

COMP130014.02 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、Earley算法
- ❖ 三、LL(1)文法和解析

# 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
- 从CFG的开始符号应用规则逐步展开其中的非终结符
  - 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

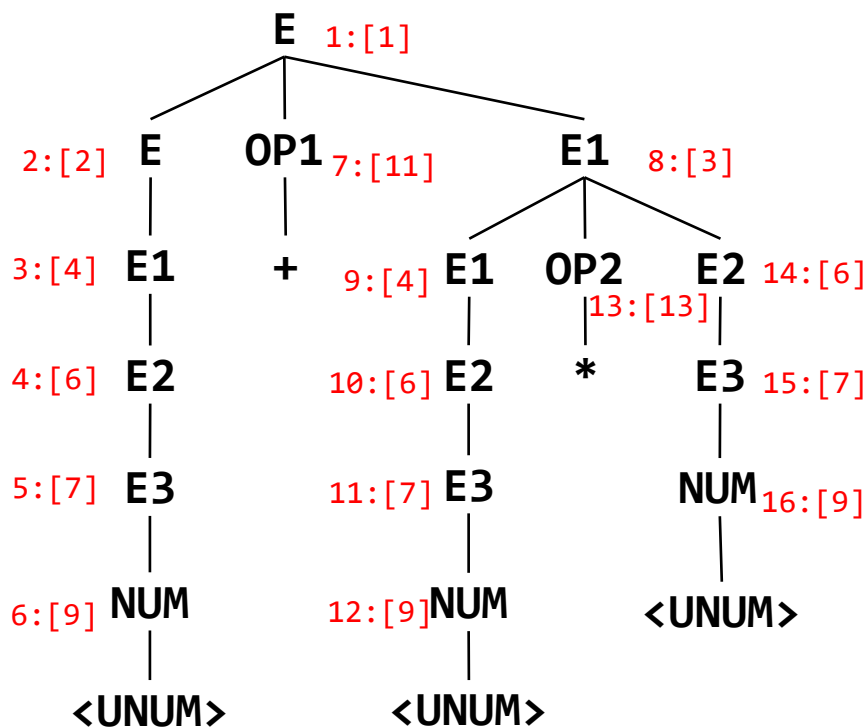
# 自顶向下解析示例

语法规则：

```
[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 → '^'
```

解析对象： 1+2\*3

标签流： <UNUM><ADD><UNUM><MUL><UNUM>



语法解析树

# 如何自动生成语法推导树？

- 如何判断当前应采用哪条规则展开？ 暴力搜索？
- 预测解析： 找出可以生成目标终结符的规则
- 可能存在多种展开方式以及递归问题

$s[0] = \circ <UNUM> '+' <UNUM> '*' <UNUM>$

↑  
当前位置

$E \rightarrow \circ E1$	
$E1 \rightarrow \circ E2$	
$E2 \rightarrow \circ E3 \text{ OP3 } E2$	$E2 \rightarrow \circ E3$
$E3 \rightarrow \circ \text{NUM}$	
$\text{NUM} \rightarrow \circ <UNUM>$	

$E \rightarrow \circ E \text{ OP1 } E1$	
$E \rightarrow \circ E1$	
$E1 \rightarrow \circ E2$	
$E2 \rightarrow \circ E3 \text{ OP3 } E2$	$E2 \rightarrow \circ E3$
$E3 \rightarrow \circ \text{NUM}$	
$\text{NUM} \rightarrow \circ <UNUM>$	

## 二、Earley算法

---

# 解析状态分析

- 句柄：语法规则中已解析的字符
- 规范项（canonical item）：规则 $X \rightarrow \gamma$ 对应 $|\gamma| + 1$ 种句柄状态

```
[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**：如果下一个终结符是  $a$ ，且存在规范项  $X \rightarrow \alpha \circ a \beta$ ，则将其更新为  $X \rightarrow \alpha a \circ \beta$
  - **完成/更新**：  $Y \rightarrow \gamma \circ$  即完成了对  $Y$  的分析，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]	$E3 \rightarrow '(' E ')'$
[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]	$\text{OP3} \rightarrow '^'$

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

序号	操作	规范项	起源
$s[1] = \langle \text{UNUM} \rangle \circ '+' \langle \text{UNUM} \rangle '*' \langle \text{UNUM} \rangle$			
1	扫描 $s[0][9]$	$\text{NUM} \rightarrow \langle \text{UNUM} \rangle \circ$	$s[0]$
2	基于[1]更新 $s[0][7]$	$E3 \rightarrow \text{NUM} \circ$	$s[0]$
3	基于[2]更新 $s[0][5]$	$E2 \rightarrow E3 \circ \text{OP3 } E2$	$s[0]$
4	基于[2]更新 $s[0][6]$	$E2 \rightarrow E3 \circ$	$s[0]$
5	基于[4]更新 $s[0][4]$	$E1 \rightarrow E2 \circ$	$s[0]$
6	基于[5]更新 $s[0][2]$	$E \rightarrow E1 \circ$	$s[0]$
7	基于[5]更新 $s[0][3]$	$E1 \rightarrow E1 \circ \text{OP2 } E2$	$s[0]$
8	基于[6]更新 $s[0][1]$	$E \rightarrow E \circ \text{OP1 } E1$	$s[0]$
9	预测[3]	$\text{OP3} \rightarrow \circ '^{'$	$s[1]$
10	预测[7]	$\text{OP2} \rightarrow \circ '*'$	$s[1]$
11	预测[7]	$\text{OP2} \rightarrow \circ '/'$	$s[1]$
12	预测[8]	$\text{OP1} \rightarrow \circ '+'$	$s[1]$
13	预测[8]	$\text{OP1} \rightarrow \circ '-'$	$s[1]$
$s[2] = \langle \text{UNUM} \rangle '+' \circ \langle \text{UNUM} \rangle '*' \langle \text{UNUM} \rangle$			
1	扫描 $s[1][12]$	$\text{OP1} \rightarrow '+' \circ$	$s[1]$
2			
3			

序号	操作	规范项	起源
$s[2] = \langle \text{UNUM} \rangle '+' \circ \langle \text{UNUM} \rangle '*' \langle \text{UNUM} \rangle$			
1	扫描 $s[1][12]$	$OP1 \rightarrow '+' \circ$	$s[1]$
2	基于[1]更新 $s[1][8]$	$E \rightarrow E OP1 \circ E1$	$s[0]$
3	预测[2]	$E1 \rightarrow \circ E1 OP2 E2$	$s[2]$
4	预测[2]	$E1 \rightarrow \circ E2$	$s[2]$
5	预测[4]	$E2 \rightarrow \circ E3 OP3 E2$	$s[2]$
6	预测[4]	$E2 \rightarrow \circ E3$	$s[2]$
7	预测[5]	$E3 \rightarrow \circ \text{NUM}$	$s[2]$
8	预测[5]	$E3 \rightarrow \circ '(' E ')'$	$s[2]$
9	预测[7]	$\text{NUM} \rightarrow \circ \langle \text{UNUM} \rangle$	$s[2]$
10	预测[7]	$\text{NUM} \rightarrow \circ '-' \langle \text{UNUM} \rangle$	$s[2]$
$s[3] = \langle \text{UNUM} \rangle '+' \langle \text{UNUM} \rangle \circ '*' \langle \text{UNUM} \rangle$			
1	扫描 $s[2][9]$	$\text{NUM} \rightarrow \langle \text{UNUM} \rangle \circ$	$s[2]$
2	基于[1]更新 $s[2][7]$	$E3 \rightarrow \text{NUM} \circ$	$s[2]$
3	基于[2]更新 $s[2][5]$	$E2 \rightarrow E3 \circ OP3 E2$	$s[2]$
4	基于[2]更新 $s[2][6]$	$E2 \rightarrow E3 \circ$	$s[2]$
5	基于[4]更新 $s[2][4]$	$E1 \rightarrow E2 \circ$	$s[2]$
6	基于[5]更新 $s[2][2]$	$E \rightarrow E OP1 E1 \circ$	$s[0]$
7	基于[5]更新 $s[2][3]$	$E1 \rightarrow E1 \circ OP2 E2$	$s[2]$
8			

序号	操作	规范项	起源
$s[3] = \langle \text{UNUM} \rangle '+' \langle \text{UNUM} \rangle \circ '*' \langle \text{UNUM} \rangle$			
1	扫描 $s[2][9]$	$\text{NUM} \rightarrow \langle \text{UNUM} \rangle \circ$	$s[2]$
2	基于[1]更新 $s[2][7]$	$E3 \rightarrow \text{NUM} \circ$	$s[2]$
3	基于[2]更新 $s[2][5]$	$E2 \rightarrow E3 \circ \text{OP3 } E2$	$s[2]$
4	基于[2]更新 $s[2][6]$	$E2 \rightarrow E3 \circ$	$s[2]$
5	基于[4]更新 $s[2][4]$	$E1 \rightarrow E2 \circ$	$s[2]$
6	基于[5]更新 $s[2][2]$	$E \rightarrow E \text{ OP1 } E1 \circ$	$s[0]$
7	基于[5]更新 $s[2][3]$	$E1 \rightarrow E1 \circ \text{OP2 } E2$	$s[2]$
8	预测[3]	$\text{OP3} \rightarrow \circ '^{'$	$s[3]$
9	预测[7]	<b><math>\text{OP2} \rightarrow \circ '*'</math></b>	<b><math>s[3]</math></b>
10	预测[7]	$\text{OP2} \rightarrow \circ '/'$	$s[3]$
$s[4] = \langle \text{UNUM} \rangle '+' \langle \text{UNUM} \rangle '*' \circ \langle \text{UNUM} \rangle$			
1	扫描 $s[3][9]$	$\text{OP2} \rightarrow \circ '*'$	$s[3]$
2	基于[1]更新 $s[3][7]$	$E1 \rightarrow E1 \text{ OP2 } \circ E2$	$s[2]$
3	预测[2]	$E2 \rightarrow \circ E3 \text{ OP3 } E2$	$s[4]$
4	预测[2]	$E2 \rightarrow \circ E3$	$s[4]$
5	预测[3]	$E3 \rightarrow \circ \text{NUM}$	$s[4]$
6	预测[3]	$E3 \rightarrow \circ '(' E ')'$	$s[4]$
7	预测[5]	<b><math>\text{NUM} \rightarrow \circ \langle \text{UNUM} \rangle</math></b>	<b><math>s[4]</math></b>
8	预测[5]	$\text{NUM} \rightarrow \circ '-' \langle \text{UNUM} \rangle$	$s[4]$

序号	操作	规范项	起源
$s[5] = \langle \text{UNUM} \rangle '+' \langle \text{UNUM} \rangle '*' \langle \text{UNUM} \rangle \circ$			
1	扫描 $s[4][7]$	$\text{NUM} \rightarrow \langle \text{UNUM} \rangle \circ$	$s[4]$
2	基于[1]更新 $s[4][5]$	$E3 \rightarrow \text{NUM} \circ$	$s[4]$
3	基于[2]更新 $s[4][3]$	$E2 \rightarrow E3 \circ \text{OP3 } E2$	$s[4]$
4	基于[2]更新 $s[4][4]$	$E2 \rightarrow E3 \circ$	$s[4]$
5	基于[4]更新 $s[4][2]$	$E1 \rightarrow E1 \text{ OP2 } E2 \circ$	$s[2]$
6	基于[5]更新 $s[2][2]$	$E \rightarrow E \text{ OP1 } E1 \circ$	$s[0]$

# 练习

- 使用Earley算法解析：
  - $1*(2+-3)$



## Earley算法复杂度分析

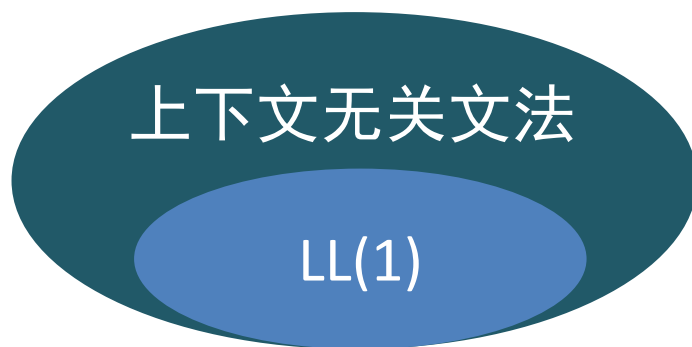
- 1)  $O(\text{句子长度} * \text{规则个数} * \text{规则长度})$
- 2)  $O(\text{句子长度}^2 * \text{规则个数} * \text{规则长度})$
- 3) 其它

### 三、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
- 可能导致搜索算法无限递归，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:
  - 引入新的非终结符 $E'$ , 使其可以为 $\epsilon$
  - 将递归规则之外的产生式右侧置于 $E'$ 之前
  - 为 $E'$ 构造产生式, 使新旧规则等价

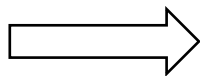
$$\begin{array}{|l} E \rightarrow E \alpha \\ E \rightarrow \beta \end{array} \Rightarrow \begin{array}{|l} E \rightarrow \beta E' \\ E' \rightarrow \alpha E' \\ E' \rightarrow \epsilon \end{array} \quad \{\beta\alpha, \beta\alpha\alpha, \beta\alpha\alpha\alpha, \dots\}$$

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

# 应用

- [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]  $\text{OP3} \rightarrow '^'$

消除左递归



- [1]  $E \rightarrow E1 E'$
- [2]  $E' \rightarrow \text{OP1 } E1 E'$
- [3]  $E' \rightarrow \epsilon$
- [4]  $E1 \rightarrow E2 E1'$
- [5]  $E1' \rightarrow \text{OP2 } E2 E1'$
- [6]  $E1' \rightarrow \epsilon$
- [7]  $E2 \rightarrow E3 \text{ OP3 } E2$
- [8]  $E2 \rightarrow E3$
- [9]  $E3 \rightarrow \text{NUM}$
- [10]  $E3 \rightarrow '(' E ')'$
- [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

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

# 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

$$[1] \quad S \rightarrow \alpha \rightarrow \cdots \rightarrow a\gamma$$

$$[2] \quad S \rightarrow \beta \rightarrow \cdots \rightarrow b\delta$$



# 消除回溯：提取左因子

- 改写语法规则，使新旧规则等价：
  - 对一组生成式提取共同前缀，置于新非终结符 $E'$ 之前
  - 为 $E'$ 编写生成式规则，使新旧规则等价

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

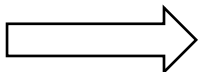


$$\begin{array}{l} E \rightarrow \alpha E' | \gamma_1 | \dots | \gamma_j \\ E' \rightarrow \beta_1 | \beta_2 | \dots | \beta_n \end{array}$$

# 应用

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

消除回溯语法



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

# First集合计算

• 对于生成式  $A \rightarrow \beta_1 \beta_2 \dots \beta_n$  来说:

- 如果  $\epsilon \notin First(\beta_1)$ , 则  $First(A) = First(\beta_1)$
- 如果  $\epsilon \in First(\beta_1) \& \dots \& \epsilon \in First(\beta_i)$ , 则  $First(A) = First(\beta_1) \cup \dots \cup First(\beta_{i+1})$

[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]
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  $A \rightarrow \epsilon$ ，选择规则时需要考虑  $A$  之后紧跟的字符

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

[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]		$Follow(E') = Follow(E)$				
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：

$$\forall 1 \leq i, j \leq n, First^+(A \rightarrow \beta_i) \cap First^+(A \rightarrow \beta_j) = \emptyset$$

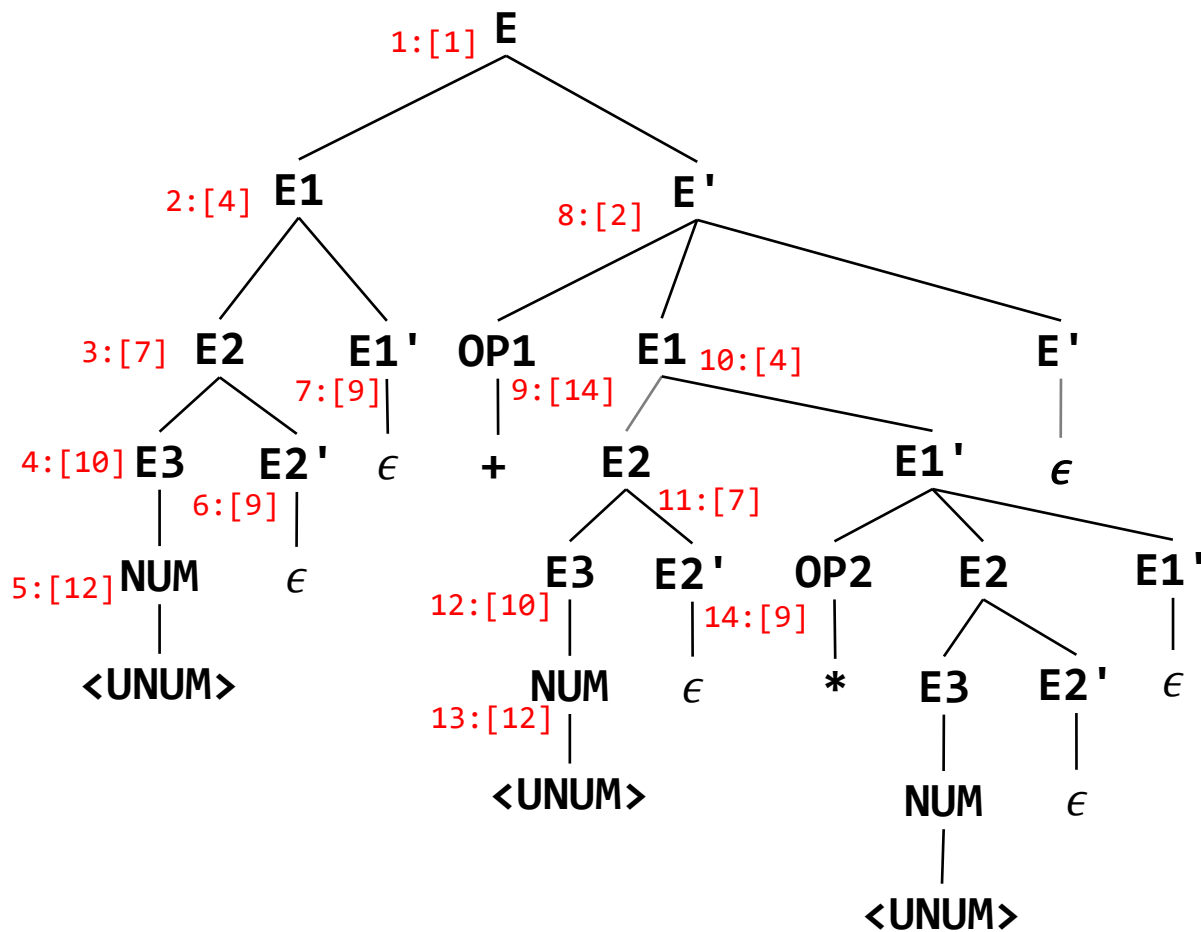
	<UNUM>	'+'	'-'	'*'	'/'	'^'	'('	')'
E	[1]		[1]				[1]	
E'		[2]	[2]					[3]
E1	[4]		[4]				[4]	
E1'		[6]	[6]	[5]	[5]			[6]
E2	[7]		[7]				[7]	
E2'		[9]	[9]	[9]	[9]	[8]		[9]
E3	[10]		[10]				[11]	
NUM	[12]		[13]					
OP1		[14]	[15]					
OP2				[16]	[17]			
OP3						[18]		

# LL(1)解析表应用示例

解析对象： 1+2\*3

标签流： <UNUM><ADD><UNUM><MUL><UNUM>

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



语法解析树

## LL(1)算法复杂度分析

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

# 练习

- 将正则表达式CFG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>