

讲稿（教学内容、步骤）

第2章 文法和语言（2）

1. 高级程序语言为什么使用上下文无关文法？

因为它有足够的能力描述现今程序设计语言的语法结构。

【举例】

算术表达式的上下文无关文法

$E \rightarrow E+E$

$E \rightarrow E * E$

$E \rightarrow (E)$

$E \rightarrow i$

条件语句上下文无关文法

$\langle \text{条件语句} \rangle \rightarrow \text{if } \langle \text{条件} \rangle \text{ then } \langle \text{语句} \rangle$

$\langle \text{条件语句} \rangle \rightarrow \text{if } \langle \text{条件} \rangle \text{ then } \langle \text{语句} \rangle \text{ else } \langle \text{语句} \rangle$

2. 上下文无关文法的语法树：用于描述上下文无关文法的句型推导的直观方法

【举例】

$G[S]: S \rightarrow aAS$

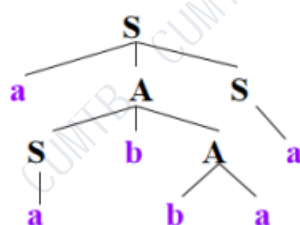
$A \rightarrow SbA$

$A \rightarrow SS$

$S \rightarrow a$

$A \rightarrow ba$

句型 **aabbba** 的语法树（推导树）



- 叶子结点：树中没有子孙的结点。
- 语法树（推导树）：从左到右读出推导树的叶子标记，所得的句型为推导树的结果。也把该推导树称为该句型的语法树。
- 推导过程中施用产生式的顺序：

$S \Rightarrow aAS \Rightarrow aAa \Rightarrow aSbAa \Rightarrow aSbbba \Rightarrow aabbba$ 最右推导（规范推导）

$S \Rightarrow aAS \Rightarrow aSbAS \Rightarrow aabAS \Rightarrow aabbaS \Rightarrow aabbba$ 最左推导

$S \Rightarrow aAS \Rightarrow aSbAS \Rightarrow aSbAa \Rightarrow aabAa \Rightarrow aabbba$

3. 二义文法

【讨论】一个句型是否对应唯一的一棵语法树？

【举例】

$G[E]: E \rightarrow i$

$E \rightarrow E+E$

$E \rightarrow E * E$

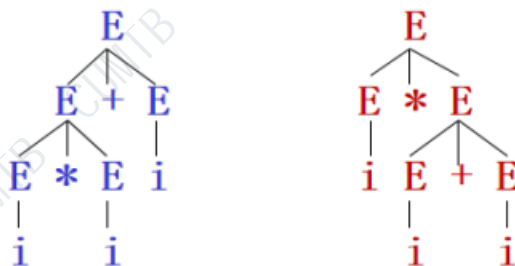
$E \rightarrow (E)$

该文法对于句型 $i * i + i$ 存在两个不同的最左推导：

推导1: $E \Rightarrow E+E \Rightarrow E * E+E \Rightarrow i * E+E \Rightarrow i * i+E \Rightarrow i * i+i$

推导2: $E \Rightarrow E * E \Rightarrow i * E \Rightarrow i * E+E \Rightarrow i * i+E \Rightarrow i * i+i$

即对于同一个句型存在两棵不同的语法树:



也就是推导过程不唯一, 称这种文法为“二义文法”。

注意:

① 描述程序设计语言语法规则的文法不能是二义文法!

② 现实中存在一些先天二义的语言

先天二义语言: 产生该语言的每一个文法都是二义的。

4. 【讨论】如何证明文法是二义的?

找出一个句子, 对应的语法树 (或最左推导过程) 不唯一。

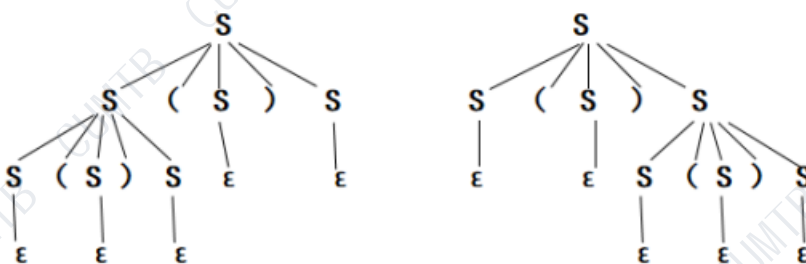
【举例】P.34 9. 文法 $S \rightarrow S(S)S | \epsilon$

(1) 生成的语言是什么

(2) 该文法是二义的吗? 请说明理由。

答: (1) 可以任意嵌套、任意连接的 $()$ 串

(2) 对于句子 $()()$ 可以构造两棵语法树, 该文法二义。



5. 改造二义文法

【举例】将 二义文法 改造为 无二义文法

$G[E]: E \rightarrow i$

$E \rightarrow E+E$

$E \rightarrow E * E$

$E \rightarrow (E)$

$G[E]: E \rightarrow T | E+T$

$T \rightarrow F | T * F$

$F \rightarrow (E) | i$

或规定优先顺序和结合律

6. 句型分析

识别一个符号串是否为某文法的句型, 是某个推导的构造过程。

7. 分析程序、分析算法

- 在语言的编译实现中,把完成句型分析的程序称为**分析程序**或识别程序。
- 分析程序所用的算法称为**分析算法**或识别算法。
- **从左到右的分析算法**:总是从左到右地识别输入符号串。首先识别符号串中的最左符号,进而依次识别右边的一个符号。
- 分析算法分类
 - ✓ 自上而下分析法
从文法的开始符号出发,反复使用各种产生式,寻找与输入符号匹配的推导。
 - ✓ 自下而上分析法
从输入符号串开始,逐步进行归约,直至归约到文法的开始符号。

两种方法反映了两种不同的语法树的构造过程。

8. 自上而下的分析

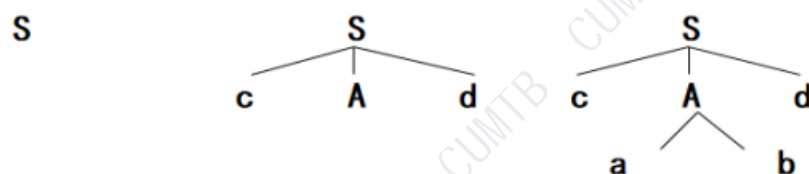
【举例】

文法 $G: S \rightarrow cAd$

$A \rightarrow ab$

$A \rightarrow a$

识别输入串 $cabd$ 是否该文法的句子。



推导过程: $S \Rightarrow cAd \Rightarrow cabd$

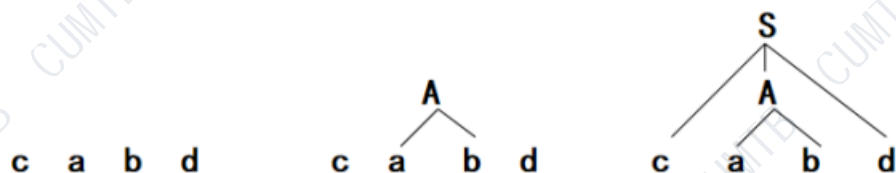
9. 自下而上的分析

【举例】文法 $G: S \rightarrow cAd$

$A \rightarrow ab$

$A \rightarrow a$

识别输入串 $cabd$ 是否该文法的句子



归约的过程: $S \Rightarrow cAd \Rightarrow cabd$

10. 句型分析的有关问题

1) 【讨论】选择使用哪个产生式进行自顶向下推导?

2) 【讨论】如何确定“可归约串”进行自下而上归约?

在自下而上的分析方法中,在分析程序工作的每一步,都是从当前串中选择一个子串,将它归约到某个非终结符号,该子串称为“可归约串”。如何确定“可归约串”?引出“句柄”的概念。

11. 短语、直接短语、句柄

• 短语

- 文法 $G[S]$, $\alpha \beta \delta$ 是 G 的一个句型, 如果:

$$S \xRightarrow{*} \alpha A \delta \quad \text{且} \quad A \xRightarrow{+} \beta$$

则称 β 是句型 $\alpha \beta \delta$ 相对于非终结符 A 的短语。

- 直接短语:** 若有 $A \Rightarrow \beta$ 则称 β 是句型 $\alpha \beta \delta$ 相对于规则 $A \rightarrow \beta$ 的直接短语 (或简单短语)。

- 句柄:** 一个句型的最左直接短语称为该句型的句柄。

【举例】求句型 $i*i+i$ 的 短语、直接短语、句柄

$$\begin{aligned} G[E]: \quad & E \rightarrow E+T \mid T \\ & T \rightarrow T*F \mid F \\ & F \rightarrow (E) \mid i \end{aligned}$$



12. 有关文法实际应用的一些说明【自学】

实际应用中, 需要对文法提出一些限制条件。

但这些限制并不真正限制该文法描述的语言。

限制 1: 文法中不得含有“有害规则”

有害规则: 引起文法二义性的规则。

$$\text{如 } E \rightarrow E+E \quad E \rightarrow E*E \quad E \rightarrow i$$

限制 2: 文法中不得含有“多余规则”

多余规则: 文法中任何句子的推导都用不到的规则。

(1) 规则中有不可到达的非终结符

(2) 规则中有不可终止的非终结符

限制 3: 不含“左递归”规则

$$\text{如: } A \rightarrow A\dots$$

限制 4: 有些文献规定: 不含“空规则”

$$\text{如: } A \rightarrow \varepsilon$$

但有些文献允许“空规则”的出现。

【举例】化简文法

G[S] : S→Be
 B→Ce
 B→Af
 A→Ae
 A→e
 C→Cf
 D→f

化简后

G[S] : S→Be
 B→Af
 A→Ae
 A→e

【本章小结】

1. 符号和符号串相关概念

(1) 字母表

(2) 符号串

长度、头、尾、固有头、固有尾、连接、幂
集合、集合的乘积、集合的闭包、正闭包

2. 文法和语言

(1) 规则、文法

(2) 推导、句子、句型

(3) 语言：一切句子的集合

(4) 文法对应的语言 语言对应的文法

(5) 文法的等价性

3. 文法的类型

0 型——短语文法

1 型——上下文有关文法

2 型——上下文无关文法

3 型——正规文法、正则文法（左线性、右线性）

4. 上下文无关文法及其语法树

(1) 语法树的构造

(2) 推导过程：最左推导、最右推导

(3) 文法二义性：一个句子对应两个或以上语法树

5. 句型的分析

(1) 自上而下

(2) 自下而上

(3) 短语、直接短语、句柄

【课堂练习】

1. “符号就是字符”，这种说法正确吗？ (B)

A. 正确 B. 不正确

2. 文法 $G[S]$: $S \rightarrow A0$

$S \rightarrow B1$

$A \rightarrow S1$

$A \rightarrow 1$

$B \rightarrow S0$

$B \rightarrow 0$

该文法是 Chomsky 的 3 型文法，该文法所描述的所有只含四个符号的句子是 1010, 0110, 1001, 0101。

3. 文法 $G[S]$: $S \rightarrow A$

$A \rightarrow B \mid \text{if } A \text{ then } A \text{ else } A$

$B \rightarrow C \mid B+C \mid +C$

$C \rightarrow D \mid C*D \mid *D$

$D \rightarrow x \mid (A) \mid -D$

哪些是终结符，哪些是非终结符？

答：终结符：if、then、else、+、*、x、(、)、-

非终结符：S、A、B、C、D

4. 文法 $G[S]$: $S \rightarrow a \mid aB$

$B \rightarrow aS$

该文法所描述的语言是 (C)

A. $L(G)=\{a^i \mid i \geq 0\}$ B. $L(G)=\{a^{2i} \mid i \geq 0\}$ C. $L(G)=\{a^{2i+1} \mid i \geq 0\}$ D. $L(G)=\{a^{2i+1} \mid i \geq 1\}$

5. 一个语言的文法是 (B)。

A. 唯一的 B. 不唯一的

6. 已知语言 $L=\{a^n b b^n \mid n \geq 1\}$ ，写出对应文法。

答案： $S \rightarrow aaAb$

$A \rightarrow aaAb \mid b$

7. 文法 $G[E]$: $E \rightarrow E+T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid a$

写出该文法句型 $E+F*(E+T)$ 的短语、直接短语和句柄。

答：短语： $E+F*(E+T)$ $F*(E+T)$ F $(E+T)$ $E+T$

直接短语： F $E+T$

句柄： F

8. (B) 正则文法能产生下面的语言: $L=\{a^n b^n \mid n \geq 1\}$

A. 存在一个 B. 不存在 C. 无法判别

9. 文法 $G[S]$: $S \rightarrow a|(T)|\epsilon$

$T \rightarrow T,S|S$

请给出句子 $(a,(a,a))$ 的最左、最右推导。

最左推导	最右推导
$S \Rightarrow (T)$	$S \Rightarrow (T)$
$\Rightarrow (T,S)$	$\Rightarrow (T,S)$
$\Rightarrow (S,S)$	$\Rightarrow (T,(T))$
$\Rightarrow (a,S)$	$\Rightarrow (T,(T,S))$
$\Rightarrow (a,(T))$	$\Rightarrow (T,(T,a))$
$\Rightarrow (a,(T,S))$	$\Rightarrow (T,(S,a))$
$\Rightarrow (a,(S,S))$	$\Rightarrow (T,(a,a))$
$\Rightarrow (a,(a,S))$	$\Rightarrow (S,(a,a))$
$\Rightarrow (a,(a,a))$	$\Rightarrow (a,(a,a))$

【课后作业】8

8. 上下文无关文法

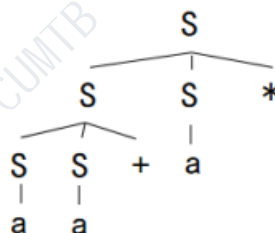
$S \rightarrow SS^* | SS+ | a$

(1) 通过此文法如何生成串 $aa+a^*$ ，并为该串构造推导树。

(2) 该文法生成的语言是什么？

答：

(1) $S \Rightarrow SS^*$
 $\Rightarrow Sa^*$
 $\Rightarrow SS+a^*$
 $\Rightarrow Sa+a^*$
 $\Rightarrow aa+a^*$



(2) 语言是

a ，或者以 aa 开头且以 a^* 或 $a+$ 结尾的 a 、 $*$ 、 $+$ 组成的串。