

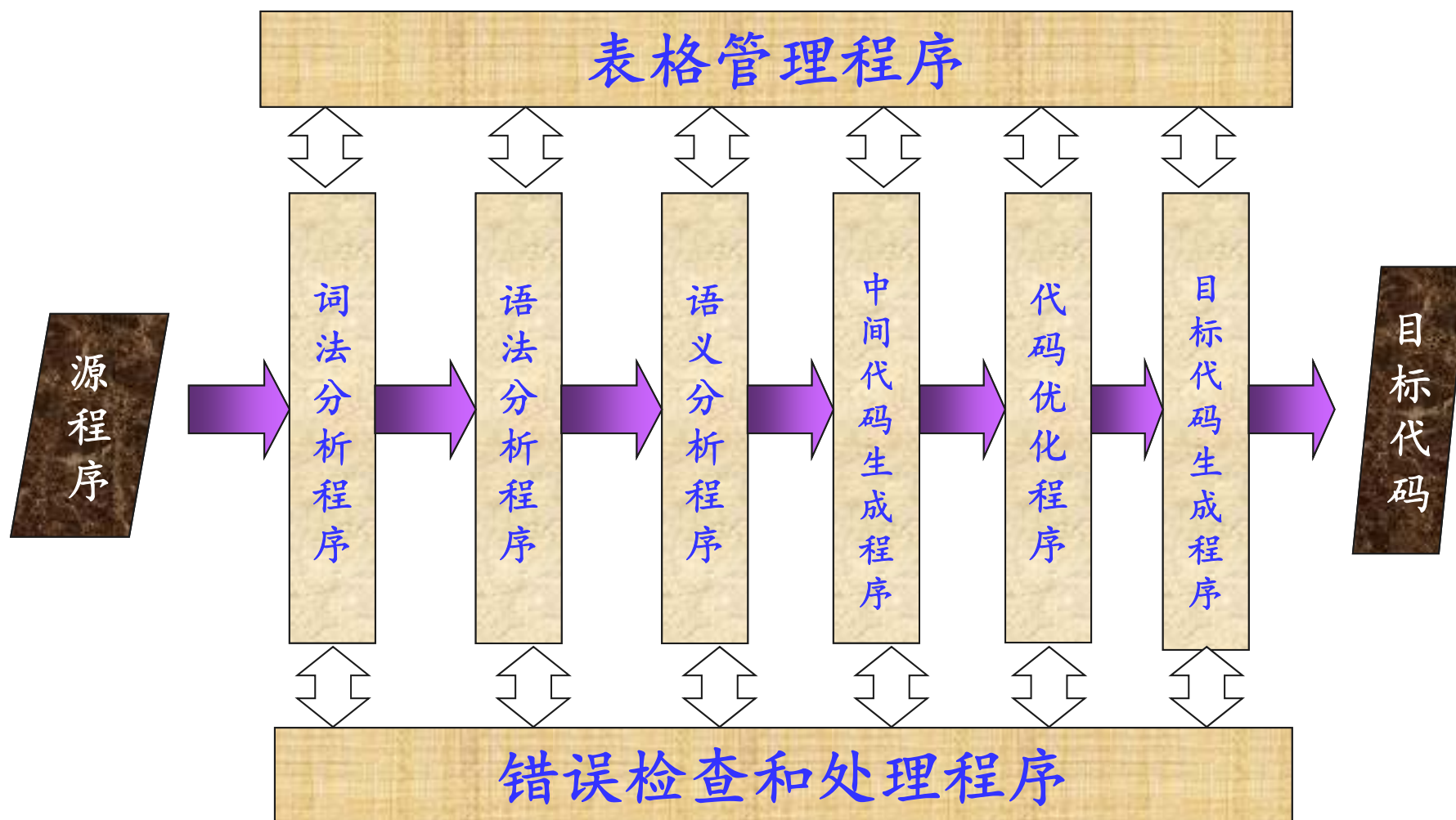


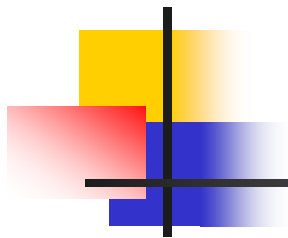
# 编译原理

---

武汉大学计算机学院  
编译原理课程组

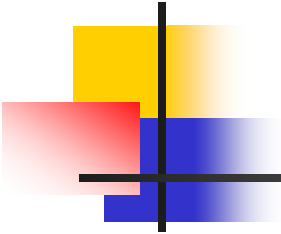
# 编译程序的结构





# 前述内容回顾 —— 自上而下语法分析

- 基本思想
- 存在的问题
- 解决方法      消除左递归      FIRST集、FOLLOW集
- LL(1) 文法
- LL(1) 方法      LL(1) 分析表的构造
- 递归子程序法      递归子程序的构造



## 第6章 自下而上语法分析

---

- 基本思想
- 存在的问题及解决方法
- 短语和句柄
- 简单优先分析方法
- 算符优先分析方法



## 6.1 自下而上语法分析

### 1. 自下而上语法分析的基本思想

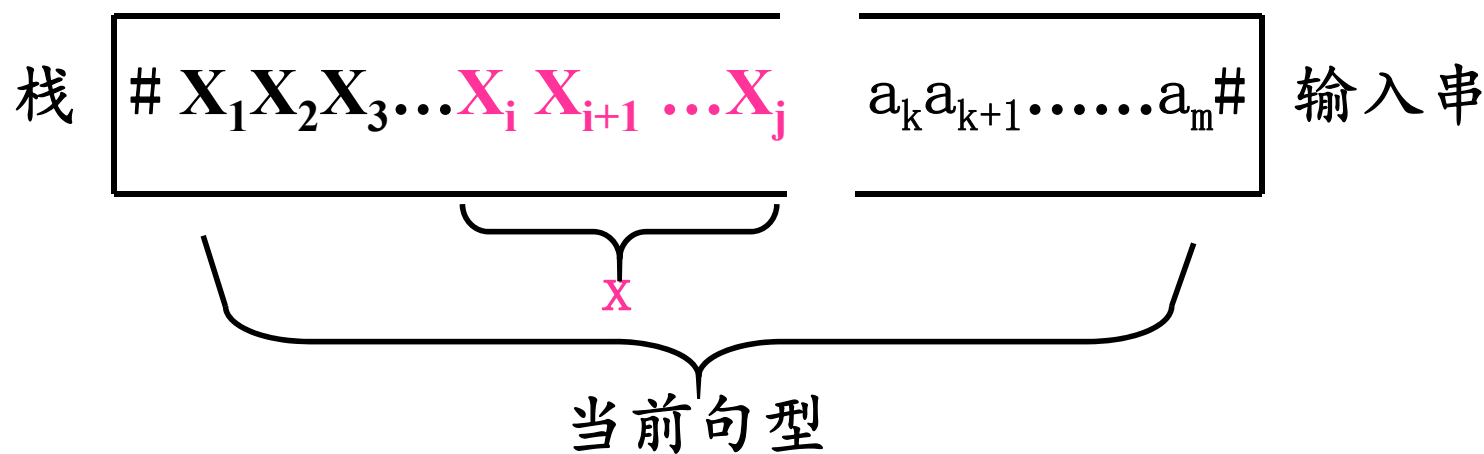
从输入的符号串开始，利用文法的产生式步步向上归约，试图归约到文法的识别符号。如果从语法树的角度看，自下而上分析的过程是以输入符号串作为末端结点符号串，向着根结点的方向往上构造语法树，使识别符号正好是该语法树的根结点。

相应于高级语言的编译过程，自下而上语法分析就是从该高级语言文法的源程序或与其等价的单词串出发，试图归约得到该文法的开始符号——<程序>。

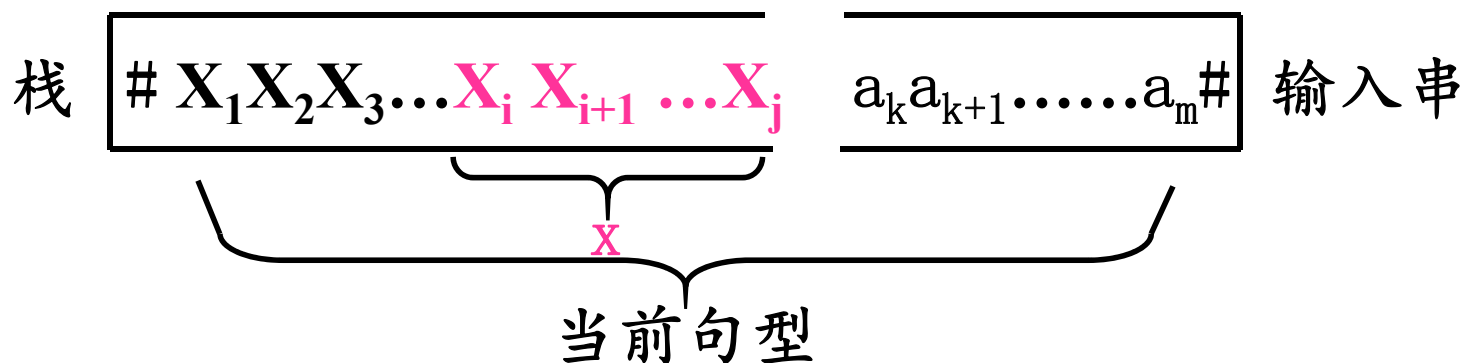
# 6.1 自下而上语法分析

## 2. 基本实现技术 —— 移进-归约法

引进一个后进先出的栈来存放符号，按照扫描顺序把当前输入符号下推入栈中(移进)，然后查看栈顶的符号串是否可以被归约。



## 6.1 自下而上语法分析

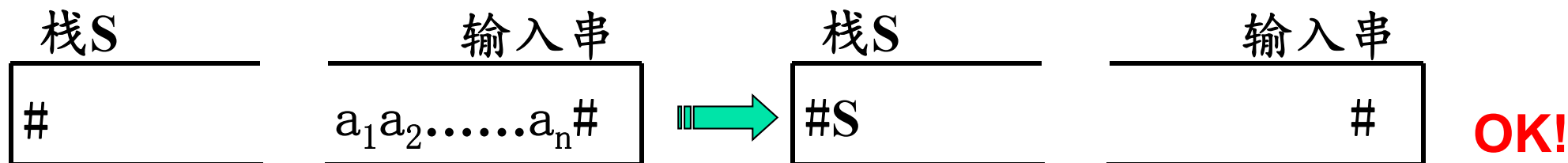


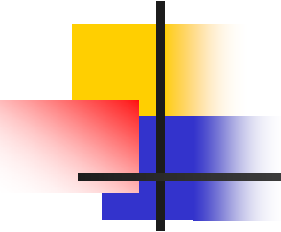
**移进**——读入下一个输入符号并把它下推进栈。

**归约**——把栈顶符号串替换为相应的非终结符号。

**接受**——当栈顶除 $\#$ 外仅有识别符号，输入也到 $\#$ 时，结束工作。

**报错**——当识别出**错误**时，调用出错处理子程序处理或停止。





## 6.1 自下而上语法分析

### 3. 自下而上语法分析遇到的基本问题

例如，对文法G[S]:

$$S \rightarrow aAcBe$$
$$A \rightarrow b$$
$$A \rightarrow Ab$$
$$B \rightarrow d$$

分析符号串abbcde。

遇到的两个问题:

问题1) 选择哪个子串作为句柄?

问题2) 选择哪个产生式进行归约?



## 6.2 短语和句柄

句柄的条件?

### 1. 短语、直接短语与句柄

设有文法 $G[S]$ ,  $w=xuy$ 是它的一个句型, 如果有

$$S \xRightarrow{*} xUy \quad \text{且} \quad U \xRightarrow{+} u$$

则称句型 $xuy$ 中的子串 $u$ 为句型 $xuy$ (相对于非终结符号 $U$ )的短语。

特别地, 如果前述条件中 $U \xRightarrow{+} u$ 为

$$U \Rightarrow u$$

则称 $u$ 为句型 $xuy$ 的直接(简单)短语。

句型的最左直接(简单)短语称为句柄。



## 6.2 短语和句柄

---

例：设有文法 $G[S]$ ：

$$S \rightarrow AB$$
$$A \rightarrow Aa \mid bB$$
$$B \rightarrow a \mid Sb$$

请给出句型baabaab的所有短语、直接短语和句柄。



## 6.2 短语和句柄

---

### 2. 短语、简单（直接）短语、句柄与语法树的关系

语法树子树的末端结点符号串即是语法树所描述句型（相对于子树的根）的短语；

简单子树（只有父子两代）的末端结点符号串即是直接短语。

最左简单子树的末端结点符号串即是句柄。



## 6.2 短语和句柄

---

例：设有文法 $G[S]$ ：

$$S \rightarrow AB$$

$$A \rightarrow Aa \mid bB$$

$$B \rightarrow a \mid Sb$$

请给出句型baabaab的所有短语、直接短语和句柄。



## 第 6 章 自学内容

### 6.4节—6.9节 自学

优先分析方法的主要思想:

定义文法符号被归约的优先级

句柄(最左素短语): 满足下列关系的最左子串

$$a_{i-1} \leftarrow a_i, a_i \pm a_{i+1}, \dots, a_{j-1} \pm a_j, a_j \rightarrow a_{j+1}$$

$$a_{i-1} \left\langle \bullet a_i, a_i \underline{\quad \bullet \quad} a_{i+1}, \dots, a_{j-1} \underline{\quad \bullet \quad} a_j, a_j \bullet \right\rangle a_{j+1}$$

优先分析方法的使用条件: 优先文法

- 1) 任意两符号间至多存在一种优先关系;
- 2) 任意两条产生式无相同右部。

## 第 6 章 自学内容

简单优先分析方法举例：文法 $G_{34}[S]$ ： $S \rightarrow (R) \mid a \mid ^$

$R \rightarrow T$

$T \rightarrow S, T \mid S$

其简单优先关系矩阵如下，试给出符号串 $((a),a)$ 的分析过程。

	R	S	T	a	^	,	(	)
R								±
S						±		→
T								→
a						→		→
^						→		→
,		←	±	←	←		←	
(	±	←	←	←	←		←	
)						→		→



# 简单优先分析法

---

局限性:

例: 判断下列文法是否为简单优先文法:

文法G[E]:

$$E \rightarrow E + T \mid T$$
$$T \rightarrow T * F \mid F$$
$$F \rightarrow (E) \mid i$$



# 算符优先分析法

## 前提

算符文法:

如果一个文法G中没有形如 $U \rightarrow \dots AB \dots$ 的规则( $A, B \in V_N$ ), 则称文法G为算符文法(OG)。

文法G[E]:

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

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

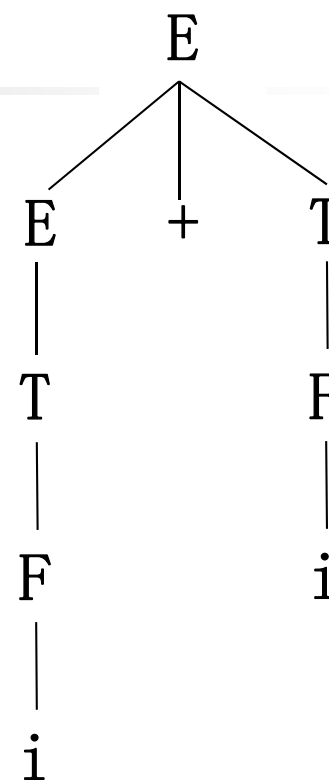
$$F \rightarrow (E) \mid i$$



# 算符优先分析法

## 算符优先文法

文法G[E]:

$$E \rightarrow E + T \mid T$$
$$T \rightarrow T * F \mid F$$
$$F \rightarrow (E) \mid i$$


句型的素短语是这样一种短语，它至少包含一个终结符号，并除自身外不再包含其它更小的素短语。

最左素短语是指句型中最左边的素短语。

# 算符优先分析法

## 算符优先文法

如果某算符文法的终结符号集 $V_T$ 中任意两个符号之间至多有一种算符优先关系成立，则称此算符文法为算符优先文法(OPG)。

一个算符优先文法的每个句型  $\#[N_1]a_1[N_2]a_2\ldots[N_n]a_n[N_{n+1}]#$  的最左素短语是满足下列条件：

$$a_{i-1} < \cdot a_i, a_i \underline{\quad \cdot \quad} a_{i+1}, \ldots, a_{j-1} \underline{\quad \cdot \quad} a_j, a_j > \cdot a_{j+1}$$

的最左符号子串  $N_i a_i \ldots N_j a_j N_{j+1}$ 。

# 算符优先分析方法

算符优先分析方法举例：文法 $G_{35}[E]$ ：

$E \rightarrow E+T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid i$

其算符优先关系矩阵如下，试给出符号串 $(i+i)*i$ 的分析过程。

	+	*	i	(	)
+	$\triangleright$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleright$
*	$\triangleright$	$\triangleright$	$\triangleleft$	$\triangleleft$	$\triangleright$
i	$\triangleright$	$\triangleright$			$\triangleright$
(	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\equiv$
)	$\triangleright$	$\triangleright$			$\triangleright$



# 算符优先分析方法

在通常的程序语言中，一些语法范畴的定义不满足**OG文法**的条件。

Pascal 中：

$\langle \text{循环语句} \rangle \rightarrow \langle \text{循环子句} \rangle \langle \text{子句} \rangle$

$\langle \text{循环子句} \rangle \rightarrow \text{for} \langle \text{变量} \rangle := \langle \text{表达式} \rangle \text{ to } \langle \text{表达式} \rangle \text{ do}$

可修改文法如下：

$\langle \text{循环语句} \rangle \rightarrow \text{for} \langle \text{变量} \rangle := \langle \text{循环表} \rangle \text{ do } \langle \text{语句} \rangle$

$\langle \text{循环表} \rangle \rightarrow \langle \text{表达式} \rangle \text{ to } \langle \text{表达式} \rangle$



## 第6章 内容小结

---

- 基本思想
- 存在的问题及解决方法
- 短语和句柄
- 简单优先分析方法
- 算符优先分析方法



## 第 6 章 作业

---

求下列句型的所有短语、简单（直接）短语和句柄。



# 下章内容简介 —— 第7章

---

LR分析法