

$\overset{*}{\Rightarrow}$ 0步或多步推导

推导和规范推导(2)

•**定义2.6:最左推导:**在 $xUy \Rightarrow xuy$ 的直接推导中，若x∈VT*, U∈VN——U是符号串 xUy中的最左非终结符，则称此直接推导为最左直接推导。每一步都为最左直接推导。

• **定义2. 7:最右推导:**在$xUy \Rightarrow xuy$ 的直接推导中，若y∈VT*, U∈VN——U是符号串 xUy中的最右非终结符，则称此直接推导为最右直接推导。每一步都为最右直接推导。

• **最右(直接)推导又称为规范推导**

最左推导与最右推导举例

- 文法 G[E]

  E→ E+T|T;

  T→ T*F|F;

  F→ i|(E);

- 写出i+(i+i)的最左推导与最右推导
- 解：最左推导：E=>E+T=>T+T=>F+T=>i+T

  =>i+F=>i+(E) =>i+(E+T) =>i+(T+T) =>i+(F+T)

  =>i+(i+T) =>i+(i+F) =>i+(i+i)

  最右推导: E=>E+T =>E+F =>E+(E) =>E+(E+T)

  =>E+(E+F) =>E+(E+i) =>E+(T+i) =>E+(F+i)

  =>E+(i+i) =>T+(i+i) =>F+(i+i) =>i+(i+i)

最左推导与最右推导举例

1. 确定下列文法的语言

G[S]：S→ABC|C;

A→1|2|3|4|5|6|7|8|9|;

B→N|BN | ε

C→0|5

N→A| 0

2. 已知文法

G[Z]：Z→UV | VU

U→Z1 | 1

V→Z0 | 0

试写出01010110的最左推导和最右推导

- G[Z]：Z→UV | VU

  U→Z1 | 1

  V→Z0 | 0

写出01010110的最左和最右推导

- 解：最左推导：Z=>UV=>Z1V

  =>VU1V=>Z0U1V=>VU0U1V

  =>0U0U1V=>010U1V=>010Z11V

  =>010UV11V=>0101V11V

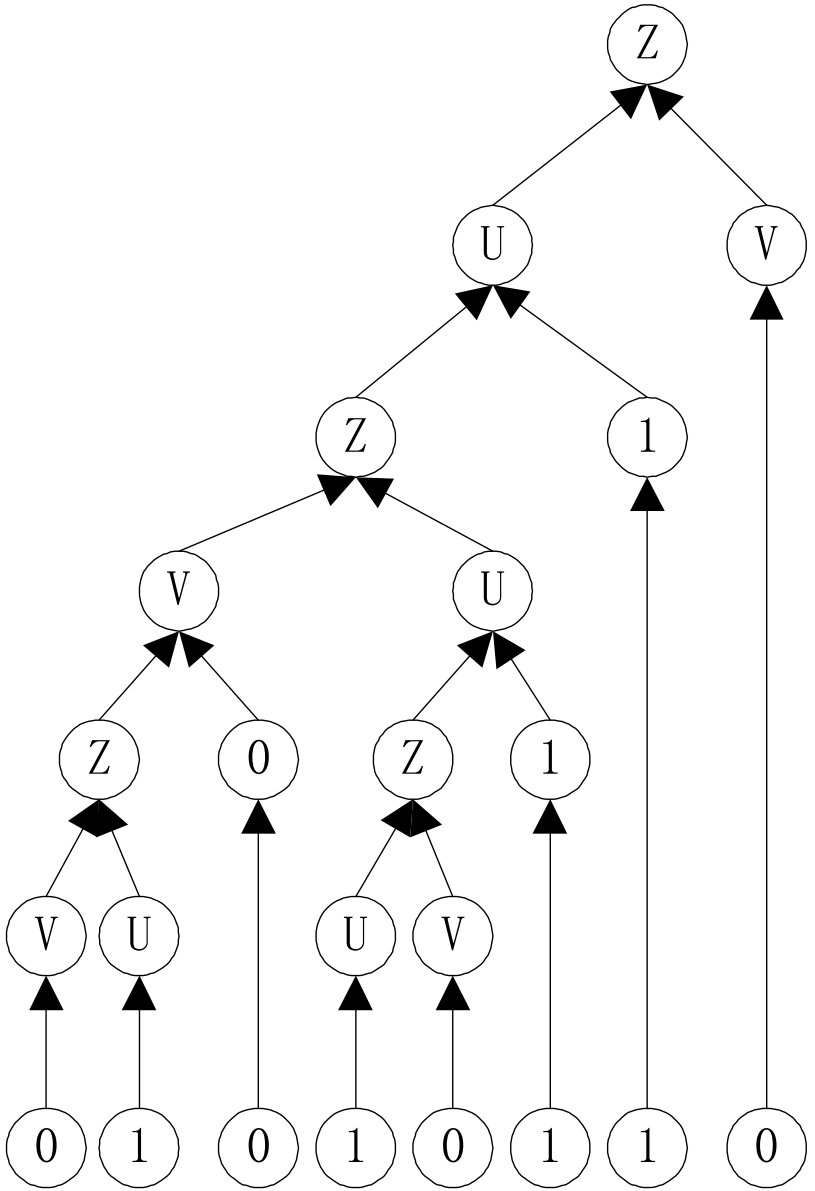
  =>0101011V=>01010110

  最右推导：Z=>UV=>U0=>Z10

=>VU10=>VZ110=>VUV110=>VU0110

=>V10110=>Z010110=>VU010110

=>V1010110=>01010110

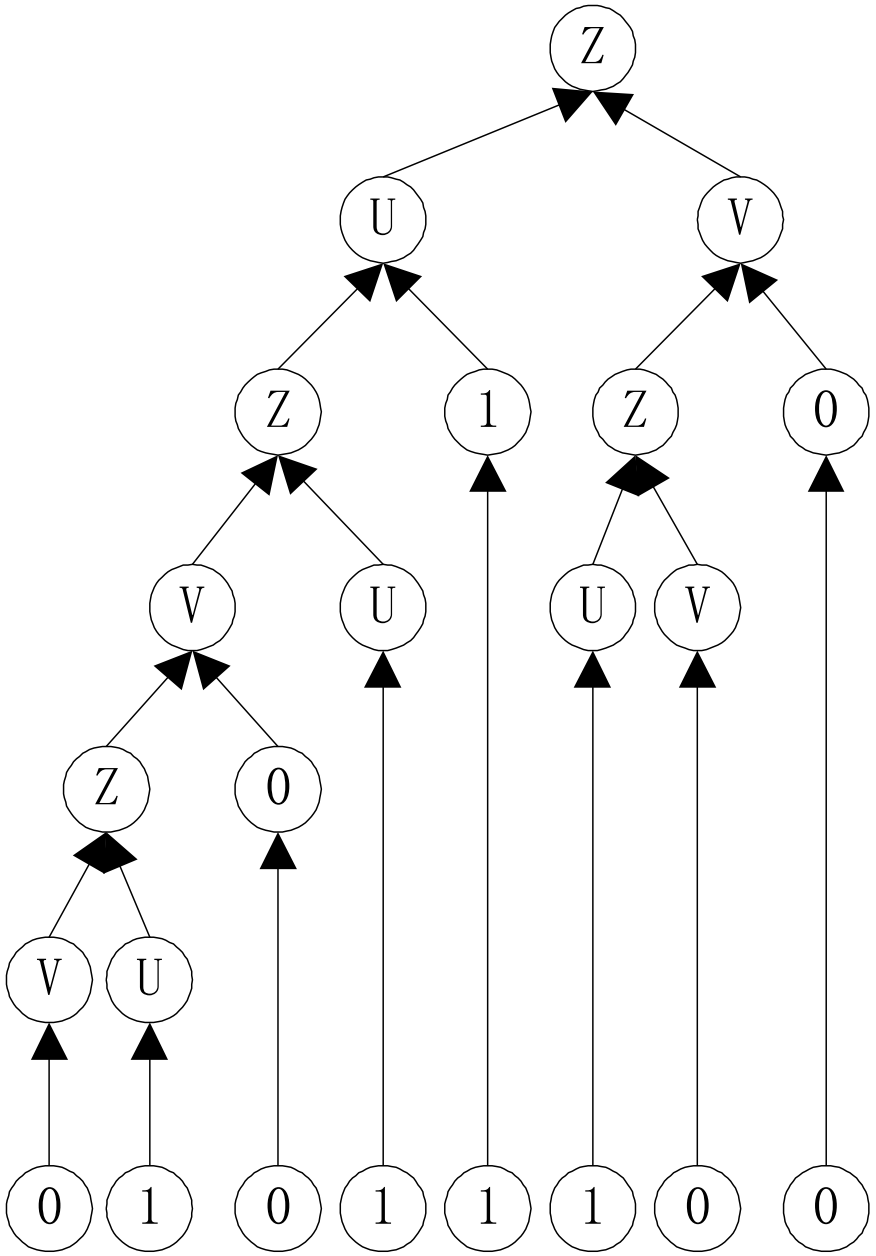


- G[Z]：Z→UV | VU

  U→Z1 | 1

  V→Z0 | 0

写出01011100的最左和最右推导



句型、句子和语言

- **定义2.8**:设S是文法G的开始符号，如果 $S \overset{*}{\Rightarrow} u$，$u \in (V_N \cup V_T)^*$，称u为文法G的**句型**。

- **定义2.9**:设S是文法G的开始符号，如果 $S \overset{*}{\Rightarrow} u$，$u \in V_T^*$，称u为文法G的**句子**。

- **定义2. 10**：设S是文法G的开始符号，文法G的**语言**

$$L(G) = \{u \mid S \overset{*}{\Rightarrow} u, u \in V_T^*\}$$

短语与句柄

- **定义6.1**: 设S是文法G的开始符号，

若 $S \overset{*}{\Rightarrow} xUy$ $U \overset{+}{\Rightarrow} \alpha$

$U \in V_N$, x,y∈（$V_N \cup V_T$）*, 称α是句型xUy相对于U的**短语**。

又若 $S \overset{*}{\Rightarrow} xUy$ $U \Rightarrow \alpha$

$U \in V_N$*， x,y∈（$V_N \cup V_T$）*， 称α是句型xUy相对于U的**直接短语**或**简单短语**。

一个句型的最左直接短语称该句型的**句柄**。

构造下列语言的文法

- $\{ a^{3n} | n \geq 1 \}$;

  解：G[S]: $S \rightarrow aaaS | aaa$

- 偶数集合

  解：G[N]: $N \rightarrow AB$

  $A \rightarrow DA | \varepsilon$

  $D \rightarrow 0|1|2|3|4|5|6|7|8|9$

  $B \rightarrow 0|2|4|6|8$

- $\{a^n b^m c^k | n,m,k \geq 0\}$

解：G[S]: $S \rightarrow ABC$

$A \rightarrow aA|\varepsilon$　$B \rightarrow bB|\varepsilon$　$C \rightarrow cC|\varepsilon$

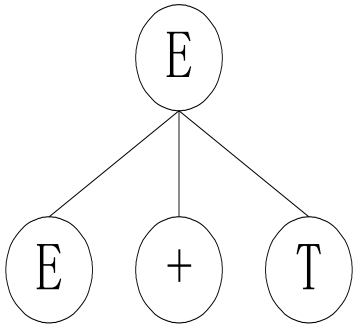
产生式树举例

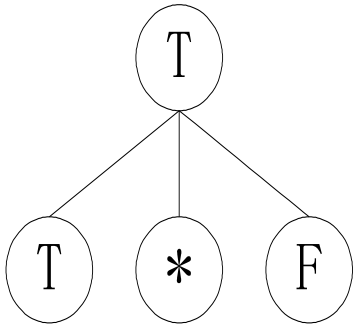
- 文法 G[E]

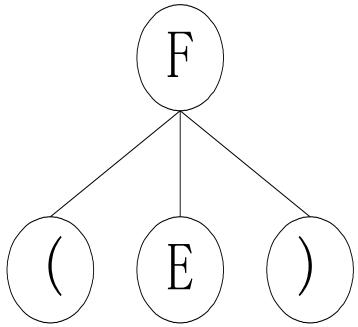
  E→ E+T|T；

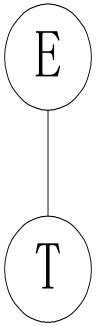
  T→ T*F|F；

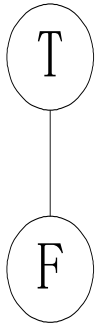
  F→ i|(E)；











语法树

设文法 G=(VT，VN，P，S)，对于文法G的任意一个句型都存在一个相应的语法树：

① 树中每个结点都有一个标记，该标记是$V_N \cup V_T$中的一个符号；

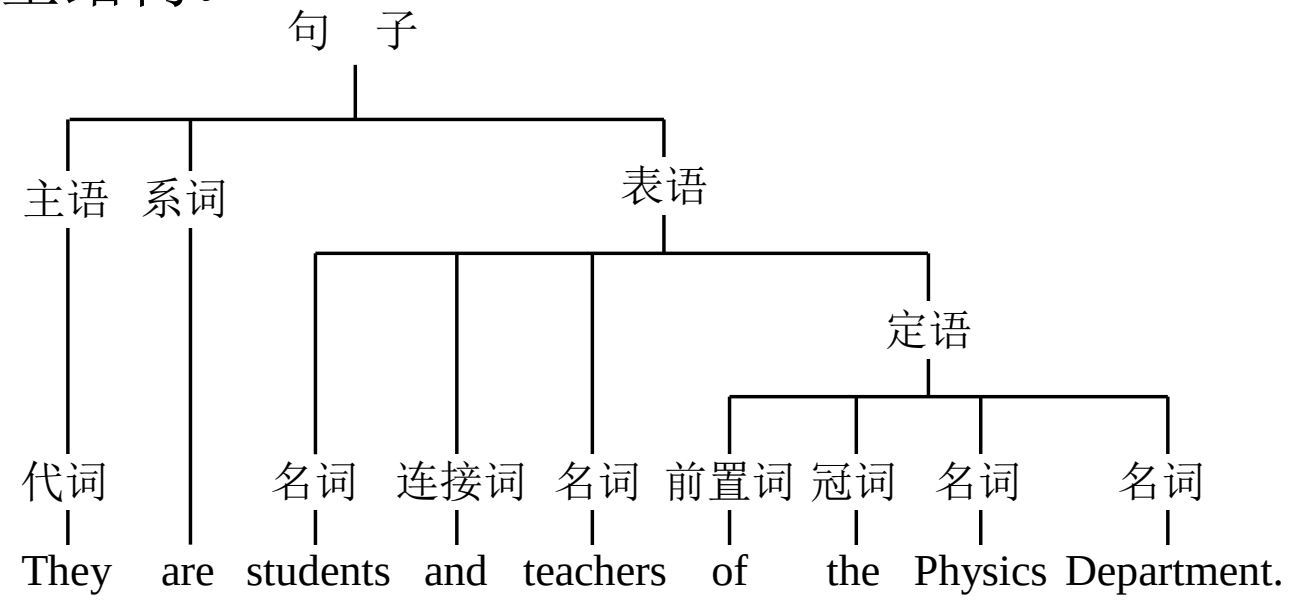
② 树的根结点标记是文法的识别符号S；

③ 若树的一个结点至少有一个叶子结点，则该结点的标记一定是一个非终结符；

④ 若树的一个结点有多个叶子结点，该结点的标记为A，这些叶子结点的标记从左到右分别是B1，B2，…，BN，则(A，B1，B2…BN) ∈P。

语法树

在自然语言中，可通过树型表示直观地分析句子结构；在形式语言中，则是通过语法树直观地分析文法的句型结构。



句子结构

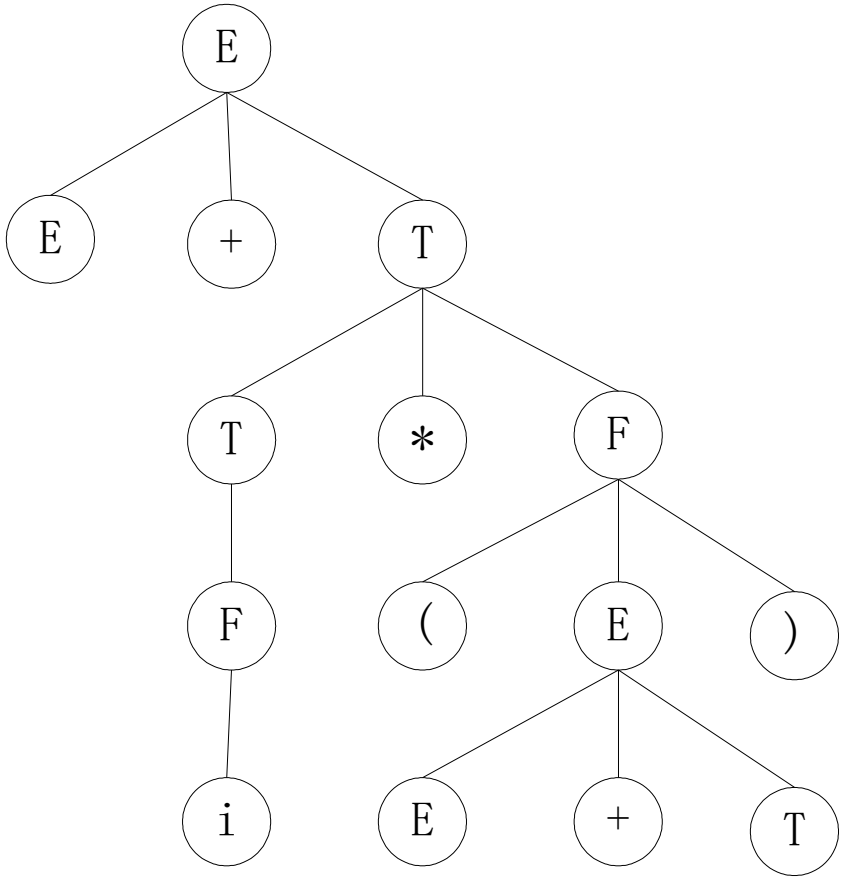
语法树举例

- 文法 G[E]

  E→ E+T|T；

  T→ T*F|F；

  F→ i|(E)；
- E+i*(E+T)



产生式树

- 文法的句型可依据文法的产生式树来生成相应的语法树。

产生式树

文法的句型都可依据文法的产生式来生成相应的语法树。以句型E+(E+T)*i为例：

① 以文法G的识别符号E作为语法树根结点的标记。选择识别符E的一个产生式E→E+T，然后生成根结点E的3个分支，根结点E的3个叶子结点的标记，从左到右分别记为E、＋和T。

② 选择产生式T→T*F，生成以结点T作为根结点的子树。

③ 选择产生式F→i，以②中子树最右边的叶子结点F为根结点，延伸相应的子树。

④ 选择产生式T→F，以③中子树的叶子结点T为根结点，延伸相应的子树。

⑤ 选择产生式F→(E)，以④中子树的叶子结点F为根结点，延伸相应的子树，扩充相应的子树。

无用V_N、不可达V_N

- 文法中规则的个数应该是恰当的。再少，不足以完全描述一个语言，再多，又无必要。
- 程序设计语言中存在有多余规则时，往往存在有错误
- 多余规则使句子的分析复杂和增加困难度。

明显的多余规则有两种。

- 一是形如U→U的单规则，引起文法上的二义性。应该首先把它们删除去。
- 另一是规则U→u，其左都非终结符号U不出现在其它任何规则的右部，也应删除。（U例外）。

无用V_N、不可达V_N举例

例如，对于文法G [Z]：

Z→Be

A→Ae | A | e

B→Ce|Af

C→Cf

D→f

首先删除规则A→A;

又由于D不出现在任何规则右部，应删去规则D。

规则不多余的条件：U

- **条件1** $S \overset{*}{\Rightarrow} xUy$ $x,y \in (V_N \cup V_T)^*$

- **条件2** $U \overset{+}{\Rightarrow} \alpha$ $\alpha \in V_T^*$。

条件1要求U在句型中出现，条件2则进一步要求能从U推导出终结符号串。

显然，如果存在某个非终结符号U，它不满足上述两条件，则以U为左部的那些规则必是多余的。

二义性

- 定义2.13 一个文法，如果它的一个句子有两棵或两棵以上的语法树，则称此句子具有二义性。如果一个文法含有二义性的句子，则该文法具有二义性。

分析方法简介

- 一个分析器或分析自动机是这样一种系统，它能够根据给定的文法G，构造语言L(G)的任意推导。分析也可看作是语法树的构造过程。我们主要讨论右线性文法和CFG的分析器。

- 分析的方法很多，可归纳为两类，一类是自上而下分析方法，另一类是自下而上分析方法。

2.6.1自上而下分析方法

自上而下分析方法的基本思想是从文法的开始符号出发，利用其中的产生式，逐步推导出待分析的符号串。如果能推导出这个符号串，则表明此符号串是该文法的一个句型或句子，否则便不是。

2.6.2 确定的自上而下分析方法

- 当文法的某一个非终结符有几条产生式、而且每条产生式右部首符号都是终结符时，应保证它们是互不相同的终结符。
- 例　设文法$G_{18}[S]$：

$S \rightarrow aBc \mid bCd$

$B \rightarrow eB \mid f$

$C \rightarrow dC \mid c$

试检查符号串aefc是不是该文法的句子。

2.6.2 确定的自上而下分析方法

上例推导过程：

$S \rightarrow aBc$，$S \Rightarrow aBc$；

$B \rightarrow eB$，$S \Rightarrow aBc \Rightarrow aeBc$；

$B \rightarrow f$，$S \Rightarrow aBc \Rightarrow aeBc \Rightarrow aefc$；

该例属确定的自上而下分析方法。

2.6.3 自下而上分析方法

自下而上分析方法的基本思想是从待检查的符号串出发，看最终是否能归约（推导的逆过程）到文法的识别符号。如果能归约到文法的识别符号，则表明此待检查的符号串是该文法的一个句型或句子，否则便不是。

2.6.4 文法在内存中的表示

用语法图的表格结构表示文法。一个文法的语法图由该文法所有非终结符号的定义图组成。每个非终结符号的定义图是一个结构型数据。

在自上而下分析方法中，用这种语法图表示文法有利于消除左递归，也有助于提取左因子。

分析方法简介

- 自上而下分析方法
- 确定性的自上而下分析方法
- 自下而上分析方法