

第4章 语法分析

- 1 语法分析程序的功能
- 2 自上而下分析法
- 3 自下而上分析法的一般原理
- 4 算符优先分析方
- 5 LR分析法

第4章 语法分析

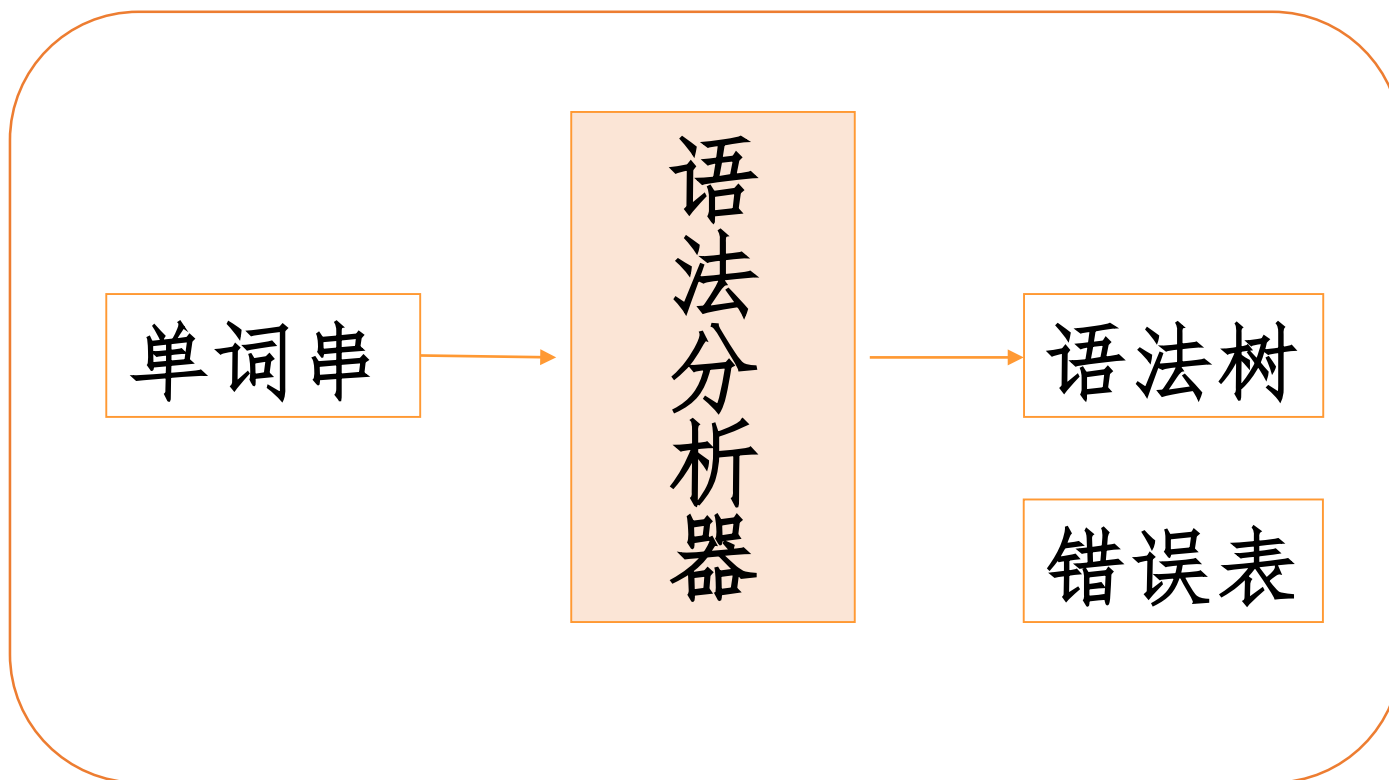
◆ 语法分析方法

◆ 自顶向下语法分析法

◆ 算符优先分析法

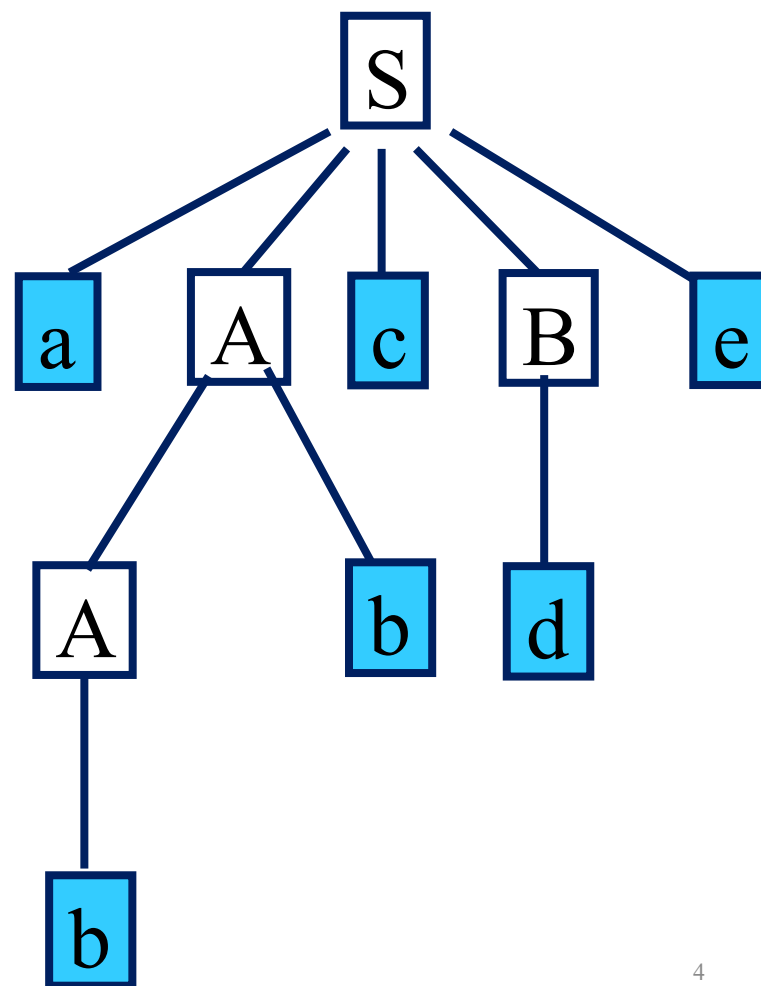
◆ LR分析法

4.1 语法分析程序的功能



4.1 语法分析程序的功能

语法分析 { 自顶向下
 自底向上

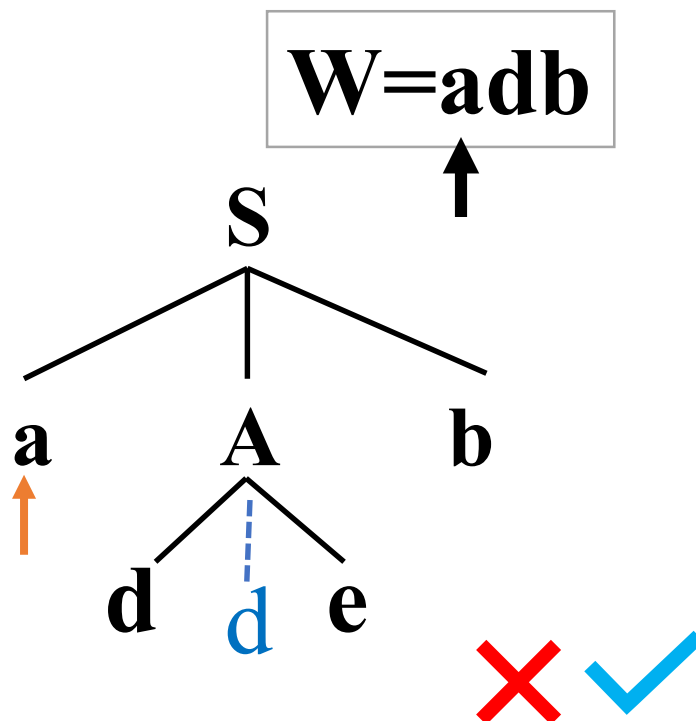


4.2.1 非确定的自上而下分析法的思想

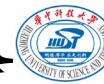


例1 设有文法 $G[S]$:

$S \rightarrow aAb$
 $A \rightarrow de \mid d$



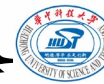
4.2.2 文法的左递归和回溯的消除



➤ 文法左递归

文法左递归：文法中的某个非终结符
A存在推导 $A \xRightarrow{+} A\alpha$

4.2.2 文法的左递归和回溯的消除



(1) 右递归代替左递归

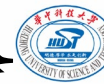
$$A \rightarrow A\alpha \mid \beta \quad \left\{ \begin{array}{l} A \rightarrow \beta A' \\ A' \rightarrow \alpha A' \mid \varepsilon \end{array} \right.$$

$$A \rightarrow A\alpha_1 \mid A\alpha_2 \mid \dots \mid A\alpha_m \mid \beta_1 \mid \beta_2 \mid \dots \mid \beta_n$$

$$\left\{ \begin{array}{l} A \rightarrow \beta_1 A' \mid \beta_2 A' \mid \dots \mid \beta_n A' \end{array} \right.$$

$$\left\{ \begin{array}{l} A' \rightarrow \alpha_1 A' \mid \alpha_2 A' \mid \dots \mid \alpha_m A' \mid \varepsilon \end{array} \right.$$

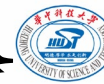
4.2.2 文法的左递归和回溯的消除



例2 设有文法G[E]:
请消除文法左递归

$$\begin{aligned} E &\rightarrow E+T \mid E-T \mid T \\ T &\rightarrow T*F \mid T/F \mid F \\ F &\rightarrow (E) \mid \text{id} \end{aligned}$$
$$E \rightarrow TE'$$
$$E' \rightarrow +TE' \mid -TE' \mid \varepsilon$$
$$T \rightarrow FT'$$
$$T' \rightarrow *FT' \mid /FT' \mid \varepsilon$$
$$F \rightarrow (E) \mid \text{id}$$

4.2.2 文法的左递归和回溯的消除



(2) 扩充BNF表示递归

$\{\alpha\}$: 符号串 α 出现0次或多次, 即 α^*

$[\alpha]$: 表示 α 可有可无

(α) : 表示公共因子

例2

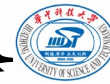
$\langle \text{标识符} \rangle \rightarrow \langle \text{标识符} \rangle l \mid \langle \text{标识符} \rangle d \mid$

$\langle \text{标识符} \rangle \rightarrow l\{l\mid d\}$

$\langle \text{条件语句} \rangle \rightarrow \text{if} \langle \text{布尔表达式} \rangle \langle \text{语句} \rangle [\text{;else} \langle \text{语句} \rangle]$

$A \rightarrow x (\alpha_1 \mid \alpha_2 \mid \dots \mid \alpha_m)$

4.2.2 文法的左递归和回溯的消除



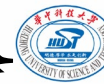
例3 请用扩充BNF表示法对左递归文法改写

$$\begin{aligned} E &\rightarrow E+T \mid E-T \mid T \\ T &\rightarrow T*F \mid T/F \mid F \\ F &\rightarrow (E) \mid \text{id} \end{aligned}$$

改写为：

$$\begin{aligned} E &\rightarrow T\{ +T \mid -T \} \\ T &\rightarrow F\{ *F \mid / F \} \\ F &\rightarrow (E) \mid \text{id} \end{aligned}$$

4.2.2 文法的左递归和回溯的消除

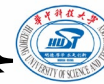


例4 用扩充BNF表示法消除左递归

$$A \rightarrow Ac \mid Aad \mid bd \mid e$$

改写为： $A \rightarrow (bd \mid e) \{ c \mid ad \}$

4.2.2 文法的左递归和回溯的消除



回溯产生的原因：

$$A \rightarrow \alpha_1 \mid \alpha_2 \mid \alpha_3 \mid \cdots \mid \alpha_n$$

$$S \rightarrow aAb$$

$$A \rightarrow de \mid d$$

(a)

$$A \rightarrow Bx$$

$$B \rightarrow x \mid \varepsilon$$

(b)

在自上而下分析过程中，为避免分析时的回溯，要求：语言的文法为LL(1)文法。

LL(1)文法的判断条件

LL(1)文法的判断，需要考虑三个集合：

First集

Follow集

Select集

LL(1)文法的判断条件

(1) 设 α 是文法 G 的任一符号串，定义文法符号串 α 的首符号集合 $\text{First}(\alpha)$ ：

$$\text{First}(\alpha) = \{ a \mid \alpha \xRightarrow{*} a... \text{ 且 } a \in V_T \}$$

规定：若 $\alpha \xRightarrow{*} \varepsilon$ ，则 $\varepsilon \in \text{First}(\alpha)$

LL(1)文法的判断条件

例5 设有文法 $G[S]$:

$$S \rightarrow Ap \mid Bq$$

计算 $\text{First}(Ap)$ 、 $\text{First}(Bq)$

$$A \rightarrow cA \mid a$$
$$B \rightarrow dB \mid b$$

$$\text{First}(Ap) = \{ c, a \}$$

$$Ap \Rightarrow cAp$$

$$Ap \Rightarrow ap$$

$$\text{First}(Bq) = \{ b, d \}$$

$$Bq \Rightarrow bq$$

$$Bq \Rightarrow dBq$$

LL(1)文法的判断条件

(2) 设文法G的开始符号为S，对于G的任何非终结符A，定义非终结符A的后继符号的集合

$$\text{Follow}(A) = \{ a \mid S \Rightarrow \dots A^* a \dots \text{且} a \in V_T \}$$

若有 $S \xRightarrow{*} \dots A$ ，则规定 $\$ \in \text{Follow}(A)$ 。

LL(1)文法的判断条件

例6 设有文法 $G[A]$:
计算 $\text{Follow}(B)$

$$A \rightarrow aB \mid d$$
$$B \rightarrow bBA \mid \varepsilon$$

$$\text{Follow}(B) = \{ \$, d, a \}$$

$$A \Rightarrow aB$$

$$A \Rightarrow aB \Rightarrow abBA \Rightarrow abBd$$

$$A \Rightarrow aB \Rightarrow abBA \Rightarrow abBaB$$

LL(1)文法的判断条件

(3) $A \rightarrow \alpha$ 是文法 G 的任一条规则, $A \in V_N$, $\alpha \in (V_N \cup V_T)^*$, $\text{Select}(A \rightarrow \alpha)$ 定义为:

$\text{Select}(A \rightarrow \alpha) =$

$\text{First}(\alpha)$

若 $\alpha \xRightarrow{*} \varepsilon$

$\text{First}(\alpha) \setminus \{\varepsilon\} \cup \text{Follow}(A)$

若 $\alpha \xRightarrow{*} \varepsilon$

LL(1)文法的判断条件

例7 设有文法 $G[A]: A \rightarrow aB \mid d$
 $B \rightarrow bBA \mid \varepsilon$

计算每条规则的Select集

$$\text{Select}(A \rightarrow aB) = \text{First}(aB) = \{ a \}$$

$$\text{Select}(A \rightarrow d) = \text{First}(d) = \{ d \}$$

$$\text{Select}(B \rightarrow bBA) = \text{First}(bBA) = \{ b \}$$

$$\text{Select}(B \rightarrow \varepsilon)$$

$$= \text{First}(\varepsilon) \setminus \{ \varepsilon \} \cup \text{Follow}(B) = \{ \$, d, a \}$$

$$\text{Follow}(B) = \{ \$, d, a \}$$

LL(1)文法的判断条件

上下文无关文法 **G** 是 **LL(1)** 文法，
当且仅当，**G** 中每个非终结符 **A** 的任何两个规则 **A** \rightarrow α 、**A** \rightarrow β ，都满足
Select(A \rightarrow α) \cap Select(A \rightarrow β) = Φ
(α 、 β 中，至多只有一个能推出 ϵ 串)

LL(1)文法的判断条件

例8 设有文法G[S]:

$$S \rightarrow aAb$$
$$A \rightarrow de \mid d$$

$\therefore \text{Select}(A \rightarrow de) = \text{First}(de) = \{d\}$

$\text{Select}(A \rightarrow d) = \text{First}(d) = \{d\}$

$\therefore \text{Select}(A \rightarrow de) \cap \text{Select}(A \rightarrow d) = \{d\} \neq \Phi$

\therefore 该文法不是LL(1)文法

LL(1)文法的判断条件

例9 设有文法G[A]是否LL (1) 文法?

$A \rightarrow aB \mid d$

$B \rightarrow bBA \mid \varepsilon$

$\text{Select}(A \rightarrow aB) = \text{First}(aB) = \{a\}$

$\text{Select}(A \rightarrow d) = \text{First}(d) = \{d\}$

$\text{Select}(B \rightarrow bBA) = \text{First}(bBA) = \{b\}$

$\text{Select}(B \rightarrow \varepsilon) = \text{First}(\varepsilon) \setminus \{\varepsilon\} \cup \text{Follow}(B)$
 $= \{a, d, \$\}$

$\text{Select}(A \rightarrow aB) \cap \text{Select}(A \rightarrow d) = \Phi$

$\text{Select}(B \rightarrow bBA) \cap \text{Select}(B \rightarrow \varepsilon) = \Phi$

G[A]是LL(1)文法

LL(1)文法的判断条件

例10 设有文法G[S]:

$$S \rightarrow aAB$$

$$A \rightarrow bB \mid dA \mid \varepsilon \quad B \rightarrow a \mid e$$

试判断该文法是否LL(1)文法。

$$\begin{aligned} \because \text{Select}(A \rightarrow bB) &= \text{First}(bB) = \{b\} \\ \text{Select}(A \rightarrow dA) &= \text{First}(dA) = \{d\} \\ \text{Select}(A \rightarrow \varepsilon) &= \text{First}(\varepsilon) \setminus \{\varepsilon\} \cup \text{Follow}(A) \\ &= \{a, e\} \end{aligned}$$

LL(1)文法的判断条件

$\therefore \text{Select}(A \rightarrow bB) = \text{First}(bB) = \{b\}$
 $\text{Select}(A \rightarrow dA) = \text{First}(dA) = \{d\}$
 $\text{Select}(A \rightarrow \varepsilon) = \text{First}(\varepsilon) \setminus \{\varepsilon\} \cup \text{Follow}(A) = \{a, e\}$

$S \rightarrow aAB$
 $A \rightarrow bB \mid dA \mid \varepsilon$
 $B \rightarrow a \mid e$

$\text{Select}(B \rightarrow a) = \text{First}(a) = \{a\}$
 $\text{Select}(B \rightarrow e) = \text{First}(e) = \{e\}$
 $\text{Select}(A \rightarrow bB) \cap \text{Select}(A \rightarrow dA) = \Phi$
 $\text{Select}(A \rightarrow bB) \cap \text{Select}(A \rightarrow \varepsilon) = \Phi$
 $\text{Select}(A \rightarrow dA) \cap \text{Select}(A \rightarrow \varepsilon) = \Phi$
 $\text{Select}(B \rightarrow a) \cap \text{Select}(B \rightarrow e) = \Phi$
 \therefore 该文法为LL(1)文法

LL(1)文法的判断条件

例11 设有文法G[S]:

$$\begin{aligned} S &\rightarrow AB \mid bC \\ A &\rightarrow b \mid \varepsilon \\ B &\rightarrow aD \mid \varepsilon \\ C &\rightarrow AD \mid \varepsilon \\ D &\rightarrow aS \mid c \end{aligned}$$

试判断该文法是否LL(1)文法

LL(1)文法的判断条件

Select($S \rightarrow AB$)

=First(AB)/ $\{\epsilon\} \cup \text{Follow}(S)$

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

First(AB)=First(A) \cup First(B)

= $\{a, b, \epsilon\}$

Follow(S)= $\{\$ \}$

Select($S \rightarrow bC$)=First(bC)= $\{b\}$

Select($S \rightarrow AB$) \cap Select($S \rightarrow bC$)

= $\{a, b, \$\} \cap \{b\} = \{b\} \neq \Phi$

$G[S]$ 不是LL(1)文法

$S \rightarrow AB \mid bC$

$A \rightarrow b \mid \epsilon$

$B \rightarrow aD \mid \epsilon$

$C \rightarrow AD \mid \epsilon$

$D \rightarrow aS \mid c$

LL(1)文法的判断条件

文法G[S]是LL(1)文法，对任何的输入串可进行确定的自上而下分析。

根据输入符号a属于的Select集，唯一地选择一条规则进行准确地匹配。

4.2.3 非LL(1)文法到LL(1)文法



➤ 存在公共左因子问题:

$$A \rightarrow \alpha\beta_1 \mid \alpha\beta_2 \mid \dots \mid \alpha\beta_n$$

$$\text{Select}(A \rightarrow \alpha\beta_i) \cap \text{Select}(A \rightarrow \alpha\beta_j) \neq \Phi$$

$$A \rightarrow \alpha A'$$

$$A' \rightarrow \beta_1 \mid \beta_2 \mid \dots \mid \beta_n$$

4.2.3 非LL(1)文法到LL(1)文法



例1 设有文法G[S]:
请改写为LL(1)文法

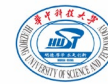
$$\begin{aligned} S &\rightarrow aAb \\ A &\rightarrow de \mid d \end{aligned}$$

改写为G' [S]:

$$S \rightarrow aAb$$
$$A \rightarrow dA'$$
$$A' \rightarrow e \mid \varepsilon$$

改写后的文法是LL(1)文法。

4.2.3 非LL(1)文法到LL(1)文法



例2 设有文法G[S]: $S \rightarrow ad \mid Ae$
改写为LL(1) $A \rightarrow aS \mid bA$

将A的规则代入非终结符S的规则中得:

$S \rightarrow ad \mid aSe \mid bAe$

$A \rightarrow aS \mid bA$

对S提取公共左因子得

$S \rightarrow bAe \mid aS'$

$S' \rightarrow d \mid Se$

$A \rightarrow aS \mid bA$

改写后的文法是LL(1)文法。

4.2.3 非LL(1)文法到LL(1)文法



提问：非LL(1)文法都能改写为LL(1)文法？

例3 文法G[S]:

$$\begin{aligned} S &\rightarrow Ae \mid Bd \\ A &\rightarrow aAe \mid b \\ B &\rightarrow aBd \mid b \end{aligned}$$

$\therefore \text{Select}(S \rightarrow Ae) \cap \text{Select}(S \rightarrow Bd)$
 $= \{a, b\} \cap \{a, b\} \neq \Phi$
 \therefore 该文法不是LL(1)文法。
且无法改写为LL(1)。

4.2.4 递归下降分析法

递归下降分析法是确定的自上而下分析法。

该分析方法要求文法为LL(1)文法。

4.2.4 递归下降分析法

例1 设有文法**G[E]**:

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

请构造一个识别该文法句子的递归下降分析程序。

4.2.4 递归下降分析法

$$\begin{aligned} E &\rightarrow E + T \mid T \\ T &\rightarrow T * F \mid F \\ F &\rightarrow (E) \mid \text{id} \end{aligned}$$

消去文法左递归，得 $G'[E]$:

$$\begin{aligned} E &\rightarrow TE' \\ E' &\rightarrow +TE' \mid \varepsilon \\ T &\rightarrow FT' \\ T' &\rightarrow *FT' \mid \varepsilon \\ F &\rightarrow (E) \mid \text{id} \end{aligned}$$

$G'[E]:$

$E \rightarrow TE'$

$E' \rightarrow +TE' \mid \varepsilon$

$T \rightarrow FT'$

$T' \rightarrow *FT' \mid \varepsilon$

$F \rightarrow (E) \mid id$

对非终结符 E', T', F 有

$\therefore E \Rightarrow TE'$

$E \Rightarrow TE' \Rightarrow T \Rightarrow FT' \Rightarrow F$

$\Rightarrow (E) \Rightarrow (TE')$

$E \Rightarrow TE' \Rightarrow T \Rightarrow FT'$

$E \Rightarrow TE' \Rightarrow^+ FT' + TE'$

$E \Rightarrow TE' \Rightarrow^+ (FT')$

$\therefore \text{SELECT}(E' \rightarrow +TE') \cap \text{SELECT}(E' \rightarrow \varepsilon)$

$= \text{First}(+TE') \cap \text{Follow}(E')$

$= \{ + \} \cap \{), \$ \} = \Phi$

$\text{SELECT}(T' \rightarrow *FT') \cap \text{SELECT}(T' \rightarrow \varepsilon)$

$= \text{First}(*FT') \cap \text{Follow}(T')$

$= \{ * \} \cap \{), $, + \} = \Phi$

$\text{SELECT}(F \rightarrow id) \cap \text{SELECT}(F \rightarrow (E))$

$= \text{First}(id) \cap \text{First}((E)) = \{ id \} \cap \{ (\} = \Phi$

4.2.4 递归下降分析法

根据文法 $G'[E]$ ，写出相应的递归下降分析程序：

$$\begin{aligned} E &\rightarrow TE' \\ E' &\rightarrow +TE' \mid \varepsilon \\ T &\rightarrow FT' \\ T' &\rightarrow *FT' \mid \varepsilon \\ F &\rightarrow (E) \mid id \end{aligned}$$

```
main()  
{  
    Scanner ();  
    E ();  
    if (sym == '$')  
        printf ("success");  
    else printf ("fail");  
}
```

4.2.4 递归下降分析法

$E \rightarrow TE'$
 $E' \rightarrow +TE' \mid \varepsilon$
 $T \rightarrow FT'$
 $T' \rightarrow *FT' \mid \varepsilon$
 $F \rightarrow (E) \mid \text{id}$

```
E()  
{  
    T();  
    E';  
}
```

```
E'()  
{  
    if (sym == '+')  
    {  
        Scanner();  
        T();  
        E';  
    }  
    else if ((sym != ')') && (sym != '$'))  
        error();  
}
```

4.2.4 递归下降分析法

$E \rightarrow TE'$
 $E' \rightarrow +TE' \mid \varepsilon$
 $T \rightarrow FT'$
 $T' \rightarrow *FT' \mid \varepsilon$
 $F \rightarrow (E) \mid id$

```
T ()  
{  
    F ();  
    T' ();  
}
```

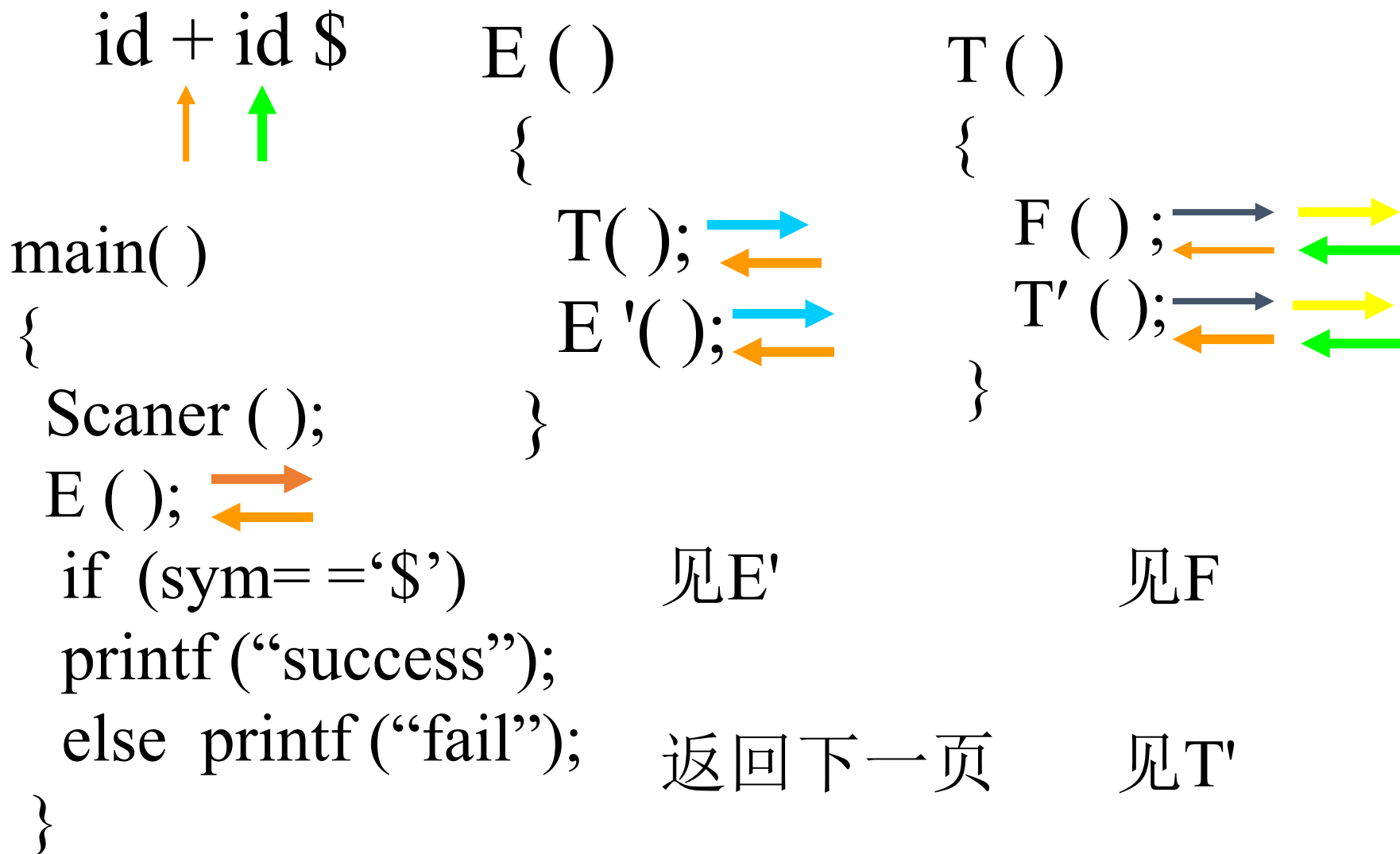
```
T' ()  
{  
    if (sym == '*')  
    {  
        Scanner();  
        F ();  
        T' ();  
    }  
    else if (sym ∉ follow(T'))  
        error();  
}
```

4.2.4 递归下降分析法

$E \rightarrow TE'$
 $E' \rightarrow +TE' \mid \varepsilon$
 $T \rightarrow FT'$
 $T' \rightarrow *FT' \mid \varepsilon$
 $F \rightarrow (E) \mid id$

```
F ( )  
{  
    if (sym == 'id') Scanner ( );  
    else if (sym == '(')  
    {  
        Scanner ( );  
        E ( );  
        if (sym == ')') Scanner ( );  
        else error ( );  
    }  
    else error ( );  
}
```

4.2.4 递归下降分析法



4.2.4 递归下降分析法

```
F ( )  
{  
    if (sym == 'id') Scanner ( );  
    else if (sym == '(')  
    {  
        Scanner ( );  
        E ( );  
        if (sym == ')') Scanner ( );  
        else error ( );  
    }  
    else error ( );  
}
```





返回T

4.2.4 递归下降分析法

```
T' ()  
{  
    follow(T')={+, ), $ }  
    if (sym == '*')  
    { Scanner();  
      F ();  
      T' ();  
    }  
    else if (sym ∉ follow(T'))  
        error();  
}
```

返回T

4.2.4 递归下降分析法

```
E'()  
{  
    if (sym == '+')  
    {  
        Scanner();    返回E  
        T();   见T  
        E'();    
    }  
    else if ((sym != '(') && (sym != '$'))  
        error();  
}
```

返回E

4.2.4 递归下降分析法

$$\begin{aligned} E &\rightarrow E + T \mid T \\ T &\rightarrow T * F \mid F \\ F &\rightarrow (E) \mid \text{id} \end{aligned}$$

若采用扩充的BNF表示法改写文法，得 $G''[E]$ 如下：

$$\begin{aligned} E &\rightarrow T \{ +T \} \\ T &\rightarrow F \{ *F \} \\ F &\rightarrow (E) \mid \text{id} \end{aligned}$$

4.2.4 递归下降分析法

```
E ( )  
{  
    T ( );  
    while ( sym == '+' )  
    {  
        Scanner ( );  
        T ( );  
    }  
}
```

```
T ( )  
{  
    F ( );  
    while ( sym == '*' )  
    {  
        Scanner ( );  
        F ( );  
    }  
}
```

```
E → T { +T }  
T → F { *F }  
F → (E) | id
```

4.2.5 预测分析法与预测分析表的构造

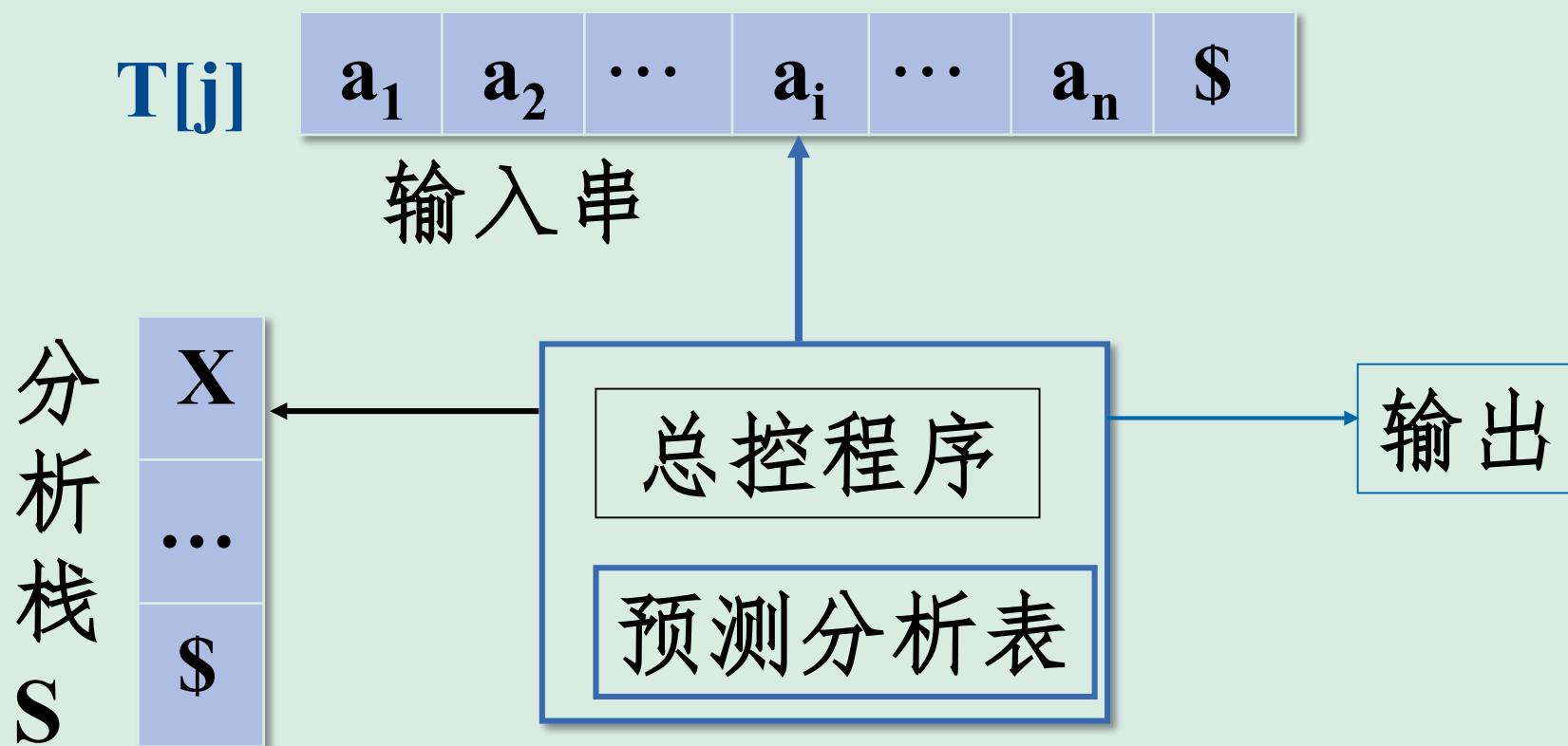


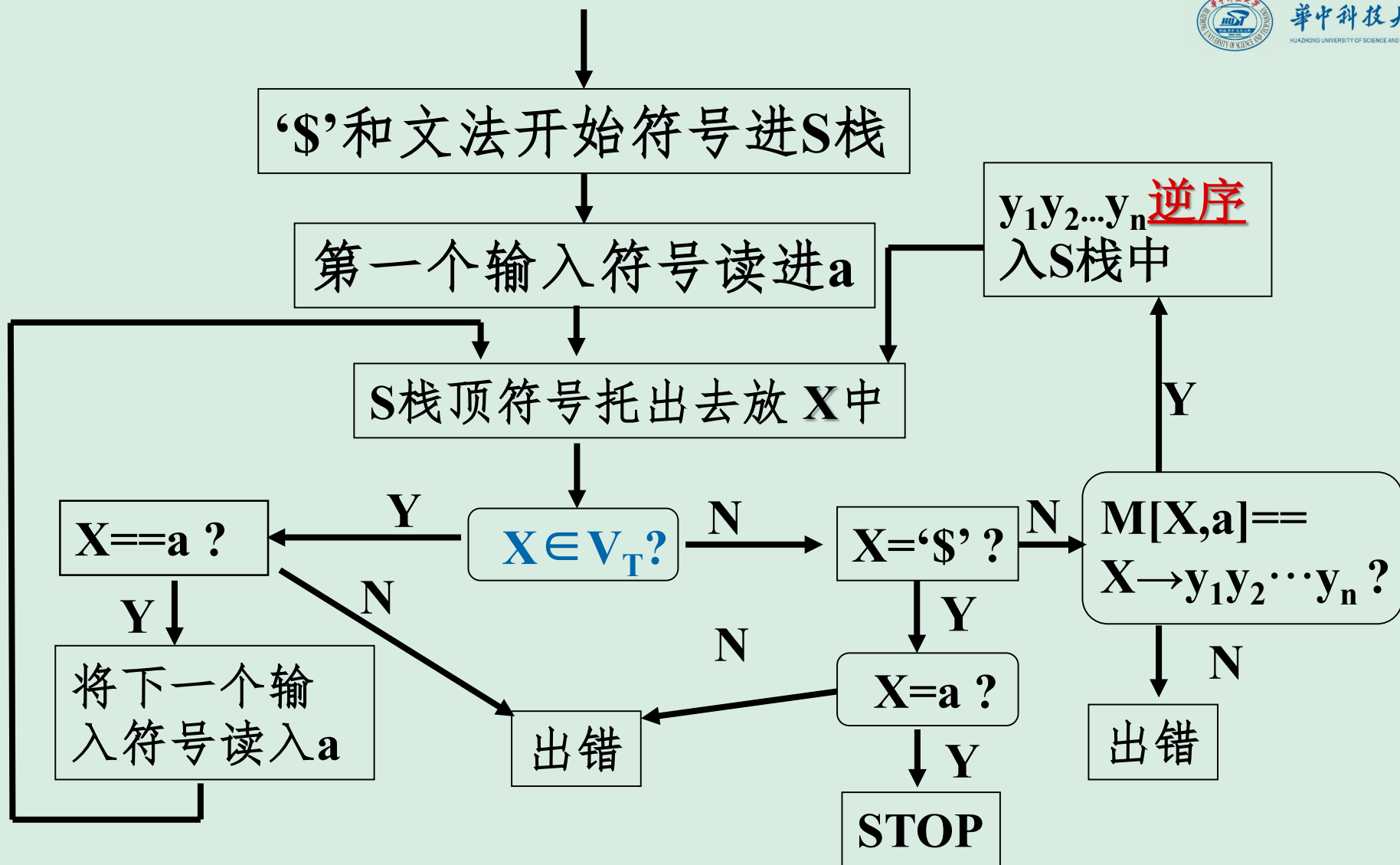
预测分析法(LL(1)分析法)是确定的自上而下分析的另一种方法

4.2.5 预测分析法与预测分析表的构造



预测分析器的逻辑结构





总控程序

4.2.5 预测分析法与预测分析表的构造



构造预测分析表的方法：

输入：文法 G

输出：预测分析表 M

4.2.5 预测分析法与预测分析表的构造



方法：

1. 计算文法 G 的每一非终结符的 First 集、 Follow 集。
2. 对文法的每个规则 $A \rightarrow \alpha$, 若 $a \in \text{First}(\alpha)$, 则置 $M[A, a] = A \rightarrow \alpha$ 。
3. 若 $\varepsilon \in \text{First}(\alpha)$, 则对 $b \in \text{Follow}(A)$, 则置 $M[A, b] = A \rightarrow \alpha$ 。
4. 分析表中未定义的元素标 **error**。表中用空白格表示

4.2.5 预测分析法与预测分析表的构造



例 设有文法 $G[E]$:

$$E \rightarrow TE'$$

$$E' \rightarrow +TE' \mid \varepsilon$$

$$T \rightarrow FT'$$

$$T' \rightarrow *FT' \mid \varepsilon$$

$$F \rightarrow (E) \mid \text{id}$$

试构造该文法的预测分析表。

4.2.5 预测分析法与预测分析表的构造



$E \rightarrow TE'$
 $E' \rightarrow +TE' \mid \varepsilon$
 $T \rightarrow FT'$
 $T' \rightarrow *FT' \mid \varepsilon$
 $F \rightarrow (E) \mid id$

	First	Follow
E	{ (, id }	{), \$ }
E'	{ (, id }	{), \$ }
T	{ (, id }	{ +,), \$ }
T'	{ *, ε }	{ +,), \$ }
F	{ (, id }	{ +,), \$, * }

$$E \rightarrow TE'$$

$$E' \rightarrow +TE' \mid \varepsilon$$

$$T \rightarrow FT'$$

$$T' \rightarrow *FT' \mid \varepsilon$$

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

对规则 $E \rightarrow TE'$

$$\therefore \text{First}(TE') = \{ (, id \}$$

对规则 $E' \rightarrow +TE'$

$$\therefore \text{First}(+TE') = \{ + \}$$

对规则 $E' \rightarrow \varepsilon$

$$\therefore \text{Follow}(E') = \{), \$ \}$$

	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \varepsilon$	$E' \rightarrow \varepsilon$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \varepsilon$	$T' \rightarrow *FT'$		$T' \rightarrow \varepsilon$	$T' \rightarrow \varepsilon$
F	$F \rightarrow id$			$F \rightarrow (E)$		

句子 $id+id*id$ \$ 的分析过程

$E \rightarrow TE'$

$E' \rightarrow +TE' \mid \varepsilon$

$T \rightarrow FT'$

$T' \rightarrow *FT' \mid \varepsilon$

$F \rightarrow (E) \mid id$

分析栈

输入串

\$ E

id+id*id \$

\$ E'T

id+id*id \$

\$ E'T'F

id+id*id \$

\$ E'T'id

id+id*id \$

\$ E'T'

+id*id \$

\$ E'

+id*id \$

\$ E'T+

+id*id \$

\$ E'T

id*id \$

...

...

\$

\$

4.2.5 预测分析法与预测分析表的构造



例 设有文法 $G[S]$

$S \rightarrow a \mid \wedge \mid (T)$

$T \rightarrow T, S \mid S$

1. 消去文法左递归, 改写文法 $G[S]$ 为 $G'[S]$
2. 计算文法 $G'[S]$ 每个非终结符的 **First** 集和 **Follow** 集。
3. 判断文法 $G'[S]$ 是否 **LL(1)** 文法。
4. 试构造文法 $G'[S]$ 的预测分析表。
5. 给出输入串 $(a)\$$ 的分析过程。

4.2.5 预测分析法与预测分析表的构造



分析 引进新的非终结,改写文法,得到文法 **$G'[S]$** :

$$\begin{aligned} S &\rightarrow a \mid \wedge \mid (T) \\ T &\rightarrow ST' \\ T' &\rightarrow ,ST' \mid \varepsilon \end{aligned}$$

$$\begin{aligned} S &\rightarrow a \mid \wedge \mid (T) \\ T &\rightarrow T, S \mid S \end{aligned}$$

根据**First**集和**Follow**集的定义,求出文法每个非终结符的**First**集和**Follow**集:

4.2.5 预测分析法与预测分析表的构造



$S \rightarrow a \mid \wedge \mid (T)$

$T \rightarrow ST'$

$T' \rightarrow ,ST' \mid \varepsilon$

	First	Follow
S	{ a, \wedge , (}	{ \$, , ,) }
T	{ a, \wedge , (}	{) }
T'	{ , , ε }	{) }

$\text{First}(S) = \text{First}(a) \cup \text{First}(\wedge) \cup \text{First}((T))$
 $= \{a, \wedge, (\}$

$\text{Follow}(S) = \{ \$, , ,) \}$

4.2.5 预测分析法与预测分析表的构造



无左递归的文法不一定是LL(1)文法，根据LL(1)文法的判断条件，对非终结符S和T'有：

$$\text{SELECT}(S \rightarrow a) \cap \text{SELECT}(S \rightarrow \wedge)$$

$$= \text{First}(a) \cap \text{First}(\wedge) = \{a\} \cap \{\wedge\} = \Phi$$

$$\text{SELECT}(S \rightarrow a) \cap \text{SELECT}(S \rightarrow (T))$$

$$= \text{First}(a) \cap \text{First}((T)) = \{a\} \cap \{(= \Phi$$

$$\text{SELECT}(S \rightarrow \wedge) \cap \text{SELECT}(S \rightarrow (T))$$

$$= \text{First}(\wedge) \cap \text{First}((T)) = \{\wedge\} \cap \{(= \Phi$$

4.2.5 预测分析法与预测分析表的构造



$$\begin{aligned} S &\rightarrow a \mid \wedge \mid (T) \\ T &\rightarrow ST' \\ T' &\rightarrow ,ST' \mid \varepsilon \end{aligned}$$

$$\text{SELECT}(T' \rightarrow ,ST') \cap \text{SELECT}(T' \rightarrow \varepsilon)$$

$$= \text{First}(,ST') \cap \text{Follow}(T')$$

$$= \{ , \} \cap \{) \} = \Phi$$

所以文法 $G'[S]$ 是 LL(1) 文法。

对规则 $S \rightarrow a$

$\therefore \text{First}(a) = \{a\}$

对规则 $S \rightarrow \wedge$

$\therefore \text{First}(\wedge) = \{\wedge\}$

对规则 $S \rightarrow (T)$

$\therefore \text{First}((T)) = \{($

对规则 $T \rightarrow ST'$

$\therefore \text{First}(ST') = \{a, \wedge, ($

对规则 $T' \rightarrow ,ST'$

$\therefore \text{First}(,ST') = \{ , \}$

对规则 $T' \rightarrow \varepsilon$

$\therefore \text{Follow}(T') = \{) \}$

	a	\wedge	()	,	\$
S	$S \rightarrow a$	$S \rightarrow \wedge$	$S \rightarrow (T)$			
T	$T \rightarrow ST'$	$T \rightarrow ST'$	$T \rightarrow ST'$			
T'	$T' \rightarrow ,ST'$			$T' \rightarrow \varepsilon$	$T' \rightarrow ,ST'$	

4.2.5 预测分析法与预测分析表的构造



符号串(a)\$ 分析过程如表：

$$S \rightarrow a \mid \wedge \mid (T)$$

$$T \rightarrow ST'$$

$$T' \rightarrow ,ST' \mid \varepsilon$$

	a	\wedge	()	,	\$
S	$S \rightarrow a$	$S \rightarrow \wedge$	$S \rightarrow (T)$			
T	$T \rightarrow ST'$	$T \rightarrow ST'$	$T \rightarrow ST'$			
T	$T \rightarrow ,ST'$			$T' \rightarrow \varepsilon$	$T' \rightarrow ,ST'$	

分析栈	输入串
\$S	(a)\$
\$)T((a) \$
\$)T	a) \$
\$)T'S	a) \$
\$)T'a	a) \$
\$)T') \$
\$)) \$
\$	\$

4.2.5 预测分析法与预测分析表的构造



若一个文法**G**的分析表**M**不含多重定义元素, 则该文法是**LL(1)**文法。

预测分析法的主要优点:

LL(1)分析表比其他分析方法中的分析表相对要小一些。

4.2.5 预测分析法与预测分析表的构造



$$\begin{aligned} E &\rightarrow TE' & E' &\rightarrow +TE' \mid \epsilon \\ T &\rightarrow FT' & T' &\rightarrow *FT' \mid \epsilon \\ F &\rightarrow (E) \mid i \end{aligned}$$

返回

返回

	i	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \epsilon$	$E' \rightarrow \epsilon$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \epsilon$	$T' \rightarrow *FT'$		$T' \rightarrow \epsilon$	$T' \rightarrow \epsilon$
F	$F \rightarrow i$			$F \rightarrow (E)$		

4.3 自下而上分析法的一般原理



符号栈

$\$ t_1 t_2 t_3 \dots t_j \dots t_{i-1} t_i$

$t_{i+1} \dots t_n \$$

可规约串

$\$ S$

A

$\$$

(1) $S \rightarrow aAcBe$

(2) $A \rightarrow b$

(3) $A \rightarrow A b$

(4) $B \rightarrow d$

输入串 $abbcde\$$

符号栈

输入串

\$

abbcde\$

\$a

bbbcde\$

\$ab

bcde\$

\$aA

bcde\$

\$aAb

cde\$

\$aA

cde\$

\$aAc

de\$

\$aAcd

e\$

\$aAcB

e\$

\$aAcBe

\$

\$S

\$

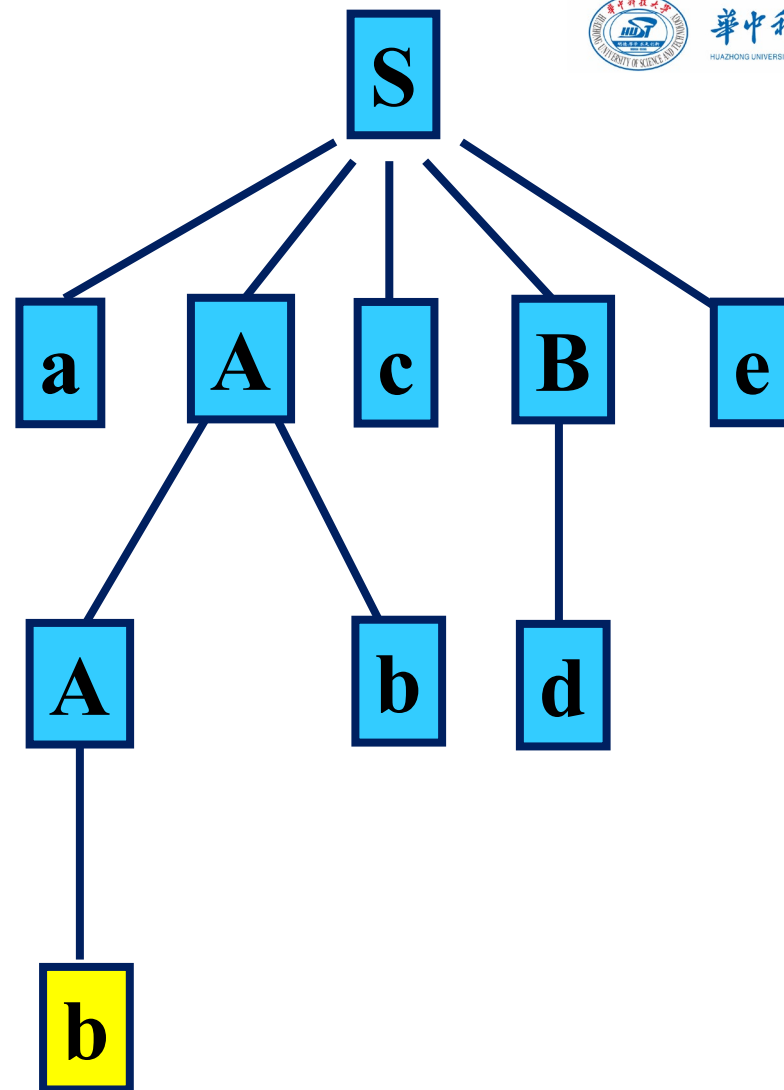
(1) $S \rightarrow aAcBe$

(2) $A \rightarrow b$

(3) $A \rightarrow Ab$

(4) $B \rightarrow d$ 输入串 **a**bbcd**e**\$

符号栈	输入串
\$	abbcde\$
\$a	bbcd e \$
\$a b	bcde\$
\$aA	bcde\$
\$aAb	cde\$
\$aA	cde\$
\$aAc	de\$
\$aAcd	e\$
\$aAcB	e\$
\$aAcBe	\$
\$S	\$



4.4.1 方法概述

$$E \rightarrow E + E \mid E * E \mid (E) \mid id$$

第一个规范归约

(1) $id + id * id$
(2) $E + id * id$
(3) $E + E * id$
(4) $E + E * E$
(5) $E + E$
(6) E

第二个规范归约

(1) $id + id * id$
(2) $E + id * id$
(3) $E + E * id$
(4) $E * id$
(5) $E * E$
(6) E

4.4.1 方法概述

1. 任何两个相邻终结符号a 和 b之间的优先关系有三种：

$a \triangleleft b$

a的优先级低于b

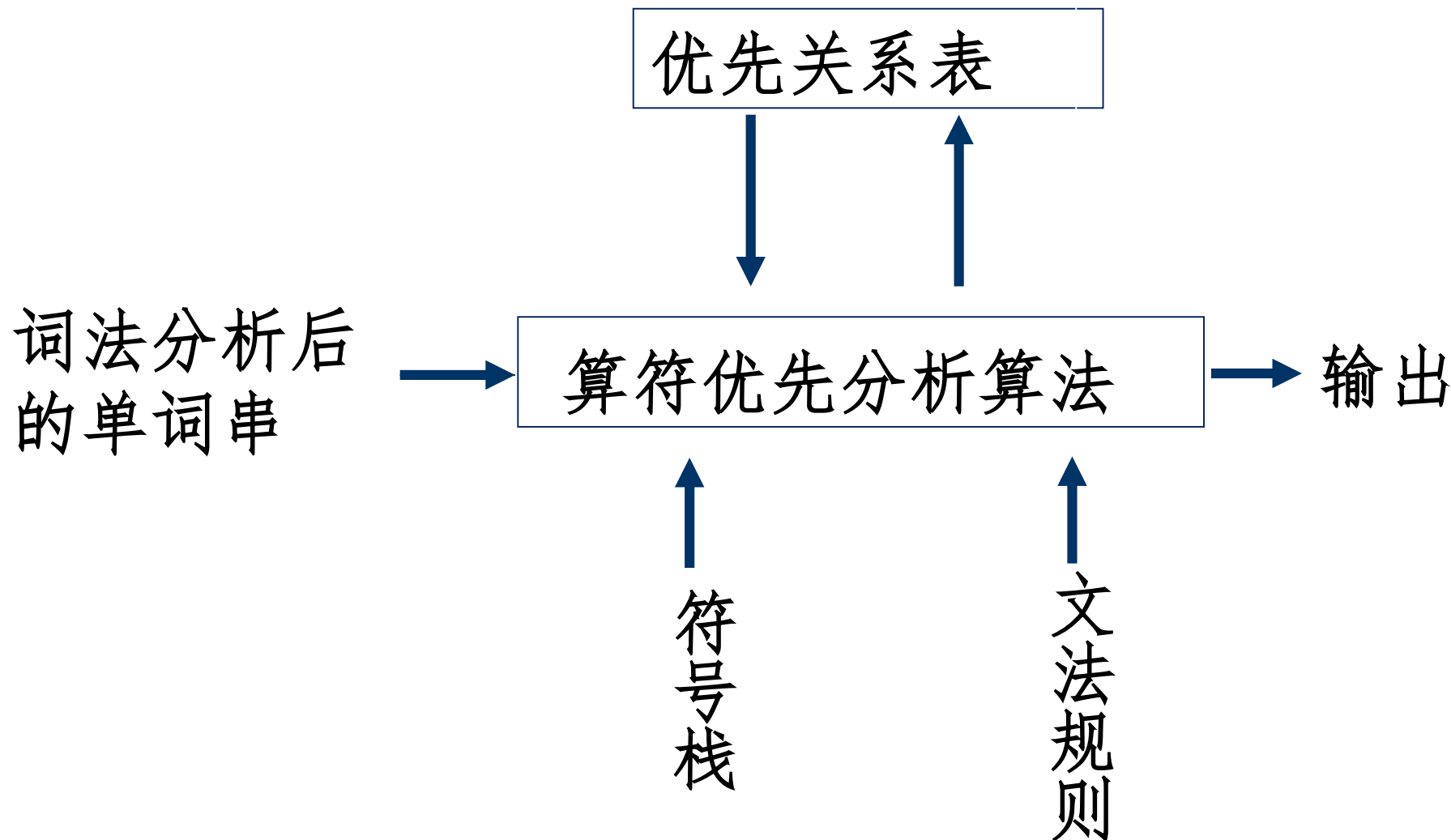
$a \equiv b$

a的优先级等于b

$a \triangleright b$

a的优先级高于b

4.4.1 方法概述



4.4.2 算符优先文法的定义

算符优先文法的定义

1. 算符文法的定义

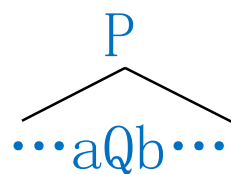
设有文法 G ，若 G 中没有形如 $U \rightarrow \dots VW \dots$ 的规则，其中 V 和 W 为非终结符，则称 G 为算符文法，也称OG文法。

4.4.2 算符优先文法的定义

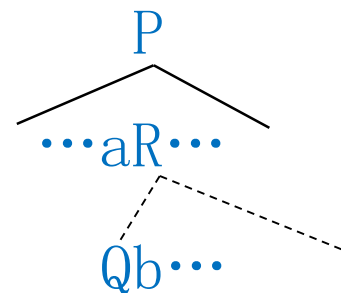
G是一个算符文法

(1) $a \preceq b$ 当且仅当G中含有形如

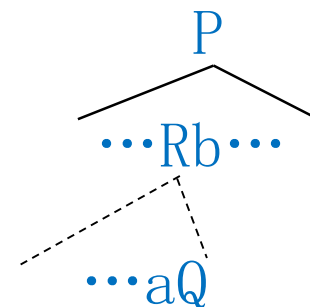
$P \rightarrow \dots ab \dots$ 或 $P \rightarrow \dots aQb \dots$ 的规则。



(2) $a \prec b$ 当且仅当G中含有形如 $P \rightarrow \dots aR \dots$ 的规则，且 $R \xRightarrow{+} b \dots$ 或 $R \xRightarrow{+} Qb \dots$



(3) $a \succ b$ 当且仅当G中含有形如 $P \rightarrow \dots Rb \dots$ 的规则，且 $R \xRightarrow{+} \dots a$ 或 $R \xRightarrow{+} \dots aQ$



4.4.2 算符优先文法的定义

2. 算符优先文法的定义

设有一个不含 ε 规则的算符文法G，如果任意两个终结符号对 (a, b) 在 \lessdot 、 \gtrdot 和 \equiv 三种关系中只有一种关系成立，则称G是算符优先文法，也称OPG文法。

4.4.2 算符优先文法的定义

对前述算术表达式的文法是否为**OPG**?

$E \rightarrow E + E \mid E * E \mid (E) \mid id$

$+ < * \text{ 同时 } + > *$

$G[E]$ 不是算符优先文法

4.4.2 算符优先文法的定义

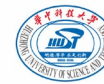
算术表达式的文法是否为OPG?

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

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

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

4.4.3 算符优先关系表的构造



首先对文法每个非终结符 A 定义两个集合：

$$\text{FirstVt}(A) = \{b \mid A \xRightarrow{+} b \dots \text{或} A \xRightarrow{+} Bb \dots, \\ b \in V_T, B \in V_N\}$$

$$\text{LastVt}(A) = \{a \mid A \xRightarrow{+} \dots a \text{ 或} A \xRightarrow{+} \dots aB, \\ a \in V_T, B \in V_N\}$$

4.4.3 算符优先关系表的构造

构造文法 G 的优先关系表的算法：

输入：算符优先文法 G

输出：关于文法 G 的优先关系表

for (每个产生式 $A \rightarrow x_1 x_2 \dots x_n$)

for ($i=1$; $i \leq n-1$; $i++$)

{ if (x_i 和 x_{i+1} 均 $\in V_T$) 置 $x_i \Leftarrow x_{i+1}$

if ($i \leq n-2$ 且 x_i 和 x_{i+2} 都 $\in V_T$, 而
 $x_{i+1} \in V_N$) 置 $x_i \Leftarrow x_{i+2}$

if ($x_i \in V_T$, $x_{i+1} \in V_N$)

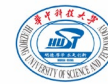
for ($\text{FirstVt}(x_{i+1})$ 中的每个 b) 置 $x_i \Leftarrow b$;

if ($x_i \in V_N$, $x_{i+1} \in V_T$)

for ($\text{LastVt}(x_i)$ 中的每个 a) 置 $a \Rightarrow x_{i+1}$;

}

4.4.3 算符优先关系表的构造



例 设有表达式的文法G[E]:

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

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

$$F \rightarrow (E) \mid \text{id}$$

构造该文法的算符优先关系表。

4.4.3 算符优先关系表的构造

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

	FirstVt	LastVt
E	{ *, +, (, id }	{ *, +,), id }
T	{ *, (, id }	{ *,), id }
F	{ (, id }	{), id }

4.4.3 算符优先关系表的构造

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid id$

$\$ \leq \text{FirstVt}(E)$

$\text{LastVt}(E) \geq \$$

$\$ = \$$

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

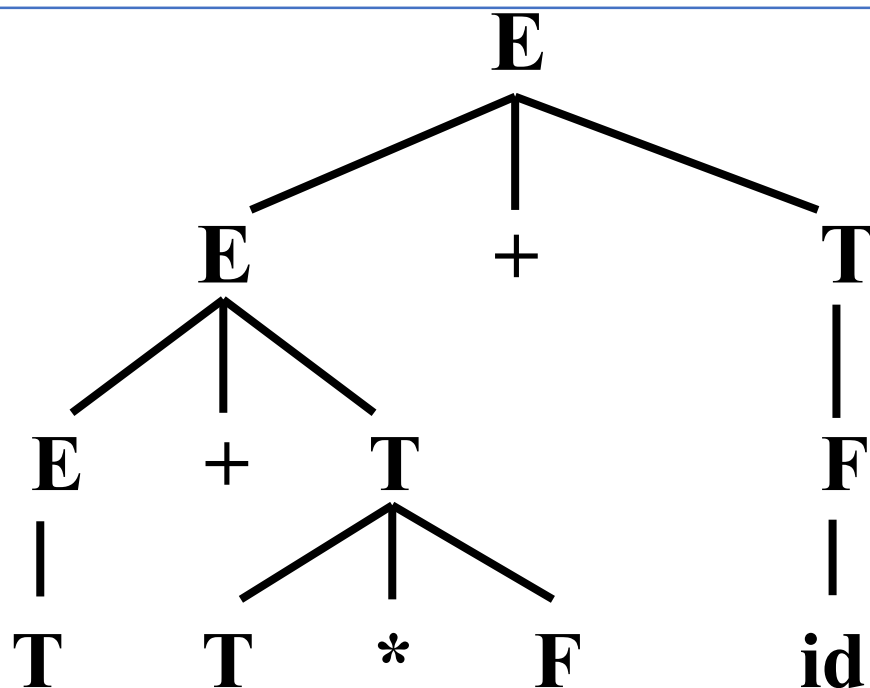
4.4.4 算符优先分析算法的设计



1. 最左素短语

句型的**素短语**是一种短语，它至少包含一个终结符，并且除自身之外，不再包含其它的素短语。句型最左边的素短语称**最左素短语**。

4.4.4 算符优先分析算法的设计

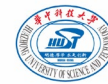


短语有：
 $T + T * F + id$
 $T + T * F$
 T
 $T * F$
 id

$T * F$ 和 id 是素短语

$T * F$ 是最左素短语

4.4.4 算符优先分析算法的设计



对算符优先文法G有如下定理：
一个算符优先文法G的任何句型的最左素短语是满足下列条件的最左子串：

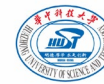
$$N_i a_i N_{i+1} a_{i+1} \dots a_j N_{j+1}$$

$$a_{i-1} < a_i$$

$$a_i = a_{i+1}, \dots, a_{j-1} = a_j$$

$$a_j > a_{j+1}$$

4.4.4 算符优先分析算法的设计



$$a_{i-1} < a_i$$

$$a_i = a_{i+1}, \dots, a_{j-1} = a_j$$

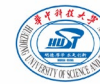
$$a_j > a_{j+1}$$

对上述句型 $\$ T + T * F + id \$$ 写成算符优先分析形式为： $\$ N_1 a_1 N_2 a_2 N_3 a_3 a_4 \$$

因为 $\$ < + < * > + < id > \$$

所以 $N_2 a_2 N_3$ 即 $T * F$ 是最左素短语。

4.4.4 算符优先分析算法的设计



3. 算符优先分析程序的设计

基本思想:

$$a_{i-1} < a_i$$

$$a_i = a_i \dots a_{j-1} = a_j$$

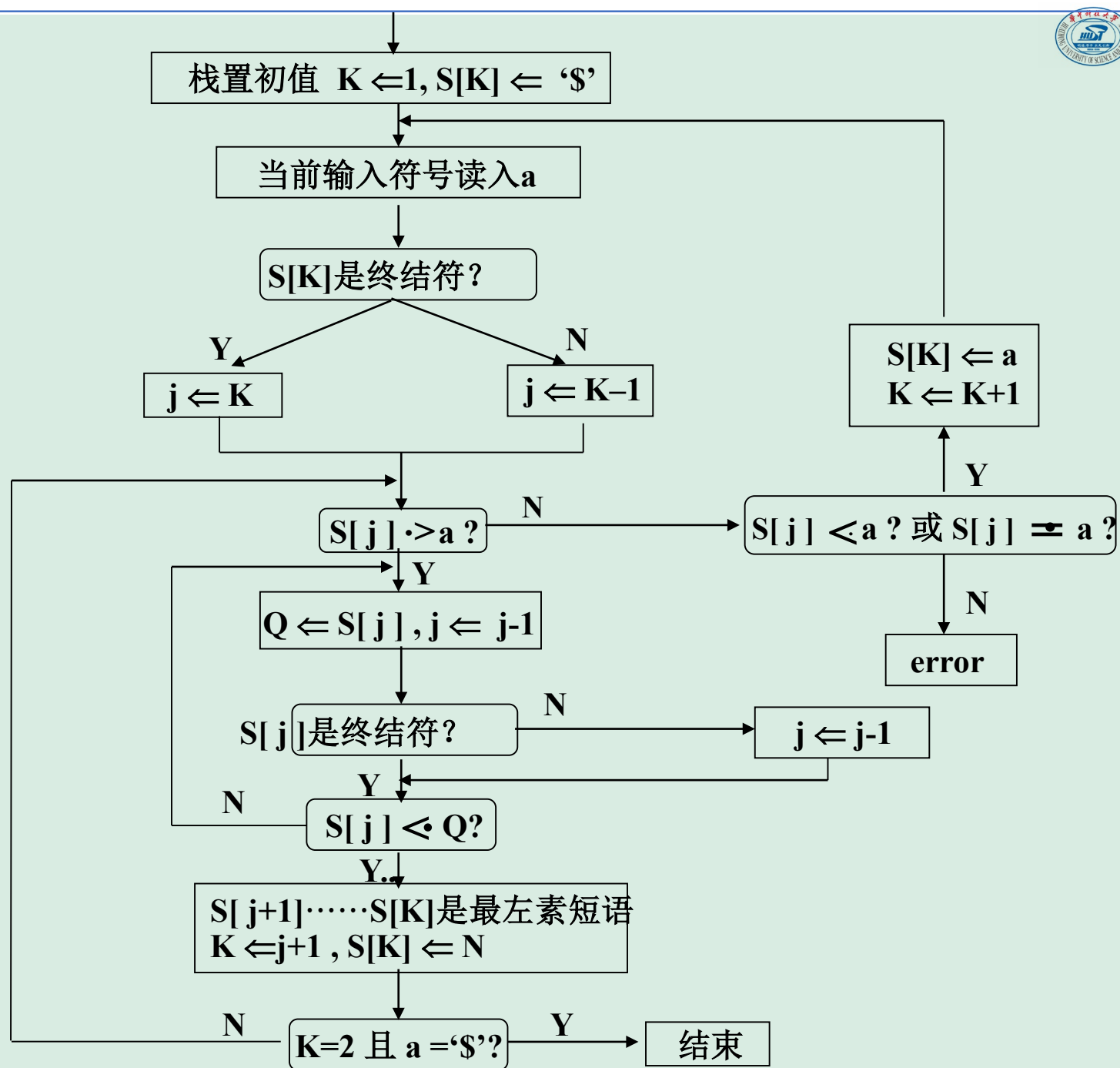
$$a_j > a_{j+1}$$

符号栈 $< =$ 优先关系

$$\$ t_1 t_2 t_3 \dots t_j t_{j+1} \dots t_i > t_{i+1} \dots t_n \$$$

↑ ↑
头 尾

最左素短语

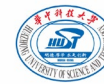


4.4.4 算符优先分析算法的设计



S栈	优先关系	当前符号	输入流	动作
\$	\lessdot	id	+ id \$	移进
\$ id	\gtrdot	+	id \$	归约
\$ N	\lessdot	+	id \$	移进
\$ N +	\lessdot	id	\$	移进
\$ N + id	\gtrdot	\$		归约
\$ N + N	\gtrdot	\$		归约
\$ N	\vdash	\$		结束

4.4.6 算符优先分析法的局限性



算符优先文法的缺陷举例

$$\mathbf{A} \rightarrow \mathbf{A};\mathbf{D} \mid \mathbf{D}$$

$$\mathbf{D} \rightarrow \mathbf{D}(\mathbf{E}) \mid \mathbf{F}$$

$$\mathbf{F} \rightarrow \mathbf{a} \mid (\mathbf{A})$$

$$\mathbf{E} \rightarrow \mathbf{E}+\mathbf{A} \mid \mathbf{A}$$

待识别串： $(\mathbf{a}+\mathbf{a})$