

CS30017 编译

# 第四讲：自顶向下解析

徐辉

xuh@fudan.edu.cn



# 自顶向下解析

一、问题定义

二、LL(1)文法和解析

三、PEG文法和解析

四、Earley算法

# 一、问题定义

---

# 自顶向下解析

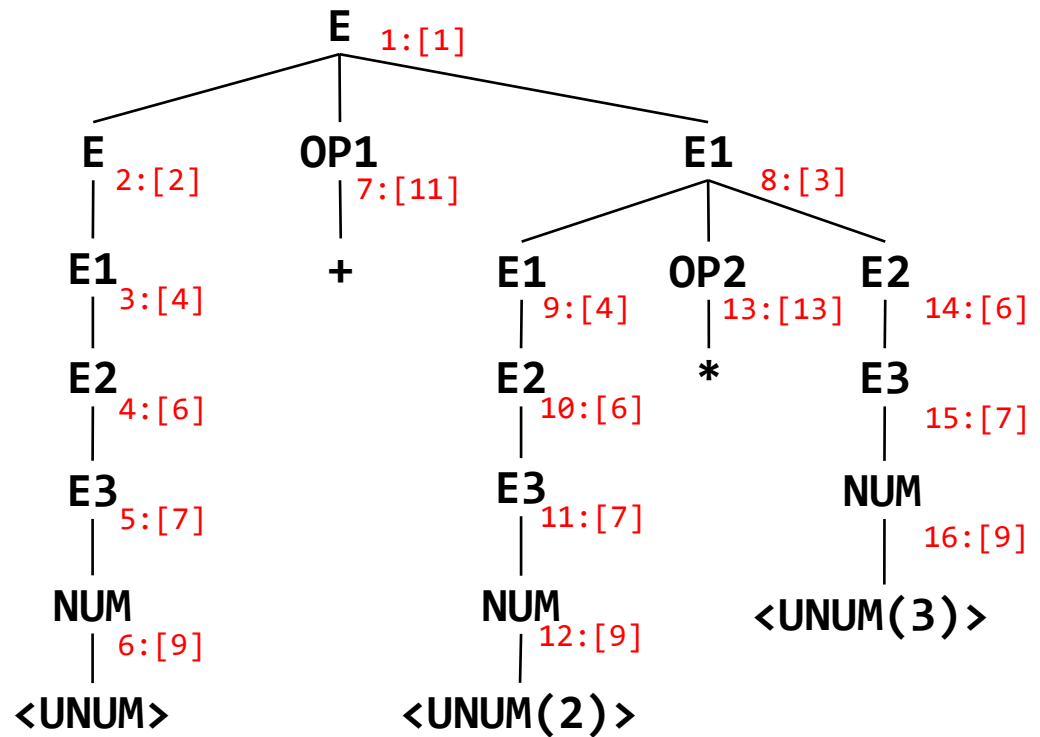
- 已知一套CFG语法规则和待解析的句子
- 应用语法规则（从左至右）逐步展开每个非终结符
  - 如果能得到目标句子=>解析成功
  - 如果不能得到目标句子=>解析失败
- 如无二义性问题，则语法解析树唯一

# 自顶向下解析示例

语法规则：

词元流：  $\langle \text{UNUM}(1) \rangle \langle \text{ADD} \rangle \langle \text{UNUM}(2) \rangle \langle \text{MUL} \rangle \langle \text{UNUM}(3) \rangle$

- [1]  $E \rightarrow E \text{ OP1 } E1$
- [2]  $E \rightarrow E1$
- [3]  $E1 \rightarrow E1 \text{ OP2 } E2$
- [4]  $E1 \rightarrow E2$
- [5]  $E2 \rightarrow E3 \text{ OP3 } E2$
- [6]  $E2 \rightarrow E3$
- [7]  $E3 \rightarrow \text{NUM}$
- [8]  $E3 \rightarrow '(' E ')'$
- [9]  $\text{NUM} \rightarrow \langle \text{UNUM} \rangle$
- [10]  $\text{NUM} \rightarrow '-' \langle \text{UNUM} \rangle$
- [11]  $\text{OP1} \rightarrow '+'$
- [12]  $\text{OP1} \rightarrow '-'$
- [13]  $\text{OP2} \rightarrow '*'$
- [14]  $\text{OP2} \rightarrow '/'$
- [15]  $\text{OP3} \rightarrow '^'$



语法解析树

# 如何自动生成语法解析树？

- 关键问题：如何判断当前应采用哪条规则展开？暴力搜索？
- 可能存在递归：递归何时终止？
- 局部词元序列可能存在多种展开方式：选择错误需要回溯

$s[0] = \circ \langle \text{UNUM} \rangle '+' \langle \text{UNUM} \rangle '*' \langle \text{UNUM} \rangle$   
          ↑  
      当前位置

E → ◦ E1	
E1 → ◦ E2	
E2 → ◦ E3 OP3 E2	E2 → ◦ E3
E3 → ◦ NUM	
NUM → ◦ <UNUM>	

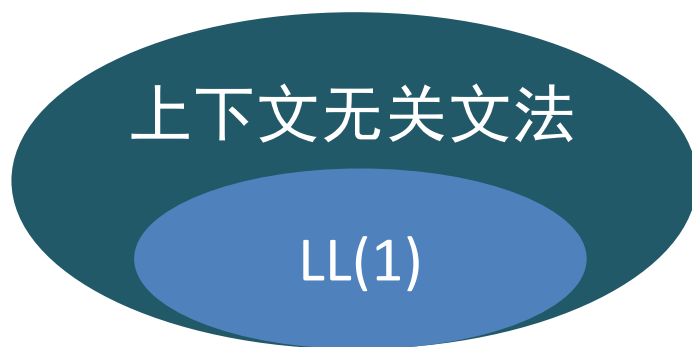
E → ◦ E OP1 E1	
E → ◦ E1	
E1 → ◦ E2	
E2 → ◦ E3 OP3 E2	E2 → ◦ E3
E3 → ◦ NUM	
NUM → ◦ <UNUM>	

## 二、LL(1)文法和解析

---

# 通过限制文法复杂度提升解析效率

- LL(1)文法的基本要求：
  - 无左递归
  - 无回溯
- LL(1)文法： Left-to-Right, Leftmost, 前瞻一个字符无回溯
- LL(k)文法： Left-to-Right, Leftmost, 前瞻k个字符无回溯



# 左递归问题

- 一条规则中右侧的第一个符号与左侧符号相同
- 采用最左推导时解析算法无限递归，不终止

```
[1] E → E OP1 E1
[2] E → E1
[3] E1 → E1 OP2 E2
[4] E1 → E2
[5] E2 → E3 OP3 E2
[6] E2 → E3
[7] E3 → NUM
[8] E3 → '(' E ') '
[9] NUM → <UNUM>
[10] NUM → '-' <UNUM>
[11] OP1 → '+'
[12] OP1 → '-'
[13] OP2 → '*'
[14] OP2 → '/'
[15] OP3 → '^'
```

# 消除左递归

- 改写语法规则，使新旧规则等价：

- 1) 引入新的非终结符 $E'$ ，并规定其可以推出空串 $\epsilon$
- 2) 为 $E'$ 构造产生式，将原左递归产生式右侧去掉开头的 $E$ ，后接 $E'$ 构成
- 3) 为 $E$ 构造产生式，其右侧为原非左递归的 $E$ 产生式右侧，后接 $E'$ 构成

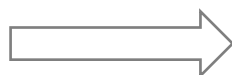
$$\begin{array}{l} E \rightarrow E \alpha \\ E \rightarrow \beta \end{array} \Rightarrow \begin{array}{l} E \rightarrow \beta E' \\ E' \rightarrow \alpha E' \\ E' \rightarrow \epsilon \end{array} \quad \{\beta\alpha, \beta\alpha\alpha, \beta\alpha\alpha\alpha, \dots\}$$

$$\begin{array}{l} E \rightarrow E \alpha \\ E \rightarrow \beta \\ E \rightarrow \gamma \end{array} \Rightarrow \begin{array}{l} E \rightarrow \beta E' \\ E \rightarrow \gamma E' \\ E' \rightarrow \alpha E' \\ E' \rightarrow \epsilon \end{array} \quad \{\beta\alpha, \gamma\alpha, \beta\alpha\alpha, \gamma\alpha\alpha, \dots\}$$

# 应用：消除左递归

[1]  $E \rightarrow E \text{ OP1 } E1$   
[2]  $E \rightarrow E1$   
[3]  $E1 \rightarrow E1 \text{ OP2 } E2$   
[4]  $E1 \rightarrow E2$   
[5]  $E2 \rightarrow E3 \text{ OP3 } E2$   
[6]  $E2 \rightarrow E3$   
[7]  $E3 \rightarrow \text{NUM}$   
[8]  $E3 \rightarrow '( \text{ E } )'$   
[9]  $\text{NUM} \rightarrow \langle \text{UNUM} \rangle$   
[10]  $\text{NUM} \rightarrow '- ' \langle \text{UNUM} \rangle$   
[11]  $\text{OP1} \rightarrow '+'$   
[12]  $\text{OP1} \rightarrow '- '$   
[13]  $\text{OP2} \rightarrow '*'$   
[14]  $\text{OP2} \rightarrow '/'$   
[15]  $\text{OP3} \rightarrow '^'$

消除左递归



[1]  $E \rightarrow E1 \text{ E}'$   
[2]  $\text{E}' \rightarrow \text{OP1 } E1 \text{ E}'$   
[3]  $\text{E}' \rightarrow \epsilon$   
[4]  $E1 \rightarrow E2 \text{ E1}'$   
[5]  $E1' \rightarrow \text{OP2 } E2 \text{ E1}'$   
[6]  $E1' \rightarrow \epsilon$   
[7]  $E2 \rightarrow E3 \text{ OP3 } E2$   
[8]  $E2 \rightarrow E3$   
[9]  $E3 \rightarrow \text{NUM}$   
[10]  $E3 \rightarrow '( \text{ E } )'$   
[11]  $\text{NUM} \rightarrow \langle \text{UNUM} \rangle$   
[12]  $\text{NUM} \rightarrow '- ' \langle \text{UNUM} \rangle$   
[13]  $\text{OP1} \rightarrow '+'$   
[14]  $\text{OP1} \rightarrow '- '$   
[15]  $\text{OP2} \rightarrow '*'$   
[16]  $\text{OP2} \rightarrow '/'$   
[17]  $\text{OP3} \rightarrow '^'$

# 注意间接左递归问题

- 对产生式进行替换展开
- 消除所得语法规则中的左递归

$$\begin{array}{l} E \rightarrow \alpha \\ \alpha \rightarrow \beta \gamma \\ \beta \rightarrow E \end{array} \quad \Longrightarrow \quad E \rightarrow E \gamma$$

# 无回溯语法

- 任意非终结符的任意两个产生式生成的首个终结符均不同
- 前瞻一个终结符总能选择正确的规则
- 消除语法规则选择时的不确定性，避免回溯

$$[1] S \rightarrow \alpha \rightarrow \dots \rightarrow a\gamma$$

$$[2] S \rightarrow \beta \rightarrow \dots \rightarrow b\delta$$

# 消除回溯：提取左公因子

- 引入新的非终结符 $E'$
- 对存在回溯问题的一组产生式提取共同前缀，置于 $E'$ 之前
- 为 $E'$ 编写产生式规则，使新旧规则等价

$$E \rightarrow \alpha\beta_1 | \alpha\beta_2 | \dots | \alpha\beta_n | \gamma_1 | \dots | \gamma_j$$

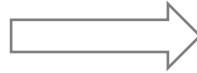


$$\begin{aligned} E &\rightarrow \alpha E' | \gamma_1 | \dots | \gamma_j \\ E' &\rightarrow \beta_1 | \beta_2 | \dots | \beta_n \end{aligned}$$

# 应用：消除回溯

[1]  $E \rightarrow E_1 E'$   
[2]  $E' \rightarrow OP_1 E_1 E'$   
[3]  $E' \rightarrow \epsilon$   
[4]  $E_1 \rightarrow E_2 E_1'$   
[5]  $E_1' \rightarrow OP_2 E_2 E_1'$   
[6]  $E_1' \rightarrow \epsilon$   
**[7]  $E_2 \rightarrow E_3 OP_3 E_2$**   
**[8]  $E_2 \rightarrow E_3$**   
[9]  $E_3 \rightarrow NUM$   
[10]  $E_3 \rightarrow '(' E ')'$   
[11]  $NUM \rightarrow \langle UNUM \rangle$   
[12]  $NUM \rightarrow '-' \langle UNUM \rangle$   
[13]  $OP_1 \rightarrow '+'$   
[14]  $OP_1 \rightarrow '-'$   
[15]  $OP_2 \rightarrow '*'$   
[16]  $OP_2 \rightarrow '/'$   
[17]  $OP_3 \rightarrow '^'$

消除回溯



[1]  $E \rightarrow E_1 E'$   
[2]  $E' \rightarrow OP_1 E_1 E'$   
[3]  $E' \rightarrow \epsilon$   
[4]  $E_1 \rightarrow E_2 E_1'$   
[5]  $E_1' \rightarrow OP_2 E_2 E_1'$   
[6]  $E_1' \rightarrow \epsilon$   
**[7]  $E_2 \rightarrow E_3 E_2'$**   
**[8]  $E_2' \rightarrow OP_3 E_2$**   
**[9]  $E_2' \rightarrow \epsilon$**   
[10]  $E_3 \rightarrow NUM$   
[11]  $E_3 \rightarrow '(' E ')'$   
[12]  $NUM \rightarrow \langle UNUM \rangle$   
[13]  $NUM \rightarrow '-' \langle UNUM \rangle$   
[14]  $OP_1 \rightarrow '+'$   
[15]  $OP_1 \rightarrow '-'$   
[16]  $OP_2 \rightarrow '*'$   
[17]  $OP_2 \rightarrow '/'$   
[18]  $OP_3 \rightarrow '^'$

# First集合计算

- 对于产生式  $X \rightarrow \beta_1\beta_2 \dots \beta_n$  来说:
  - 如果  $\epsilon \notin First(\beta_1)$ , 则  $First(X) = First(\beta_1)$
  - 如果  $\epsilon \in First(\beta_1) \& \dots \& \epsilon \in First(\beta_i)$ , 则  $First(X) = First(\beta_1) \cup \dots \cup First(\beta_{i+1})$

[1]  $E \rightarrow E_1 E'$   
 [2]  $E' \rightarrow OP_1 E_1 E'$   
 [3]  $E' \rightarrow \epsilon$   
 [4]  $E_1 \rightarrow E_2 E_1'$   
 [5]  $E_1' \rightarrow OP_2 E_2 E_1'$   
 [6]  $E_1' \rightarrow \epsilon$   
 [7]  $E_2 \rightarrow E_3 E_2'$   
 [8]  $E_2' \rightarrow OP_3 E_2$   
 [9]  $E_2' \rightarrow \epsilon$   
 [10]  $E_3 \rightarrow NUM$   
 [11]  $E_3 \rightarrow '(' E ')'$   
 [12]  $NUM \rightarrow <UNUM>$   
 [13]  $NUM \rightarrow '-' <UNUM>$   
 [14]  $OP_1 \rightarrow '+'$   
 [15]  $OP_1 \rightarrow '-'$   
 [16]  $OP_2 \rightarrow '*'$   
 [17]  $OP_2 \rightarrow '/'$   
 [18]  $OP_3 \rightarrow '^'$

	<UNUM>	'+'	'-'	'*'	'/'	'^'	'('	')'	$\epsilon$
E	[1]		[1]				[1]		
E'		[2]	[2]						[3]
E1	[4]		[4]				[4]		
E1'				[5]	[5]				[6]
E2	[7]		[7]				[7]		
E2'						[8]			[9]
E3	[10]		[10]				[11]		
NUM	[12]		[13]						
OP1		[14]	[15]						
OP2				[16]	[17]				
OP3						[18]			

# Follow集合计算

- 如果存在规则  $X \rightarrow \epsilon$ ，选择规则时需要考虑  $X$  之后紧跟的字符

$$First^+(X \rightarrow \beta) = \begin{cases} First(\beta), & \text{if } \epsilon \notin First(\beta) \\ First(\beta) \cup Follow(X), & \text{otherwise} \end{cases}$$

[1]  $E \rightarrow E1 E'$   
 [2]  $E' \rightarrow OP1 E1 E'$   
 [3]  $E' \rightarrow \epsilon$   
 [4]  $E1 \rightarrow E2 E1'$   
 [5]  $E1' \rightarrow OP2 E2 E1'$   
 [6]  $E1' \rightarrow \epsilon$   
 [7]  $E2 \rightarrow E3 E2'$   
 [8]  $E2' \rightarrow OP3 E2$   
 [9]  $E2' \rightarrow \epsilon$   
 [10]  $E3 \rightarrow NUM$   
 [11]  $E3 \rightarrow '(' E ')'$   
 [12]  $NUM \rightarrow <UNUM>$   
 [13]  $NUM \rightarrow '-' <UNUM>$   
 [14]  $OP1 \rightarrow '+'$   
 [15]  $OP1 \rightarrow '-'$   
 [16]  $OP2 \rightarrow '*'$   
 [17]  $OP2 \rightarrow '/'$   
 [18]  $OP3 \rightarrow '^'$

	<UNUM>	'+'	'-'	'*'	'/'	'^'	'('	')'	$\epsilon$
E	[1]		[1]				[1]		
E'		[2]	[2]					[3]	[3]
E1	[4]		[4]						
E1'		[6]	[6]	[5]	[5]			[6]	[6]
E2	$Follow(E1') = Follow(E1) = First^+(E')$							[7]	
E2'		[9]	[9]	[9]	[9]	[8]		[9]	[9]
E3	$Follow(E2') = Follow(E2) = First^+(E1')$						[1]		
NUM	[12]		[13]						
OP1		[14]	[15]						
OP2				[16]	[17]				
OP3						[18]			

# 基于First+集合得到解析表

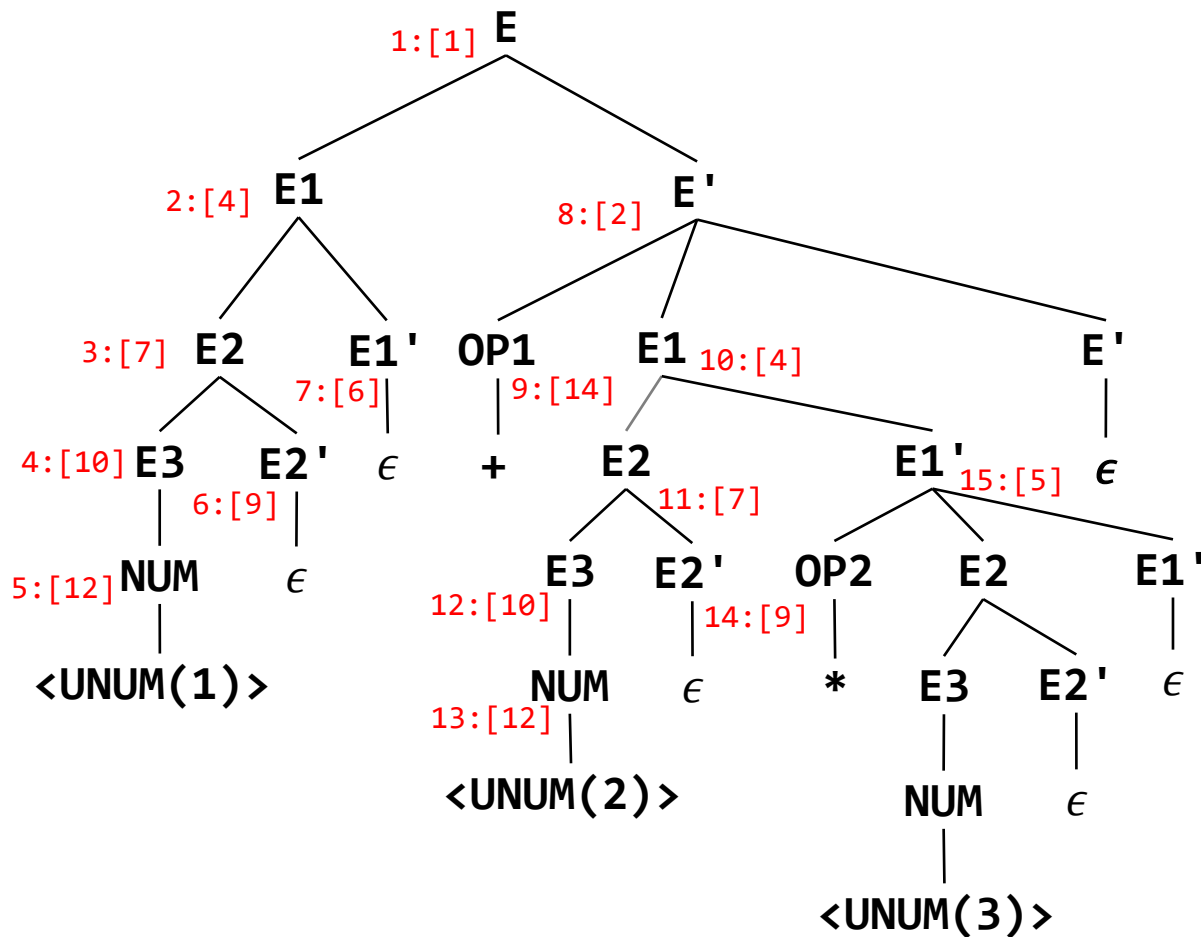
- 解析表每一个单元格最多只有一条可选规则
- 无回溯语法特性:  $\forall 1 \leq i, j \leq n, First^+(X \rightarrow \beta_i) \cap First^+(X \rightarrow \beta_j) = \emptyset$

	<UNUM>	'+'	'-'	'*'	'/'	'^'	'('	')'
E	[1]		[1]				[1]	
E'		[2]	[2]					[3]
E1	[4]		[4]				[4]	
E1'		[6]	[6]	[5]	[5]			[6]
E2	[7]		[7]				[7]	
E2'		[9]	[9]	[9]	[9]	[8]		[9]
E3	[10]		[10]				[11]	
NUM	[12]		[13]					
OP1		[14]	[15]					
OP2				[16]	[17]			
OP3						[18]		

# LL(1)解析表应用示例

- [1]  $E \rightarrow E_1 E'$
- [2]  $E' \rightarrow OP_1 E_1 E'$
- [3]  $E' \rightarrow \epsilon$
- [4]  $E_1 \rightarrow E_2 E_1'$
- [5]  $E_1' \rightarrow OP_2 E_2 E_1'$
- [6]  $E_1' \rightarrow \epsilon$
- [7]  $E_2 \rightarrow E_3 E_2'$
- [8]  $E_2' \rightarrow OP_3 E_2$
- [9]  $E_2' \rightarrow \epsilon$
- [10]  $E_3 \rightarrow NUM$
- [11]  $E_3 \rightarrow '(' E ')'$
- [12]  $NUM \rightarrow \langle UNUM \rangle$
- [13]  $NUM \rightarrow '-' \langle UNUM \rangle$
- [14]  $OP_1 \rightarrow '+'$
- [15]  $OP_1 \rightarrow '-'$
- [16]  $OP_2 \rightarrow '*'$
- [17]  $OP_2 \rightarrow '/'$
- [18]  $OP_3 \rightarrow '^'$

词元流:  $\langle UNUM(1) \rangle \langle ADD \rangle \langle UNUM(2) \rangle \langle MUL \rangle \langle UNUM(3) \rangle$



语法解析树

# LL(1)算法复杂度分析

- $O(\text{句子长度} * \text{规则个数})?$
- $O(\text{句子长度} * \text{解析表行数})?$
- 其它?

# 练习

- 将正则表达式语法规则改写为LL(1)语法并写出解析表

[1]	REGEX	→	REGEX '   ' CONCAT
[2]	REGEX	→	CONCAT
[3]	CONCAT	→	CONCAT CLOSURE
[4]	CONCAT	→	CLOSURE
[5]	CLOSURE	→	CLOSURE '*'
[6]	CLOSURE	→	ITEM
[7]	ITEM	→	' ( ' REGEX ' ) '
[8]	ITEM	→	<CHAR>

## 三、PEG文法解析

---

# PEG文法

- 不允许左递归
- 允许回溯：通过/（顺序选择）消除二义性

```
[1] E → E OP1 E1
[2] E → E1
[3] E1 → E1 OP2 E2
[4] E1 → E2
[5] E2 → E3 OP3 E2
[6] E2 → E3
[7] E3 → NUM
[8] E3 → '(' E ')'
```

```
[9] NUM → <UNUM>
[10] NUM → '-' <UNUM>
[11] OP1 → '+'
[12] OP1 → '-'
[13] OP2 → '*'
[14] OP2 → '/'
[15] OP3 → '^'
```

消除左递归



设置选择顺序

```
E → E1 E'
E' → OP1 E1 E' / ε
E1 → E2 E1'
E1' → OP2 E2 E1' / ε
E2 → E3 OP3 E2 / E3
E3 → NUM / '(' E ')'
```

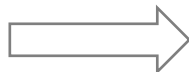
```
NUM → <UNUM> / '-' <UNUM>
OP1 → '+' / '-'
OP2 → '*' / '/'
OP3 → '^'
```

PEG文法

# PEG语法优化

```
E → E1 E'  
E' → OP1 E1 E' / ε  
E1 → E2 E1'  
E1' → OP2 E2 E1' / ε  
E2 → E3 OP3 E2 / E3  
E3 → NUM / '(' E ')'  
NUM → <UNUM> / '-' <UNUM>  
OP1 → '+' / '-'  
OP2 → '*' / '/'  
OP3 → '^'
```

优化



```
E → E1 (OP1 E1)*  
E1 → E2 (OP2 E2)*  
E2 → E3 OP3 E2 / E3  
E3 → NUM / '(' E ')'  
NUM → <UNUM> / '-'  
<UNUM>  
OP1 → '+' / '-'  
OP2 → '*' / '/'  
OP3 → '^'
```



# 解析

$\langle \text{UNUM}(1) \rangle \circ '+' \langle \text{UNUM}(2) \rangle '*' \langle \text{UNUM}(3) \rangle$

$E \rightarrow E1 E'$

$E' \rightarrow OP1 E1 E' / \epsilon$

$E1 \rightarrow E2 E1'$

$E1' \rightarrow OP2 E2 E1' / \epsilon$

**$E2 \rightarrow E3 OP3 E2 / E3$**

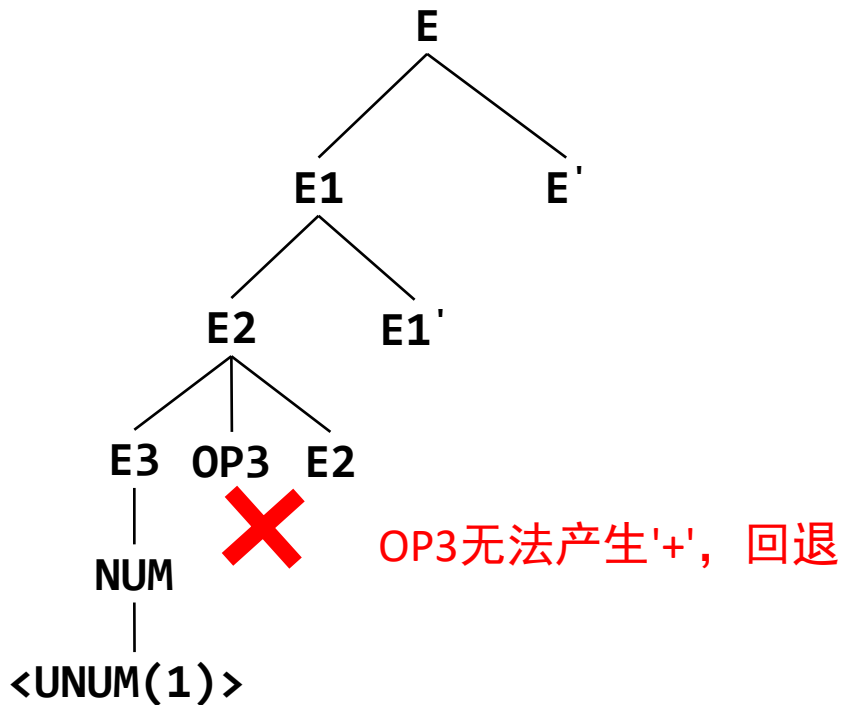
$E3 \rightarrow \text{NUM} / '(' E ')'$

$\text{NUM} \rightarrow \langle \text{UNUM} \rangle / '-' \langle \text{UNUM} \rangle$

$OP1 \rightarrow '+' / '-'$

$OP2 \rightarrow '*' / '/'$

$OP3 \rightarrow '^'$



# 解析

$\langle \text{UNUM}(1) \rangle \circ '+' \langle \text{UNUM}(2) \rangle '*' \langle \text{UNUM}(3) \rangle$

$E \rightarrow E1 E'$

$E' \rightarrow OP1 E1 E' / \epsilon$

$E1 \rightarrow E2 E1'$

**$E1' \rightarrow OP2 E2 E1' / \epsilon$**

$E2 \rightarrow E3 OP3 E2 / E3$

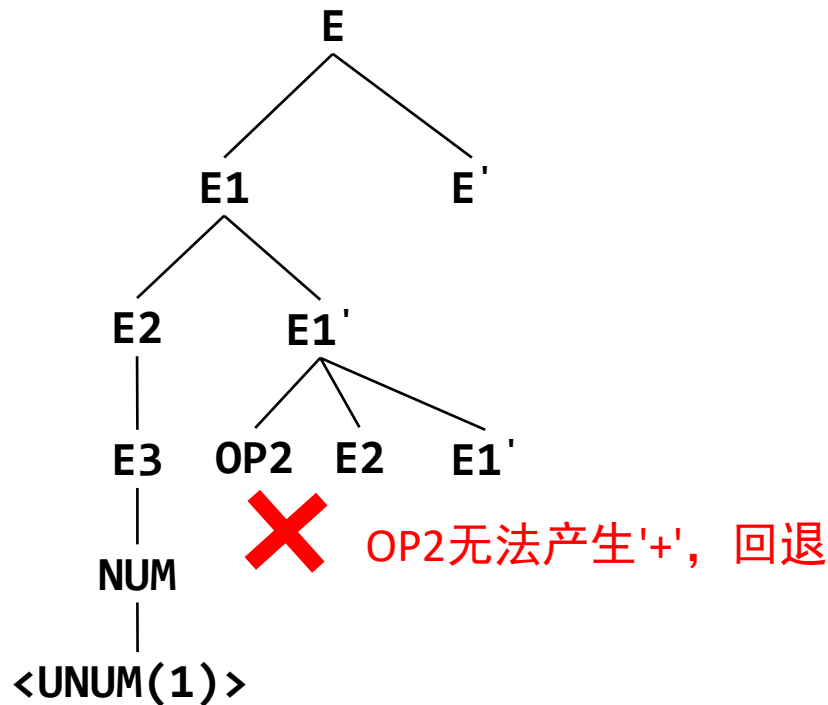
$E3 \rightarrow \text{NUM} / '(' E ')'$

$\text{NUM} \rightarrow \langle \text{UNUM} \rangle / '-' \langle \text{UNUM} \rangle$

$OP1 \rightarrow '+' / '-'$

$OP2 \rightarrow '*' / '/'$

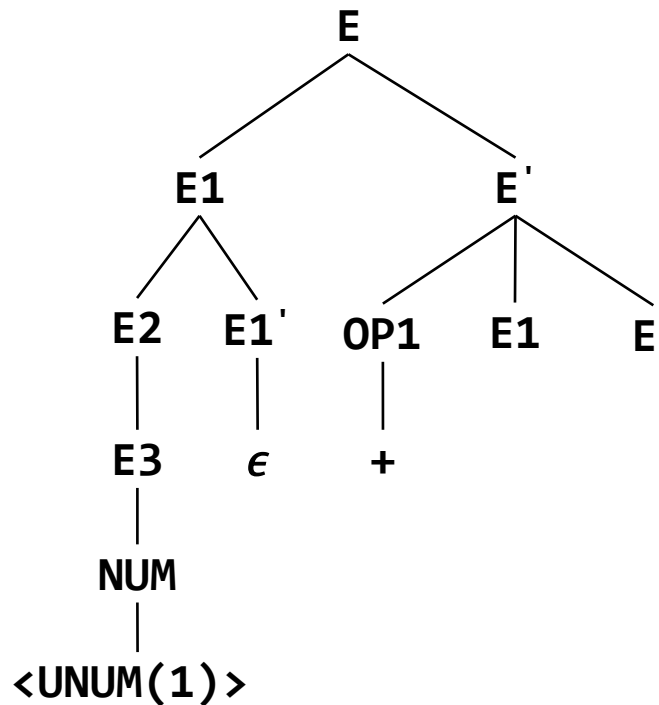
$OP3 \rightarrow '^'$



# 解析

$\langle \text{UNUM}(1) \rangle + \circ \langle \text{UNUM}(2) \rangle * \langle \text{UNUM}(3) \rangle$

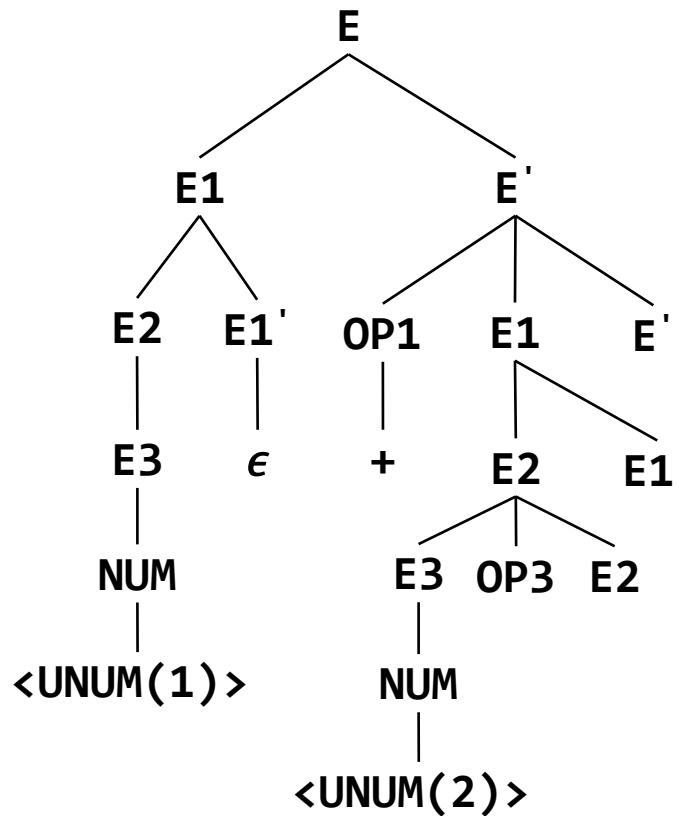
$E \rightarrow E1 E'$   
 $E' \rightarrow OP1 E1 E' / \epsilon$   
 $E1 \rightarrow E2 E1'$   
 $E1' \rightarrow OP2 E2 E1' / \epsilon$   
 $E2 \rightarrow E3 OP3 E2 / E3$   
 $E3 \rightarrow \text{NUM} / '(' E ')'$   
 $\text{NUM} \rightarrow \langle \text{UNUM} \rangle / '-' \langle \text{UNUM} \rangle$   
 $OP1 \rightarrow '+' / '-'$   
 $OP2 \rightarrow '*' / '/'$   
 $OP3 \rightarrow '^'$



# 解析

$\langle \text{UNUM}(1) \rangle \text{'+'} \circ \langle \text{UNUM}(2) \rangle \text{'*'} \langle \text{UNUM}(3) \rangle$

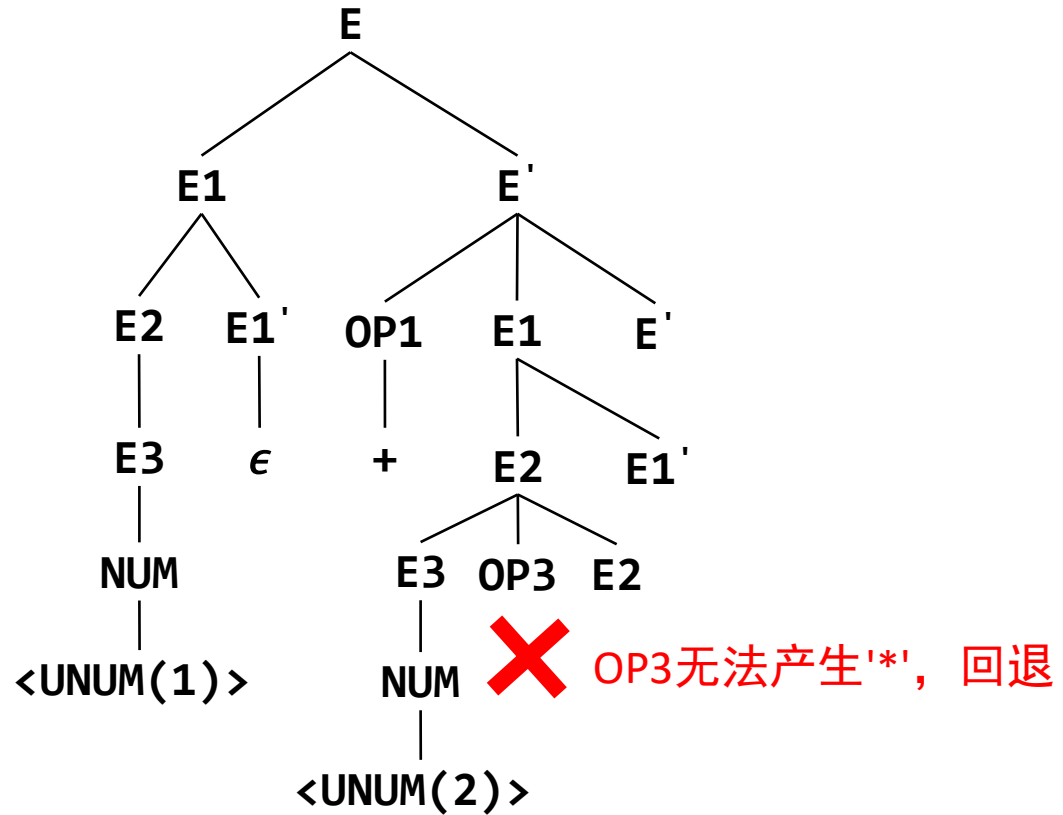
$E \rightarrow E1 E'$   
 $E' \rightarrow OP1 E1 E' / \epsilon$   
 $E1 \rightarrow E2 E1'$   
 $E1' \rightarrow OP2 E2 E1' / \epsilon$   
 $E2 \rightarrow E3 OP3 E2 / E3$   
 $E3 \rightarrow \text{NUM} / \text{'(' } E \text{' )'}$   
 $\text{NUM} \rightarrow \langle \text{UNUM} \rangle / \text{'-' } \langle \text{UNUM} \rangle$   
 $OP1 \rightarrow \text{'+'} / \text{'-'}$   
 $OP2 \rightarrow \text{'*'} / \text{'/'}$   
 $OP3 \rightarrow \text{'^'}$



# 解析

$\langle \text{UNUM}(1) \rangle \text{'+'} \langle \text{UNUM}(2) \rangle \circ \text{'*'} \langle \text{UNUM}(3) \rangle$

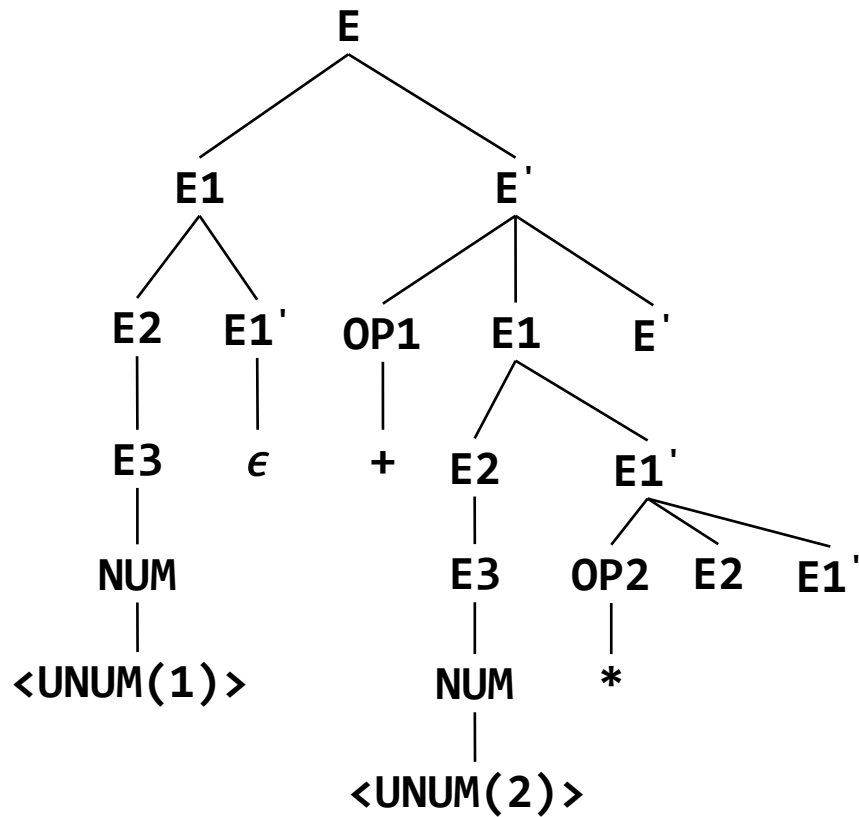
$E \rightarrow E1 E'$   
 $E' \rightarrow OP1 E1 E' / \epsilon$   
 $E1 \rightarrow E2 E1'$   
 $E1' \rightarrow OP2 E2 E1' / \epsilon$   
 **$E2 \rightarrow E3 OP3 E2 / E3$**   
 $E3 \rightarrow \text{NUM} / \text{'('} E \text{'})'$   
 $\text{NUM} \rightarrow \langle \text{UNUM} \rangle / \text{'-'}$   $\langle \text{UNUM} \rangle$   
 $OP1 \rightarrow \text{'+'} / \text{'-'}$   
 $OP2 \rightarrow \text{'*'} / \text{'/'}$   
 $OP3 \rightarrow \text{'^'}$



# 解析

$\langle \text{UNUM}(1) \rangle + \langle \text{UNUM}(2) \rangle * \langle \text{UNUM}(3) \rangle$

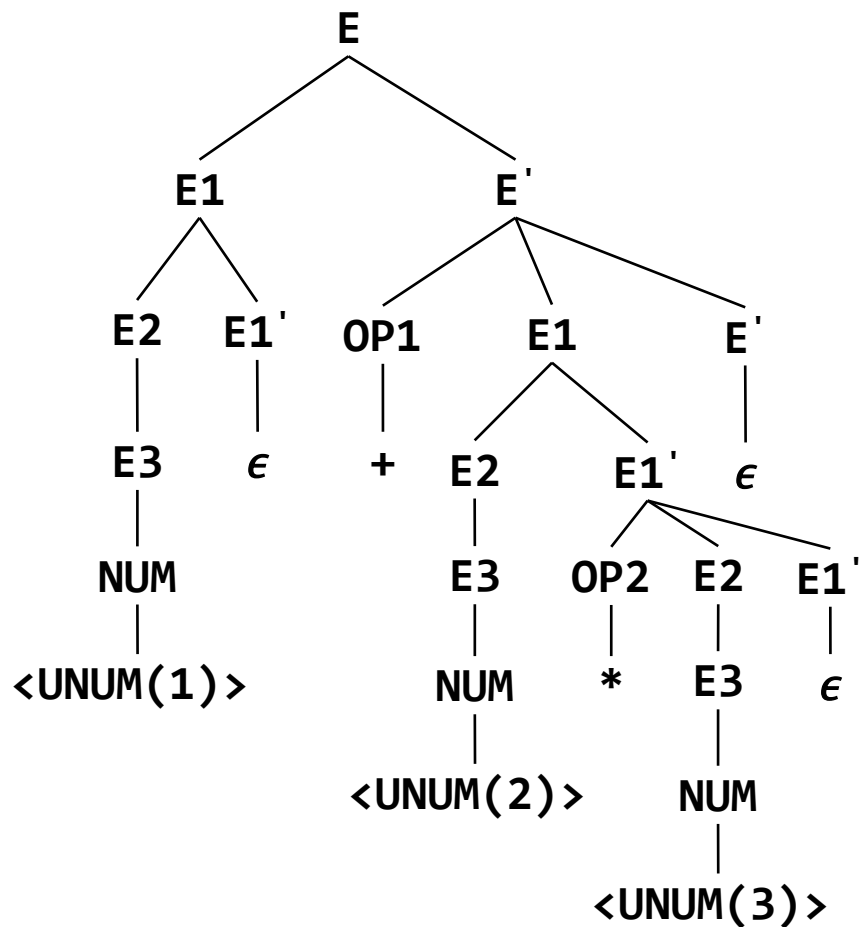
$E \rightarrow E1 E'$   
 $E' \rightarrow OP1 E1 E' / \epsilon$   
 $E1 \rightarrow E2 E1'$   
 $E1' \rightarrow OP2 E2 E1' / \epsilon$   
 $E2 \rightarrow E3 OP3 E2 / E3$   
 $E3 \rightarrow \text{NUM} / '(' E ')'$   
 $\text{NUM} \rightarrow \langle \text{UNUM} \rangle / '-' \langle \text{UNUM} \rangle$   
 $OP1 \rightarrow '+' / '-'$   
 $OP2 \rightarrow '*' / '/'$   
 $OP3 \rightarrow '^'$



# 解析（省略后续迭代和回溯）

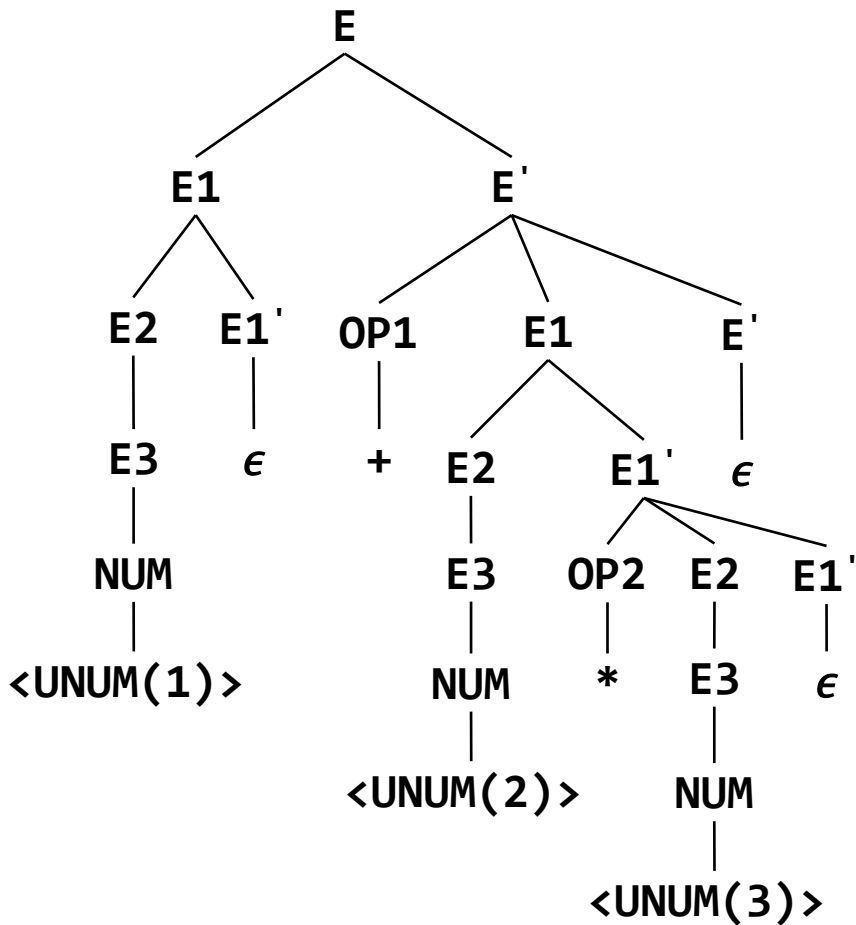
$\langle \text{UNUM}(1) \rangle$  '+'  $\langle \text{UNUM}(2) \rangle$  '\*'  $\langle \text{UNUM}(3) \rangle$  ◦

$E \rightarrow E1 E'$   
 $E' \rightarrow OP1 E1 E' / \epsilon$   
 $E1 \rightarrow E2 E1'$   
 $E1' \rightarrow OP2 E2 E1' / \epsilon$   
 $E2 \rightarrow E3 OP3 E2 / E3$   
 $E3 \rightarrow \text{NUM} / '(' E ')'$   
 $\text{NUM} \rightarrow \langle \text{UNUM} \rangle / '-' \langle \text{UNUM} \rangle$   
 $OP1 \rightarrow '+' / '-'$   
 $OP2 \rightarrow '*' / '/'$   
 $OP3 \rightarrow '^'$



# Packrat Parser

- Packrat Table: 以“输入位置 + 非终结符”为键缓存解析结果
- 利用查表避免重复回溯，提高解析效率

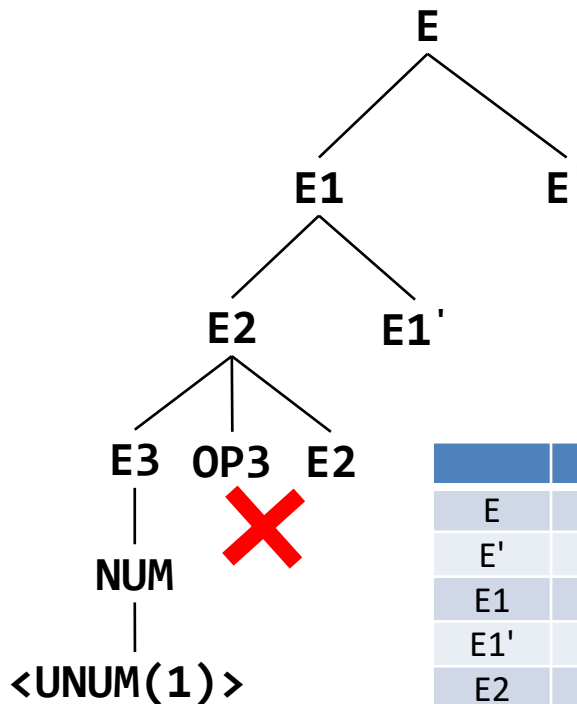


	1	2	3	4	5
E	5				
E'		4			
E1	1		3		
E1'				2	
E2	1		1		1
E3	1		1		1
NUM	1		1		1
OP1		1			
OP2				1	
OP3					
	1	+	2	*	3

# 优化效果举例：解析回退场景举例

$\langle \text{UNUM}(1) \rangle \circ '+' \langle \text{UNUM}(2) \rangle '*' \langle \text{UNUM}(3) \rangle$

$E \rightarrow E1 E'$   
 $E' \rightarrow OP1 E1 E' / \epsilon$   
 $E1 \rightarrow E2 E1'$   
 $E1' \rightarrow OP2 E2 E1' / \epsilon$   
 $E2 \rightarrow E3 OP3 E2 / E3$   
 $E3 \rightarrow \text{NUM} / '(' E ')'$   
 $\text{NUM} \rightarrow \langle \text{UNUM} \rangle / '-' \langle \text{UNUM} \rangle$   
 $OP1 \rightarrow '+' / '-'$   
 $OP2 \rightarrow '*' / '/'$   
 $OP3 \rightarrow '^'$



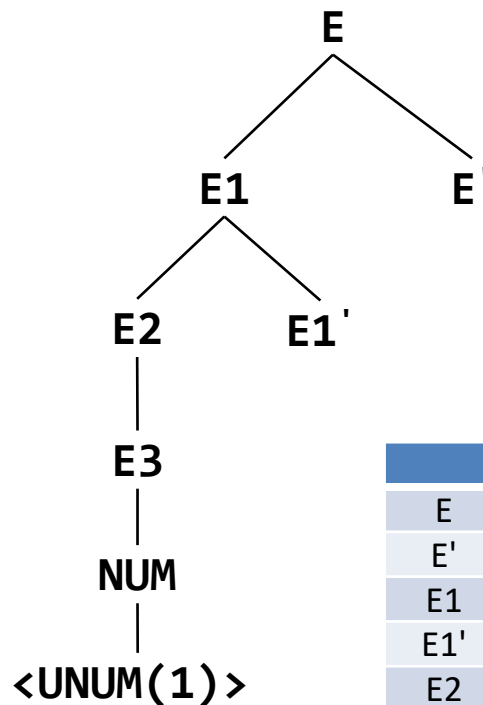
已知E3可以产生 $\langle \text{UNUM}(1) \rangle$

	1	2	3	4	5
E					
E'					
E1					
E1'					
E2					
E3	1				
NUM	1				
OP1					
OP2					
OP3					
	1	+	2	*	3

# 优化效果举例：解析回退场景

$\langle \text{UNUM}(1) \rangle \circ '+' \langle \text{UNUM}(2) \rangle '*' \langle \text{UNUM}(3) \rangle$

$E \rightarrow E1 E'$   
 $E' \rightarrow OP1 E1 E' / \epsilon$   
 $E1 \rightarrow E2 E1'$   
 $E1' \rightarrow OP2 E2 E1' / \epsilon$   
 $E2 \rightarrow E3 OP3 E2 / E3$   
 $E3 \rightarrow \text{NUM} / '(' E ')'$   
 $\text{NUM} \rightarrow \langle \text{UNUM} \rangle / '-' \langle \text{UNUM} \rangle$   
 $OP1 \rightarrow '+' / '-'$   
 $OP2 \rightarrow '*' / '/'$   
 $OP3 \rightarrow '^'$



	1	2	3	4	5
E					
E'					
E1					
E1'					
E2	1				
E3	1				
NUM	1				
OP1					
OP2					
OP3					
	1	+	2	*	3

再次展开E3时复用已有结果

## 四、Earley算法

---

# 解析状态分析

- 规则  $X \rightarrow \beta$  对应  $|\beta| + 1$  种解析状态（规范项）

```
[1] E → E OP1 E1
[2] E → E1
[3] E1 → E1 OP2 E2
[4] E1 → E2
[5] E2 → E3 OP3 E2
[6] E2 → E3
[7] E3 → NUM
[8] E3 → '(' E ') '
[9] NUM → <UNUM>
[10] NUM → '-' <UNUM>
[11] OP1 → '+'
[12] OP1 → '-'
[13] OP2 → '*'
[14] OP2 → '/'
[15] OP3 → '^'
```

句柄分析



```
[1] E → ◦ E OP1 E1
[1] E → E ◦ OP1 E1
[1] E → E OP1 ◦ E1
[1] E → E OP1 E1 ◦
[2] E → ◦ E1
[2] E → E1 ◦
[3] E1 → ◦ E1 OP2 E2
[3] E1 → E1 ◦ OP2 E2
[3] E1 → E1 OP2 ◦ E2
[3] E1 → E1 OP2 E2 ◦
...
```

规范项

# Earley算法：通用自顶向下语法分析算法

- 支持递归
- 三种基本操作：
  - **预测**：对于每个规范项 $X \rightarrow \alpha \circ Y \beta$ ，根据语法规则展开 $Y \rightarrow \circ \gamma$
  - **扫描**：如果下一个待解析终结符是 $a$ ，且存在规范项 $X \rightarrow \alpha \circ a \beta$ ，则移进 $a$ ，并将规范项更新为 $X \rightarrow \alpha a \circ \beta$
  - **完成**： $Y \rightarrow \gamma \circ$ 表示完成了产生了符号 $Y$ ，更新 $X \rightarrow \alpha \circ Y \beta$ 为 $X \rightarrow \alpha Y \circ \beta$

# 解析过程演示

[1]	$E \rightarrow E \text{ OP1 } E1$
[2]	$E \rightarrow E1$
[3]	$E1 \rightarrow E1 \text{ OP2 } E2$
[4]	$E1 \rightarrow E2$
[5]	$E2 \rightarrow E3 \text{ OP3 } E2$
[6]	$E2 \rightarrow E3$
[7]	$E3 \rightarrow \text{NUM}$
[8]	$E3 \rightarrow \text{'(' E ')}$
[9]	$\text{NUM} \rightarrow \langle \text{UNUM} \rangle$
[10]	$\text{NUM} \rightarrow \text{'-'} \langle \text{UNUM} \rangle$
[11]	$\text{OP1} \rightarrow \text{'+'}$
[12]	$\text{OP1} \rightarrow \text{'-'}$
[13]	$\text{OP2} \rightarrow \text{'*'}$
[14]	$\text{OP2} \rightarrow \text{'/'}$
[15]	$\text{OP3} \rightarrow \text{'^'}$

序号	操作	规范项	起源
$s[0]: \circ \langle \text{UNUM} \rangle \text{'+'} \langle \text{UNUM} \rangle \text{'*'}$ $\langle \text{UNUM} \rangle$			
1	起始状态	$E \rightarrow \circ E \text{ OP1 } E1$	$s[0]$
2	起始状态	$E \rightarrow \circ E1$	$s[0]$
3	预测2	$E1 \rightarrow \circ E1 \text{ OP2 } E2$	$s[0]$
4	预测2	$E1 \rightarrow \circ E2$	$s[0]$
5	预测4	$E2 \rightarrow \circ E3 \text{ OP3 } E2$	$s[0]$
6	预测4	$E2 \rightarrow \circ E3$	$s[0]$
7	预测5	$E3 \rightarrow \circ \text{NUM}$	$s[0]$
8	预测5	$E3 \rightarrow \circ \text{'(' E ')}$	$s[0]$
9	预测7	$\text{NUM} \rightarrow \circ \langle \text{UNUM} \rangle$	$s[0]$
10	预测7	$\text{NUM} \rightarrow \circ \text{'-'} \langle \text{UNUM} \rangle$	$s[0]$
$s[1]: \langle \text{UNUM} \rangle \circ \text{'+'} \langle \text{UNUM} \rangle \text{'*'}$ $\langle \text{UNUM} \rangle$			
1	基于 $s[0]$ -9扫描	$\text{NUM} \rightarrow \langle \text{UNUM} \rangle \circ$	$s[0]$

序号	操作	规范项	起源
<b>s[0]:</b> ◦ <UNUM> '+' <UNUM> '*' <UNUM>			
1	起始状态	E → ◦ E OP1 E1	s[0]
2	起始状态	E → ◦ E1	s[0]
3	预测2	E1 → ◦ E1 OP2 E2	s[0]
4	预测2	E1 → ◦ E2	s[0]
5	预测4	E2 → ◦ E3 OP3 E2	s[0]
6	预测4	E2 → ◦ E3	s[0]
7	预测5	E3 → ◦ NUM	s[0]
8	预测5	E3 → ◦ '(' E ')'	s[0]
<b>9</b>	<b>预测7</b>	<b>NUM → ◦ &lt;UNUM&gt;</b>	<b>s[0]</b>
10	预测7	NUM → ◦ '-' <UNUM>	s[0]
<b>s[1]:</b> <UNUM> ◦ '+' <UNUM> '*' <UNUM>			
1	扫描s[0]-9	NUM → <UNUM> ◦	s[0]
2	基于1更新s[0]-7	E3 → NUM ◦	s[0]
3	基于2更新s[0]-5	E2 → E3 ◦ OP3 E2	s[0]
4	基于2更新s[0]-6	E2 → E3 ◦	s[0]
5	基于4更新s[0]-4	E1 → E2 ◦	s[0]
6	基于5更新s[0]-2	E → E1 ◦	s[0]
7	基于5更新s[0]-3	E1 → E1 ◦ OP2 E2	s[0]
8	基于6更新s[0]-1	E → E ◦ OP1 E1	s[0]

序号	操作	规范项	起源
<b>s[1]: &lt;UNUM&gt; ◦ '+' &lt;UNUM&gt; '*' &lt;UNUM&gt;</b>			
1	扫描s[0]-9	NUM → <UNUM> ◦	s[0]
2	基于1更新s[0]-7	E3 → NUM ◦	s[0]
3	基于2更新s[0]-5	E2 → E3 ◦ OP3 E2	s[0]
4	基于2更新s[0]-6	E2 → E3 ◦	s[0]
5	基于4更新s[0]-4	E1 → E2 ◦	s[0]
6	基于5更新s[0]-2	E → E1 ◦	s[0]
7	基于5更新s[0]-3	E1 → E1 ◦ OP2 E2	s[0]
8	基于6更新s[0]-1	E → E ◦ OP1 E1	s[0]
9	预测3	OP3 → ◦ '^'	s[1]
10	预测7	OP2 → ◦ '*'	s[1]
11	预测7	OP2 → ◦ '/'	s[1]
<b>12</b>	<b>预测8</b>	<b>OP1 → ◦ '+'</b>	<b>s[1]</b>
13	预测8	OP1 → ◦ '-'	s[1]
<b>s[2]: &lt;UNUM&gt; '+' ◦ &lt;UNUM&gt; '*' &lt;UNUM&gt;</b>			
1	扫描s[1]-12	OP1 → '+' ◦	s[1]
2			
3			

序号	操作	规范项	起源
s[2]: <UNUM> '+' ◦ <UNUM> '*' <UNUM>			
1	扫描s[1]-12	OP1 → '+' ◦	s[1]
2	基于1更新s[1]-8	E → E OP1 ◦ E1	s[0]
3	预测2	E1 → ◦ E1 OP2 E2	s[2]
4	预测2	E1 → ◦ E2	s[2]
5	预测4	E2 → ◦ E3 OP3 E2	s[2]
6	预测4	E2 → ◦ E3	s[2]
7	预测5	E3 → ◦ NUM	s[2]
8	预测5	E3 → ◦ '(' E ')'	s[2]
9	预测7	NUM → ◦ <UNUM>	s[2]
10	预测7	NUM → ◦ '-' <UNUM>	s[2]
s[3]: <UNUM> '+' <UNUM> ◦ '*' <UNUM>			
1	扫描s[2]-9	NUM → <UNUM> ◦	s[2]
2	基于1更新s[2]-7	E3 → NUM ◦	s[2]
3	基于2更新s[2]-5	E2 → E3 ◦ OP3 E2	s[2]
4	基于2更新s[2]-6	E2 → E3 ◦	s[2]
5	基于4更新s[2]-4	E1 → E2 ◦	s[2]
6	基于5更新s[2]-2	E → E OP1 E1 ◦	s[0]
7	基于5更新s[2]-3	E1 → E1 ◦ OP2 E2	s[2]
8			

序号	操作	规范项	起源
s[3]: <UNUM> '+' <UNUM> ◦ '*' <UNUM>			
1	扫描s[2]-9	NUM → <UNUM>◦	s[2]
2	基于1更新s[2]-7	E3 → NUM◦	s[2]
3	基于2更新s[2]-5	E2 → E3 ◦ OP3 E2	s[2]
4	基于2更新s[2]-6	E2 → E3 ◦	s[2]
5	基于4更新s[2]-4	E1 → E2 ◦	s[2]
6	基于5更新s[2]-2	E → E OP1 E1 ◦	s[0]
7	基于5更新s[2]-3	E1 → E1 ◦ OP2 E2	s[2]
8	预测3	OP3 → ◦ '^'	s[3]
<b>9</b>	<b>预测7</b>	<b>OP2 → ◦ '*'</b>	<b>s[3]</b>
10	预测7	OP2 → ◦ '/'	s[3]
s[4]: <UNUM> '+' <UNUM> '*' ◦ <UNUM>			
1	扫描s[3]-9	OP2 → ◦ '*'	s[3]
2	基于1更新s[3]-7	E1 → E1 OP2 ◦ E2	s[2]
3	预测2	E2 → ◦ E3 OP3 E2	s[4]
4	预测2	E2 → ◦ E3	s[4]
5	预测3	E3 → ◦ NUM	s[4]
6	预测3	E3 → ◦ '(' E ')'	s[4]
<b>7</b>	<b>预测5</b>	<b>NUM → ◦ &lt;UNUM&gt;</b>	<b>s[4]</b>
8	预测5	NUM → ◦ '-' <UNUM>	s[4]

序号	操作	规范项	起源
s[5]: <UNUM> '+' <UNUM> '*' <UNUM> ◦			
1	扫描s[4]-7	NUM → <UNUM> ◦	s[4]
2	基于1更新s[4]-5	E3 → NUM ◦	s[4]
3	基于2更新s[4]-3	E2 → E3 ◦ OP3 E2	s[4]
4	基于2更新s[4]-4	E2 → E3 ◦	s[4]
5	基于4更新s[4]-2	E1 → E1 OP2 E2 ◦	s[2]
6	基于5更新s[2]-2	E → E OP1 E1 ◦	s[0]

# 练习

- 使用Earley算法解析： $1*(2+-3)$