

# 第三章：语法分析

## LL(1)语法分析



# 1. LL(1)语法分析原理

## ◆ 基本思想

从左到右扫描，按最左推导的方式推出输入流

## ◆ LL(1)文法定义：

对于文法G中任一非终极符A，其任意两个产生式  $A \rightarrow \alpha$  和  $A \rightarrow \beta$ ，都要满足下面条件：

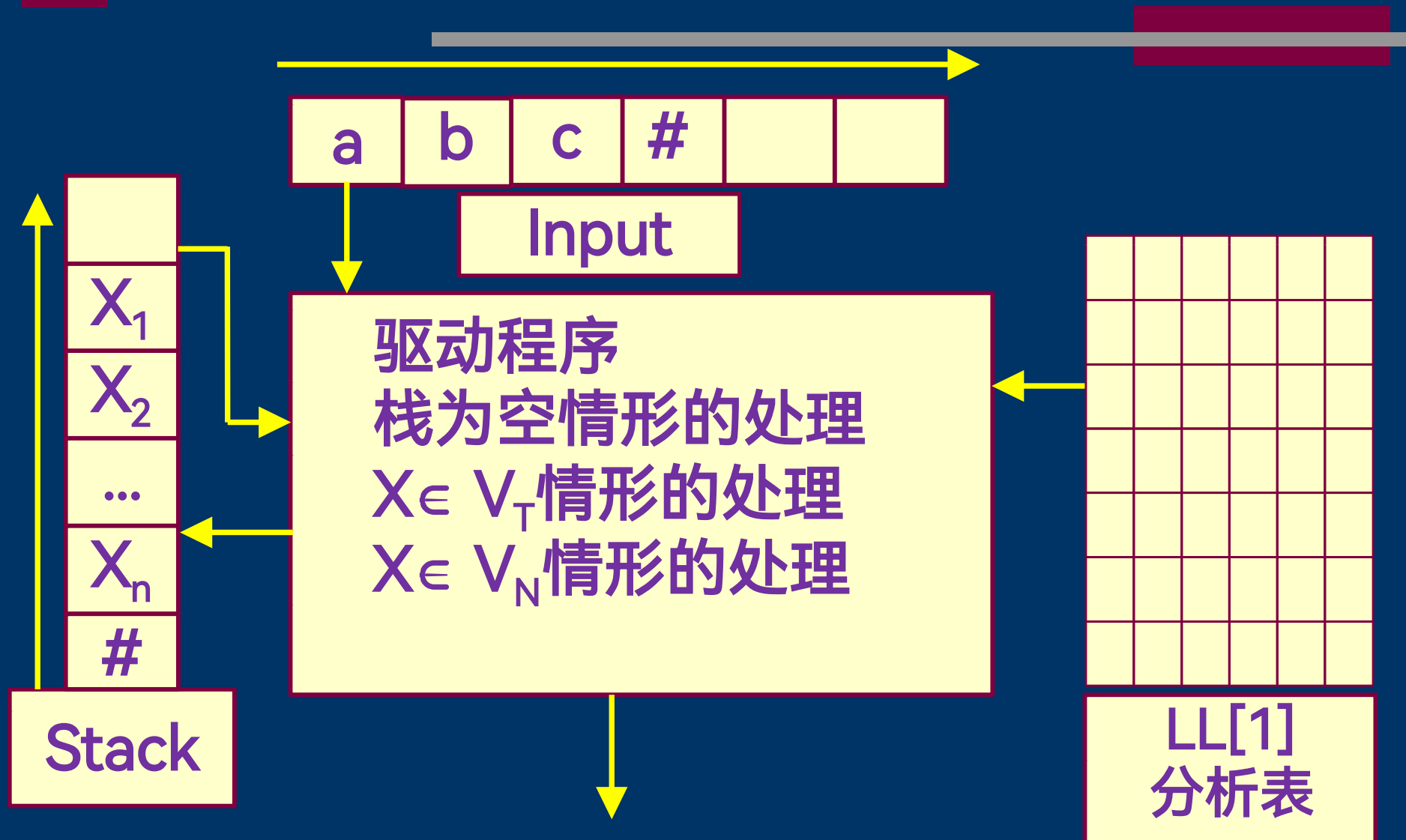
$$\text{Predict}(A \rightarrow \alpha) \cap \text{Predict}(A \rightarrow \beta) = \emptyset$$

# 1. LL(1)语法分析原理

## □ 语法分析的动作

- ❖ **替换** 当分析栈的第一个符号是 $V_N$ 时,就要用规则进行推导,用规则右部将其替换
- ❖ **匹配** 分析栈第一个符号是 $V_T$ 时,与输入流中的第一个符号进行匹配
- ❖ **成功**
- ❖ **出错、失败**

# 1. LL(1)语法分析原理



## 2. LL(1)分析表的构造

- 构造的目的是为了让分析更方便
  - 结构:  $T: V_N \times V_T \rightarrow P \cup \{\text{Error}\}$   
具体:  $T(A, t) = A \rightarrow a$  若  $t \in \text{Predict}(A \rightarrow a)$   
 $T(A, t) = \text{Error}$  否则
- 特殊的: # 作为一个终极符

## 2. LL(1)分析表的构造

- 【1】  $E \rightarrow TE'$
- 【2】  $E' \rightarrow +TE'$
- 【3】  $E' \rightarrow \varepsilon$
- 【4】  $T \rightarrow FT'$
- 【5】  $T' \rightarrow *FT'$
- 【6】  $T' \rightarrow \varepsilon$
- 【7】  $F \rightarrow (E)$
- 【8】  $F \rightarrow i$

{i, (}

{+}

{#, )}

{i, (}

{\*}

{+, #, )}

{(}

{i}

{#, )}

	i	+	*	(	)	#
E	1			1		
E'		2			3	3
T	4			4		
T'		6	5		6	6
F	8			7		

$First(F) = \{ (, \varepsilon \}$   $First(T') = \{ \#, \varepsilon \}$

## 3.1 驱动程序的设计

□ 分析的初始格局 ( $S\#, a_1...a_n\#$ )

□ 一般格局 ( $X_1...X_m\#, a_1...a_n\#$ )

设存在格局为 ( $X_1...X_m\#, a_1...a_n\#$ )

❖ 若  $X_1 \in V_T$  &  $X_1 = a_1$  则有 ( $X_2...X_m\#, a_2...a_n\#$ )

❖ 若  $X_1 \in V_N$  则查表, 若  $T(X_1, a_1) = X_1 \rightarrow \alpha$ , 格局为 ( $\alpha X_2...X_m\#, a_1...a_n\#$ ), 若  $T(X_1, a_1) = \text{error}$ , 则报错。

❖ 若格局为 ( $\#, \#$ ), 则分析成功

❖ 其他情况 报错

## 分析栈S

## 输入流T

## 矩阵元素

E #

TE' #

FT' E' #

iT' E' #

T' E' #

E' #

+TE' #

TE' #

FT' E' #

iT' E' #

T' E' #

\*FT' E' #

FT' E' #

iT' E' #

T' E' #

E' #

#

i + i \* i #

i + i \* i #

i + i \* i #

i + i \* i #

+ i \* i #

+ i \* i #

+ i \* i #

i \* i #

i \* i #

i \* i #

\* i #

\* i #

i #

i #

#

#

#

T[E, i] = [1]

T[T, i] = [4]

T[F, i] = [8]

Match

T[T', +] = [6]

T[E', +] = [2]

Match

T[T, i] = [4]

T[F, i] = [8]

Match

T[T', \*] = [5]

Match

T[F, i] = [8]

Match

T[T', #] = [6]

T[E', #] = [3]

ok



## 3.2 注意的一些问题

- 错误的处理
- LL(1)方法对文法的限制

假如不满足限定：

- ❖ 进行文法等价变换，使其满足我们的限定
- ❖ 不是所有的文法都适用于该方法，有时进行文法等价变换会破坏其可读性
- 通常来说，对于高级语言的语法分析也足够用了。

# 判断是否为LL(1)文法

1.  $S \rightarrow (SS' | \varepsilon$   
 $S' \rightarrow ) | \varepsilon$

$$\begin{aligned} \text{Predict}(S \rightarrow (SS') &= \{ ( \} \\ \text{Predict}(S \rightarrow \varepsilon) &= \{ ), \# \} \\ \text{Predict}(S' \rightarrow ) &= \{ ) \} \\ \text{Predict}(S' \rightarrow \varepsilon) &= \{ (, \# \} \end{aligned}$$

2.  $S \rightarrow (S | S'$   
 $S' \rightarrow (S') | \varepsilon$

$$\begin{aligned} \text{Predict}(S \rightarrow (S) &= \{ ( \} \\ \text{Predict}(S \rightarrow S') &= \{ (, \# \} \\ \text{Predict}(S' \rightarrow (S')) &= \{ ( \} \\ \text{Predict}(S' \rightarrow \varepsilon) &= \{ (, ), \# \} \end{aligned}$$

# 构造LL(1)分析表

$S \rightarrow aBc[1]$

$S \rightarrow bAB[2]$

$A \rightarrow aAb[3]$

$A \rightarrow b[4]$

$B \rightarrow b[5]$

$B \rightarrow \varepsilon[6]$