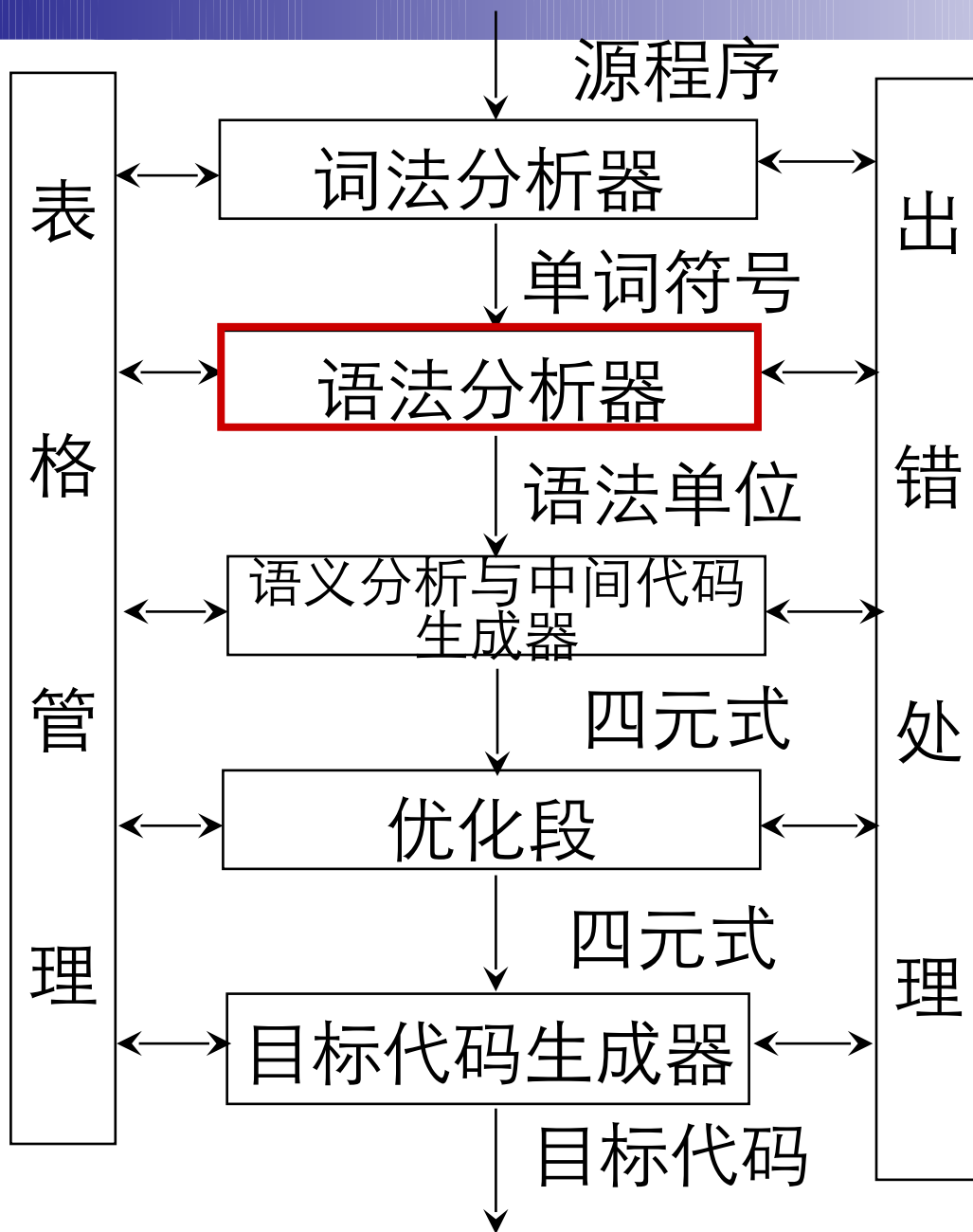




编译原理

第五章 语法分析——自下而上分析

编译程序总框



第五章 语法分析——自下而上分析

- 自下而上分析的基本问题
- 算符优先分析算法
- LR 分析法

第五章 语法分析——自下而上分析

- 自下而上分析的基本问题
- 算符优先分析算法
- LR 分析法

语法分析的方法

- 自上而下分析法 (Top-down)
- 自下而上分析法 (Bottom-up)

语法分析的方法

■ 自上而下分析法 (Top-down)

□ 基本思想

- 它从文法的开始符号出发，反复使用各种产生式，寻找"匹配"的**推导**

□ **递归下降分析法**

- 对每一语法变量（非终结符）构造一个相应的子程序，每个子程序识别一定的语法单位
- 通过子程序间的相互调用实现对输入串的认识

□ **预测分析程序**

- 非递归实现
- 直观、简单

语法分析的方法

■ 自下而上分析法 (Bottom-up)

□ 基本思想

- 从输入串开始，逐步进行**归约**，直到文法的开始符号
- **归约**：根据文法的产生式规则，把产生式的右部替换成左部符号
- 从树末端开始，构造语法树

□ **算符优先分析法**

- 按照算符的优先关系和结合性质进行语法分析
- 适合分析表达式

□ **LR 分析法**

- 规范归约

$G(E) : E \rightarrow i \mid E + E \mid E - E \mid E * E \mid E / E \mid (E)$

$i * i + i$

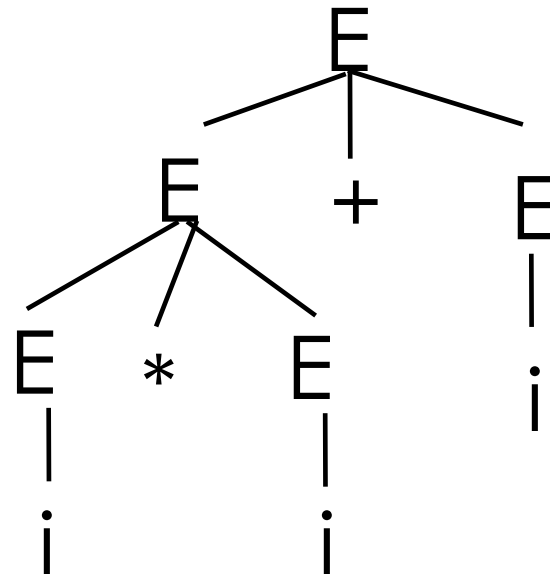
$E * i + i$

$E * E + i$

$E + i$

$E + E$

E

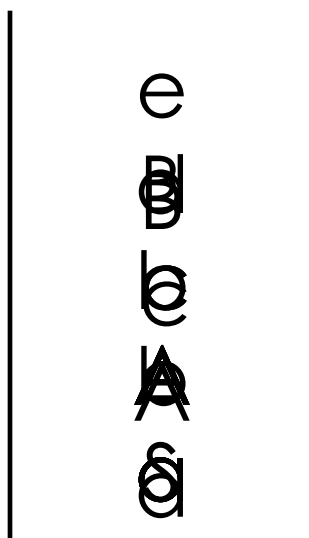


自下而上分析的基本问题

- 采用“**移进 - 归约**”思想进行自下而上分析
- 基本思想
 - 用一个寄存符号的先进后出栈，把输入符号一个一个地移进到栈里，当栈顶形成某个产生式的候选式时，即把栈顶的这一部分替换成（**归约**为）该产生式的左部符号。

- 例：设文法 $G(S)$:
 - (1) $S \rightarrow aAcBe$
 - (2) $A \rightarrow b$
 - (3) $A \rightarrow Ab$
 - (4) $B \rightarrow d$

试对 $abbcde$ 进行“移进－归约”分析。

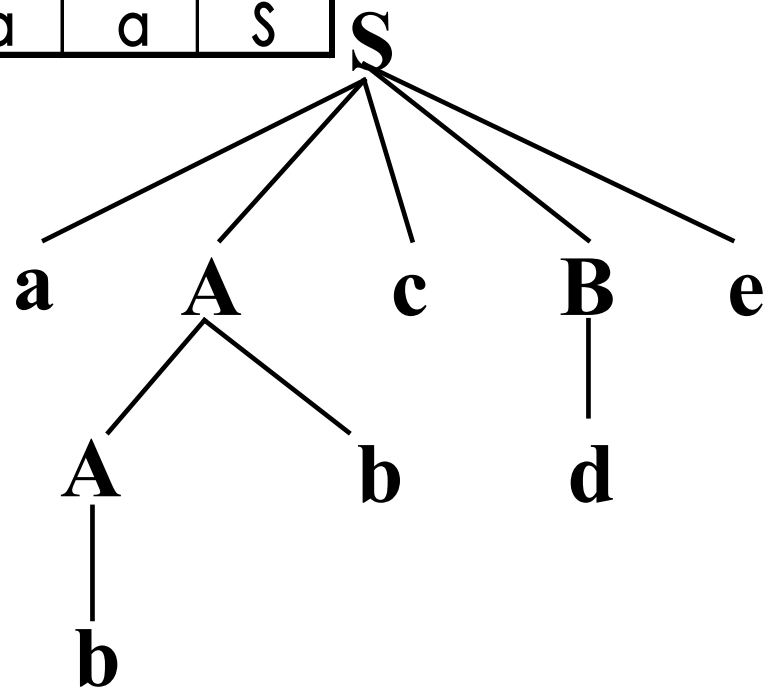


~~abbcde~~

步骤: 1 2 3 4 5 6 7 8 9 10
 动作: 进a 进b 归(2) 进b 归(3) 进c 进d 归(4) 进e 归(1)

								e	
						d	B	B	
			b		c	c	c	c	
	b	A	A	A	A	A	A	A	
a	a	a	a	a	a	a	a	a	S

- 分析树和语法树不一定一致
- 自下而上分析过程：边输入单词符号，边归约
- 核心问题：识别可归约串



思考：可归约串

■ 你认为什么是可归约串？

A. 连续出现的单词序列

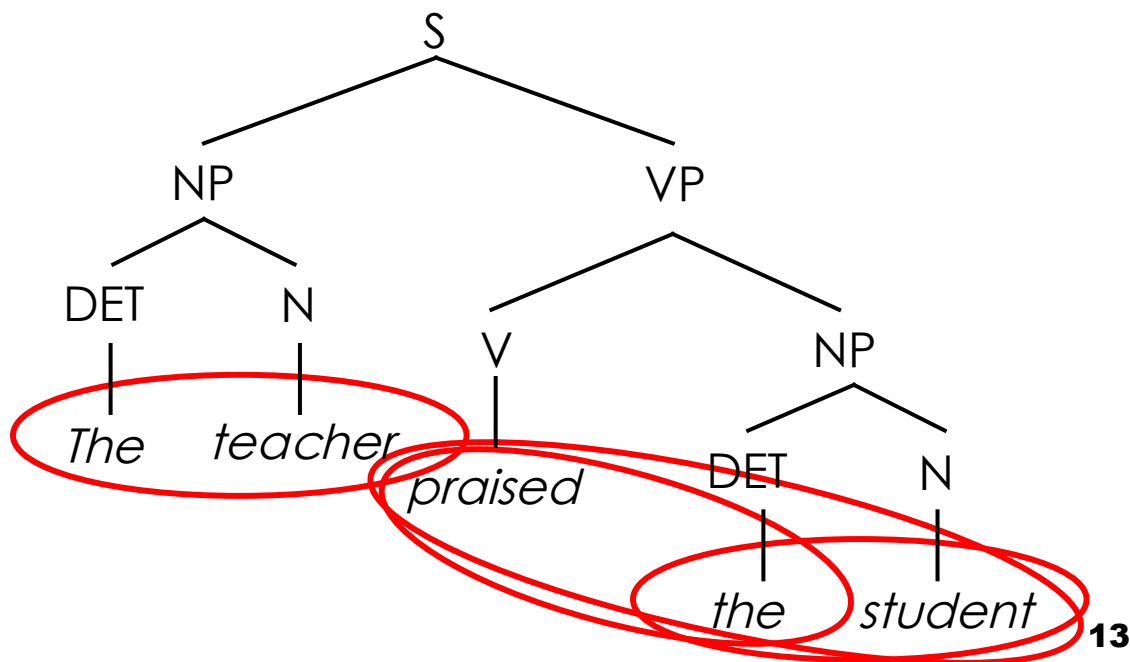
B. 短语

短语

- 定义：令 G 是一个文法， S 是文法的开始符号，假定 $\alpha\beta\delta$ 是文法 G 的一个句型，如果有

$$S \Rightarrow \alpha A \delta \text{ 且 } A \Rightarrow^+ \beta$$

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



短语、直接短语和句柄

- 定义：令 G 是一个文法， S 是文法的开始符号，假定 $\alpha\beta\delta$ 是文法 G 的一个句型，如果有
$$S \xRightarrow{*} \alpha A \delta \quad A \xRightarrow{+} \beta$$

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

特别是，如果有 $A \Rightarrow \beta$ ，则称 β 是句型 $\alpha\beta\delta$ 相对于规则 $A \rightarrow \beta$ 的直接短语。一个句型的最左直接短语称为该句型的句柄。

示例：短语、直接短语和句柄

考虑文法 $G(E)$:

$$E \rightarrow T \mid E+T$$

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

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

和句型 $i_1*i_2+i_3$:

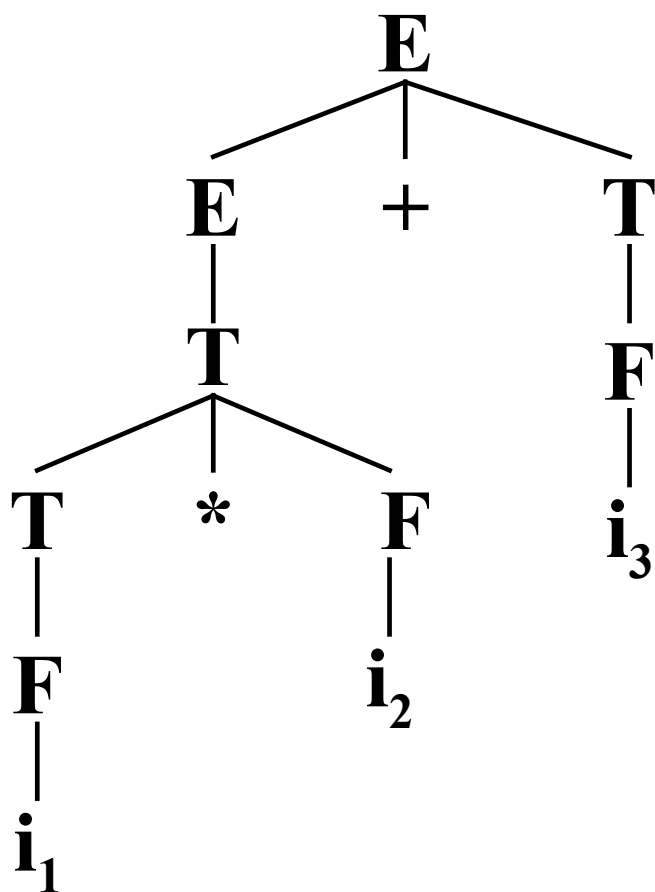
■ 短语: $i_1, i_2, i_3, i_1*i_2, i_1*i_2+i_3$

■ 直接短语: i_1, i_2, i_3

■ 句柄: i_1

$$\begin{aligned} E &\Rightarrow E+T \\ &\Rightarrow E+F \\ &\Rightarrow E+i_3 \\ &\Rightarrow T+i_3 \\ &\Rightarrow T^*F+i_3 \\ &\Rightarrow T^*i_2+i_3 \\ &\Rightarrow F^*i_2+i_3 \\ &\Rightarrow i_1*i_2+i_3 \end{aligned}$$

短语、直接短语和句柄



■ 在一个句型对应的语法树中

□ 以某非终结符为根的两代以上的子树的所有末端结点从左到右排列就是相对于该非终结符的一个短语

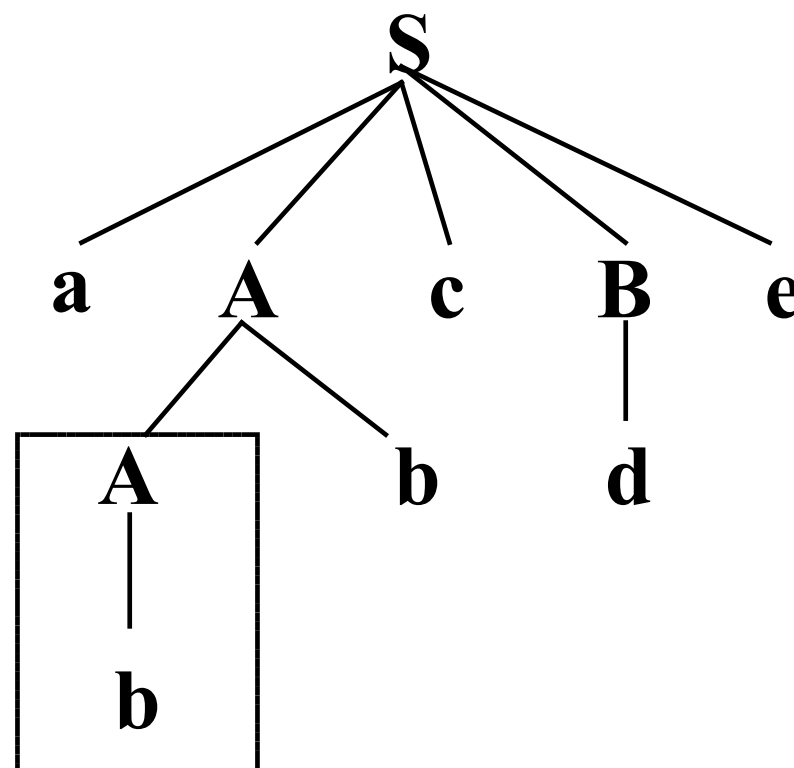
□ 如果子树只有两代，则该短语就是直接短语

短语、直接短语和句柄

- 可用句柄来对句子进行归约

句型 归约规则

abbcde (2) $A \rightarrow b$



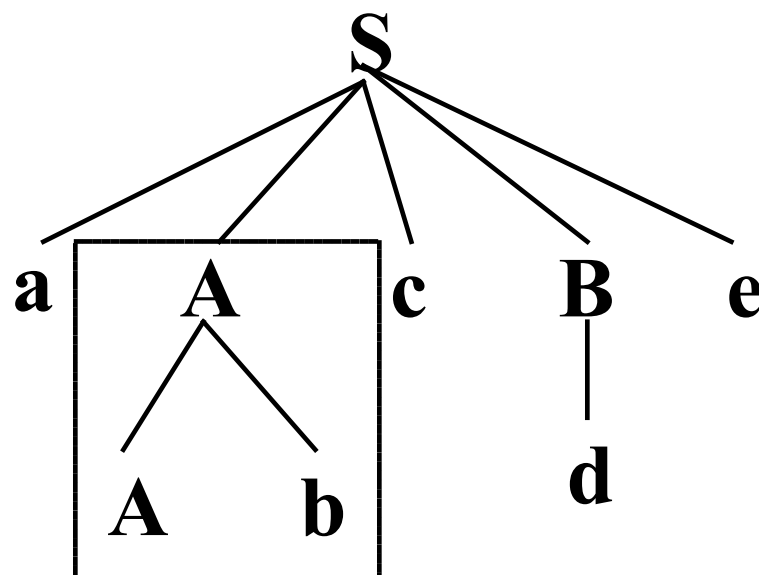
短语、直接短语和句柄

- 可用句柄来对句子进行归约

句型 归约规则

$a\underline{b}bcde$ (2) $A \rightarrow b$

$a\underline{A}bcde$ (3) $A \rightarrow Ab$



短语、直接短语和句柄

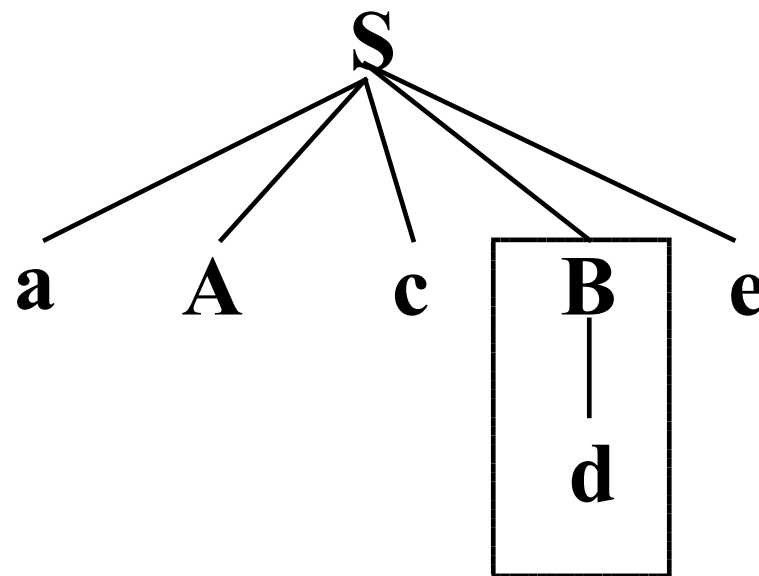
- 可用句柄来对句子进行归约

句型 归约规则

$a\underline{b}bcde$ (2) $A \rightarrow b$

$a\underline{A}bcde$ (3) $A \rightarrow Ab$

$aAc\underline{d}e$ (4) $B \rightarrow d$



短语、直接短语和句柄

- 可用句柄来对句子进行归约

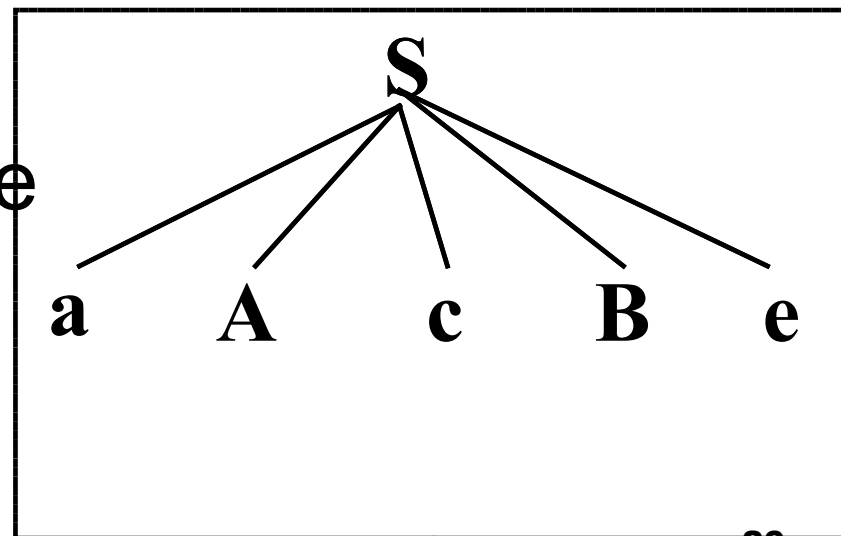
句型 归约规则

abbcde (2) $A \rightarrow b$

aAbcde (3) $A \rightarrow Ab$

aAcde (4) $B \rightarrow d$

aAcBe (1) $S \rightarrow aAcBe$



短语、直接短语和句柄

- 可用句柄来对句子进行归约

句型 归约规则

abbcde (2) $A \rightarrow b$

aAbcde (3) $A \rightarrow Ab$

aAcde (4) $B \rightarrow d$

aAcBe (1) $S \rightarrow aAcBe$

S

规范归约

- 定义：假定 α 是文法 G 的一个句子，我们称序列

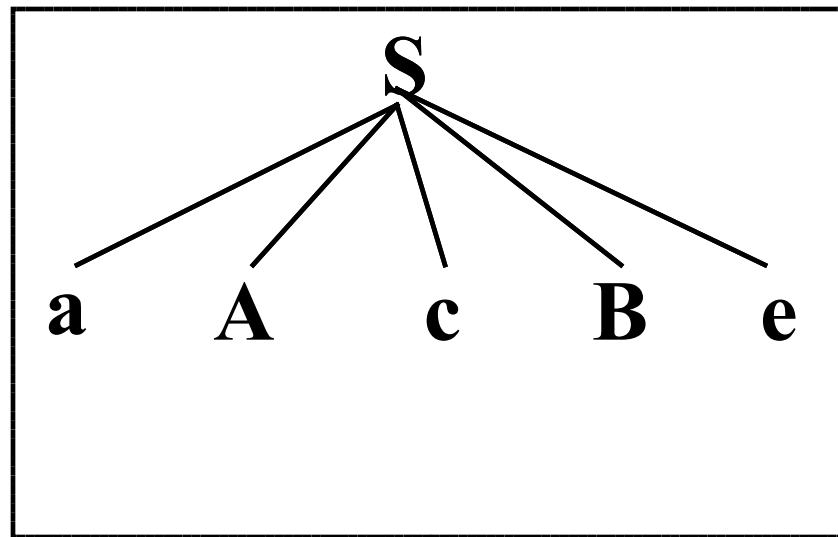
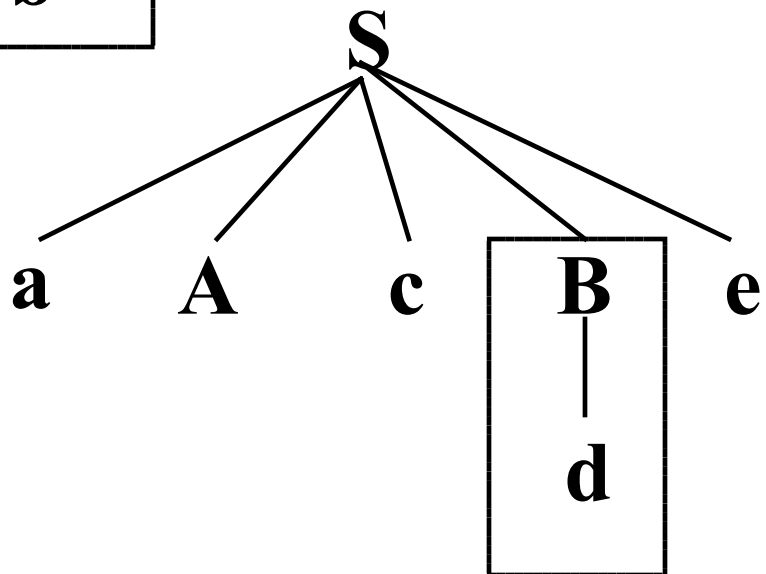
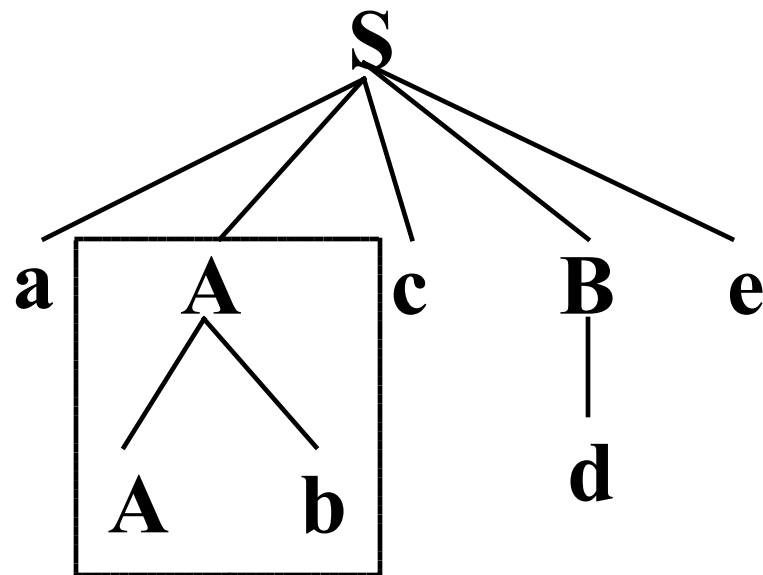
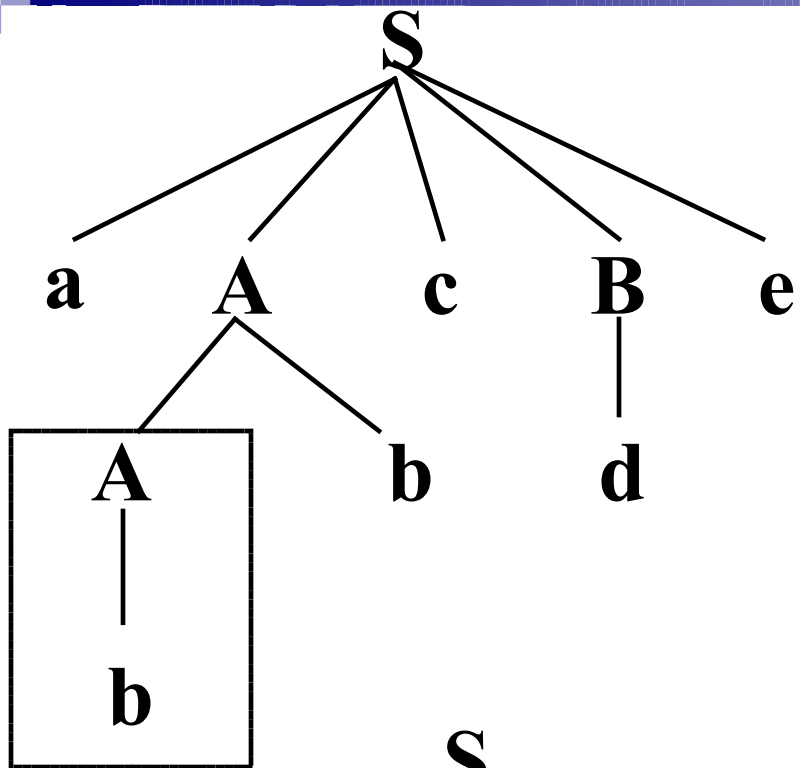
$$\alpha_n, \alpha_{n-1}, \dots, \alpha_0$$

是 α 的一个规范归约，如果此序列满足：

1. $\alpha_n = \alpha$
2. α_0 为文法的开始符号，即 $\alpha_0 = S$
3. 对任何 i ， $0 \leq i \leq n$ ， α_{i-1} 是从 α_i 经把句柄替换成为相应产生式左部符号而得到的

规范归约与规范推导

- 把上例倒过来写，则得到 $S \Rightarrow aAcBe \Rightarrow aAcde \Rightarrow aAbcde \Rightarrow abbcde$
- 显然这是一个最右推导。
- 规范归约是最右推导的逆过程
- 最左归约 规范推导
- 由规范推导推出的句型称为规范句型



符号栈的使用

- 栈是语法分析的一种基本数据结构。 ' # ' 作为栈底符号
- 考虑文法 $G(E)$:

$$E \rightarrow T \mid E+T$$

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

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

输入串为 $i_1 * i_2 + i_3$, 分析过程如何?

■ $G(E)$:

$E \rightarrow T \mid E+T$

$T \rightarrow F \mid T * F$

$F \rightarrow (E) \mid i$

步骤	符号栈	输入串	动作
0	#	$i_1 * i_2 + i_3 \#$	预备
1	$\#i_1 * i_2 + i_3 \#$		进
2	$\#F$	$*i_2 + i_3 \#$	归, 用 $F \rightarrow i$
3	$\#T$	$*i_2 + i_3 \#$	归, 用 $T \rightarrow F$
4	$\#T^*$	$i_2 + i_3 \#$	进

■ $G(E)$:

$E \rightarrow T \mid E+T$

$T \rightarrow F \mid T * F$

$F \rightarrow (E) \mid i$

步骤	符号栈	输入串	动作
4	#T*	$i_2+i_3\#$	进
5	#T* i_2	$+i_3\#$	进
6	#T*F	$+i_3\#$	归, 用 $F \rightarrow i$
7	#T	$+i_3\#$	归, 用 $T \rightarrow T * F$
8	#E	$+i_3\#$	归, 用 $E \rightarrow T$
9	#E+	$i_3\#$	进

■ $G(E)$:

$E \rightarrow T \mid E+T$

$T \rightarrow F \mid T * F$

$F \rightarrow (E) \mid i$

<u>步骤</u>	<u>符号栈</u>	<u>输入串</u>	<u>动作</u>
9	#E+	$i_3\#$	进
10	#E+ i_3	#	进
11	#E+F	#	归, 用 $F \rightarrow i$
12	#E+T	#	归, 用 $T \rightarrow F$
13	#E	#	归, 用 $E \rightarrow E+T$
14	#E	#	接受

小结

- 归约、移进 - 归约分析
- 核心问题：识别可归约串
- 短语、直接短语和句柄
- 规范归约
- 符号栈的使用

作业

- P133—1 , 2