

Lecture 4

LL(1)文法和解析

徐 辉

xuh@fudan.edu.cn



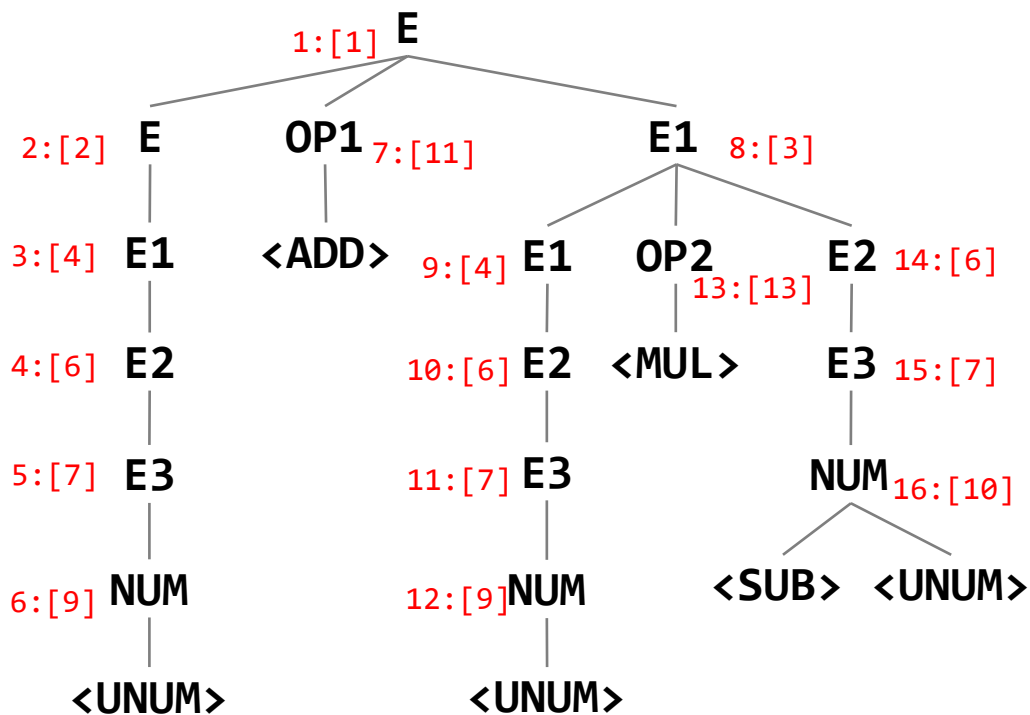
自顶向下解析

语法规则：

- [1] $E \rightarrow E \text{ OP1 } E1$
- [2] | $E1$
- [3] $E1 \rightarrow E1 \text{ OP2 } E2$
- [4] | $E2$
- [5] $E2 \rightarrow E3 \text{ OP3 } E2$
- [6] | $E3$
- [7] $E3 \rightarrow \text{NUM}$
- [8] | $\langle \text{LPAR} \rangle E \langle \text{RPAR} \rangle$
- [9] $\text{NUM} \rightarrow \langle \text{UNUM} \rangle$
- [10] | $\langle \text{SUB} \rangle \langle \text{UNUM} \rangle$
- [11] $\text{OP1} \rightarrow \langle \text{ADD} \rangle$
- [12] | $\langle \text{SUB} \rangle$
- [13] $\text{OP2} \rightarrow \langle \text{MUL} \rangle$
- [14] | $\langle \text{DIV} \rangle$
- [15] $\text{OP3} \rightarrow \langle \text{POW} \rangle$

解析对象： $1+2*-3$

标签流： $\langle \text{UNUM} \rangle \langle \text{ADD} \rangle \langle \text{UNUM} \rangle \langle \text{MUL} \rangle \langle \text{SUB} \rangle \langle \text{UNUM} \rangle$



语法解析树

如何自动生成语法推导树？

- 应用语法规则（从左至右）逐步展开每个非终结符
- 如无二义性问题，则语法解析树唯一
- 如何精准判断当前应采用哪个展开式？避免盲目搜索
- 预测解析（Predictive Parsing）
 - LL(1)文法：Left-to-Right, Leftmost, 前瞻一个字符
 - LL(1)文法的基本要求
 - 无左递归
 - 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
- 会导致搜索算法无限递归下去，不终止

```
[1] E → E OP1 E1
[2]   | E1
[3] E1 → E1 OP2 E2
[4]   | E2
[5] E2 → E3 OP3 E2
[6]   | E3
[7] E3 → NUM
[8]   | <LPAR> E <RPAR>
[9] NUM → <UNUM>
[10]   | <SUB> <UNUM>
[11] OP1 → <ADD>
[12]   | <SUB>
[13] OP2 → <MUL>
[14]   | <DIV>
[15] OP3 → <POW>
```

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 ϵ
 - 将递归规则之外的生成式右侧置于 E' 之前
 - 为 E' 构造生成式,使新旧规则等价

$$\begin{array}{|c} E \rightarrow E \alpha \\ | \beta \end{array} \Rightarrow \begin{array}{|c} E \rightarrow \beta E' \\ E' \rightarrow \alpha E' \\ | \epsilon \end{array}$$

$\{\beta\alpha, \beta\alpha\alpha, \beta\alpha\alpha\alpha, \dots\}$

$$\begin{array}{|c} E \rightarrow E \alpha \\ | \beta \\ | \gamma \end{array} \Rightarrow \begin{array}{|c} E \rightarrow \beta E' \mid \gamma E' \\ E' \rightarrow \alpha E' \\ | \epsilon \end{array}$$

$\{\beta\alpha, \gamma\alpha, \beta\alpha\alpha, \gamma\alpha\alpha, \dots\}$

应用

[1] $E \rightarrow E \text{ OP1 } E1$

[2] | $E1$

[3] $E1 \rightarrow E1 \text{ OP2 } E2$

[4] | $E2$

[5] $E2 \rightarrow E3 \text{ OP3 } E2$

[6] | $E3$

[7] $E3 \rightarrow \text{NUM}$

[8] | $\langle \text{LPAR} \rangle E \langle \text{RPAR} \rangle$

[9] $\text{NUM} \rightarrow \langle \text{UNUM} \rangle$

[10] | $\langle \text{SUB} \rangle \langle \text{UNUM} \rangle$

[11] $\text{OP1} \rightarrow \langle \text{ADD} \rangle$

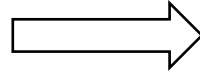
[12] | $\langle \text{SUB} \rangle$

[13] $\text{OP2} \rightarrow \langle \text{MUL} \rangle$

[14] | $\langle \text{DIV} \rangle$

[15] $\text{OP3} \rightarrow \langle \text{POW} \rangle$

消除左递归



[1] $E \rightarrow E1 E'$

[2] $E' \rightarrow \text{OP1 } E1 E'$

[3] | ϵ

[4] $E1 \rightarrow E2 E1'$

[5] $E1' \rightarrow \text{OP2 } E2 E1'$

[6] | ϵ

[7] $E2 \rightarrow E3 \text{ OP3 } E2$

[8] | $E3$

[9] $E3 \rightarrow \text{NUM}$

[10] | $\langle \text{LPAR} \rangle E \langle \text{RPAR} \rangle$

[11] $\text{NUM} \rightarrow \langle \text{UNUM} \rangle$

[12] | $\langle \text{SUB} \rangle \langle \text{UNUM} \rangle$

[13] $\text{OP1} \rightarrow \langle \text{ADD} \rangle$

[14] | $\langle \text{SUB} \rangle$

[15] $\text{OP2} \rightarrow \langle \text{MUL} \rangle$

[16] | $\langle \text{DIV} \rangle$

[17] $\text{OP3} \rightarrow \langle \text{POW} \rangle$

注意间接左递归问题

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

无回溯语法

- 任意非终结符的任意两个生成式产生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
- 消除语法生成规则选择时的不确定性，避免回溯

$$[1] S \rightarrow \alpha \rightarrow \dots \rightarrow \textcolor{red}{< a >} \gamma$$

$$[2] S \rightarrow \beta \rightarrow \dots \rightarrow \textcolor{green}{< b >} \delta$$

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

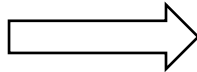
$$E \rightarrow \alpha\beta_1 | \alpha\beta_2 | \dots | \alpha\beta_n | \gamma_1 | \dots | \gamma_j$$

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

应用

[1] $E \rightarrow E1 E'$
[2] $E' \rightarrow OP1 E1 E'$
[3] | ϵ
[4] $E1 \rightarrow E2 E1'$
[5] $E1' \rightarrow OP2 E2 E1'$
[6] | ϵ
[7] $E2 \rightarrow E3 OP3 E2$
[8] | $E3$
[9] $E3 \rightarrow NUM$
[10] | $\langle LPAR \rangle E \langle RPAR \rangle$
[11] $NUM \rightarrow \langle UNUM \rangle$
[12] | $\langle SUB \rangle \langle UNUM \rangle$
[13] $OP1 \rightarrow \langle ADD \rangle$
[14] | $\langle SUB \rangle$
[15] $OP2 \rightarrow \langle MUL \rangle$
[16] | $\langle DIV \rangle$
[17] $OP3 \rightarrow \langle POW \rangle$

消除回溯语法



[1] $E \rightarrow E1 E'$
[2] $E' \rightarrow OP1 E1 E'$
[3] | ϵ
[4] $E1 \rightarrow E2 E1'$
[5] $E1' \rightarrow OP2 E2 E1'$
[6] | ϵ
[7] $E2 \rightarrow E3 E2'$
[8] $E2' \rightarrow OP3 E2$
[9] | ϵ
[10] $E3 \rightarrow NUM$
[11] | $\langle LPAR \rangle E \langle RPAR \rangle$
[12] $NUM \rightarrow \langle UNUM \rangle$
[13] | $\langle SUB \rangle \langle UNUM \rangle$
[14] $OP1 \rightarrow \langle ADD \rangle$
[15] | $\langle SUB \rangle$
[16] $OP2 \rightarrow \langle MUL \rangle$
[17] | $\langle DIV \rangle$
[18] $OP3 \rightarrow \langle POW \rangle$

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] | ϵ
 [4] $E1 \rightarrow E2 E1'$
 [5] $E1' \rightarrow OP2 E2 E1'$
 [6] | ϵ
 [7] $E2 \rightarrow E3 E2'$
 [8] $E2' \rightarrow OP3 E2$
 [9] | ϵ
 [10] $E3 \rightarrow NUM$
 [11] | $\langle LPAR \rangle E \langle RPAR \rangle$
 [12] $NUM \rightarrow \langle UNUM \rangle$
 [13] | $\langle SUB \rangle \langle UNUM \rangle$
 [14] $OP1 \rightarrow \langle ADD \rangle$
 [15] | $\langle SUB \rangle$
 [16] $OP2 \rightarrow \langle MUL \rangle$
 [17] | $\langle DIV \rangle$
 [18] $OP3 \rightarrow \langle POW \rangle$

	<UNUM>	<ADD>	<SUB>	<MUL>	<DIV>	<POW>	<LPAR>	<RPAR>	ϵ
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] ϵ
 [4] $E1 \rightarrow E2 E1'$
 [5] $E1' \rightarrow OP2 E2 E1'$
 [6] ϵ
 [7] $E2 \rightarrow E3 E2'$
 [8] $E2' \rightarrow OP3 E2$
 [9] ϵ
 [10] $E3 \rightarrow NUM$
 [11] $\mid \langle LPAR \rangle E \langle RPAR \rangle$
 [12] $NUM \rightarrow \langle UNUM \rangle$
 [13] $\mid \langle SUB \rangle \langle UNUM \rangle$
 [14] $OP1 \rightarrow \langle ADD \rangle$
 [15] $\mid \langle SUB \rangle$
 [16] $OP2 \rightarrow \langle MUL \rangle$
 [17] $\mid \langle DIV \rangle$
 [18] $OP3 \rightarrow \langle POW \rangle$

	<UNUM>	<ADD>	<SUB>	<MUL>	<DIV>	<POW>	<LPAR>	<RPAR>	ϵ
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

- 解析表同一个单元格最多只有一条可选规则
- 无回溯语法特性：

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

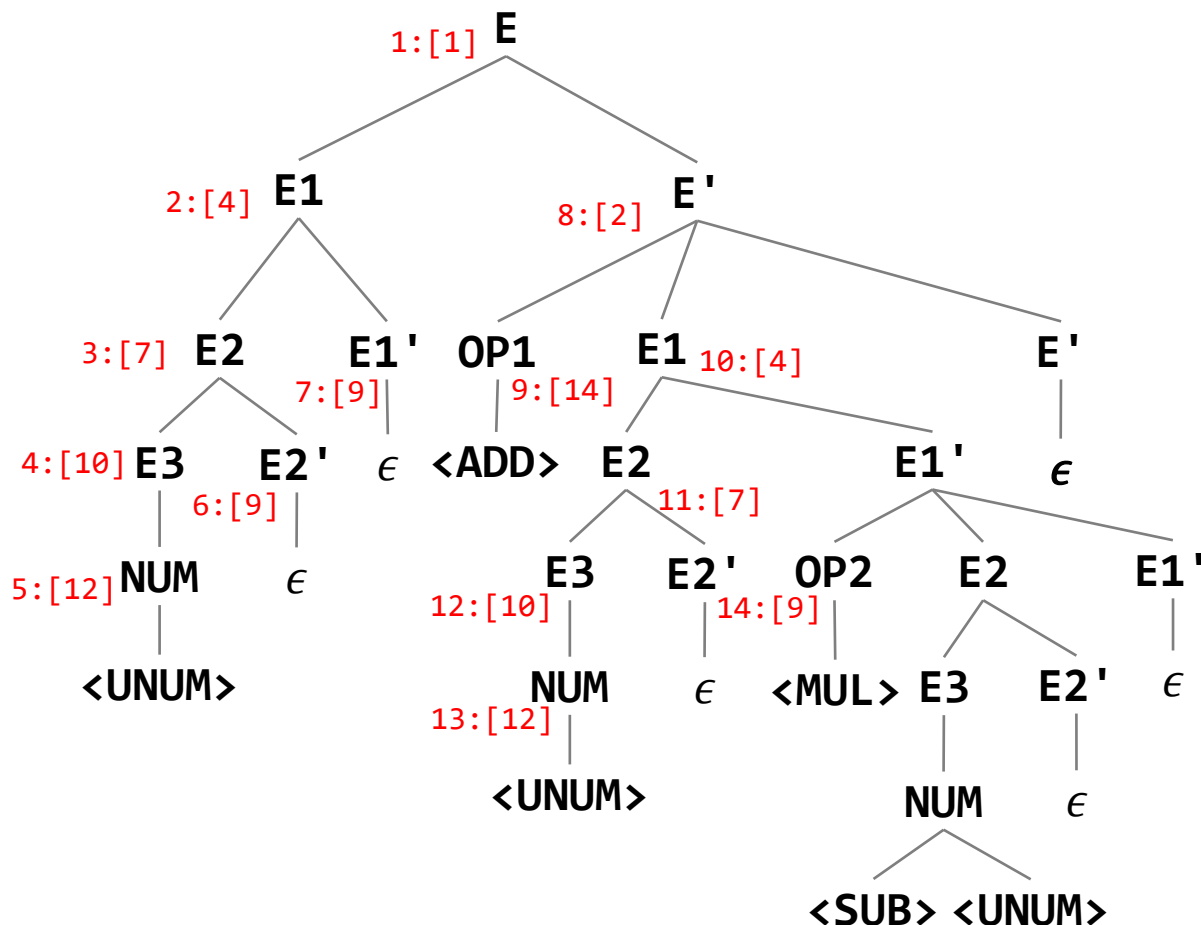
	<UNUM>	<ADD>	<SUB>	<MUL>	<DIV>	<POW>	<LPAR>	<RPAR>
E	[1]		[1]				[1]	
E'		[2]	[2]					[3]
E1	[4]		[4]				[4]	
E1'		[6]	[6]	[5]	[5]		[6]	
E2	[7]		[7]				[7]	
E2'		[9]	[9]	[9]	[9]	[8]	[9]	
E3	[10]		[10]				[11]	
NUM	[12]		[13]					
OP1		[14]	[15]					
OP2				[16]	[17]			
OP3						[18]		

自顶向下解析

解析对象：1+2*-3

标签流：<UNUM><ADD><UNUM><MUL><SUB><UNUM>

- [1] $E \rightarrow E1 E'$
- [2] $E' \rightarrow OP1 E1 E'$
- [3] $\quad \quad \quad | \epsilon$
- [4] $E1 \rightarrow E2 E1'$
- [5] $E1' \rightarrow OP2 E2 E1'$
- [6] $\quad \quad \quad | \epsilon$
- [7] $E2 \rightarrow E3 E2'$
- [8] $E2' \rightarrow OP3 E2$
- [9] $\quad \quad \quad | \epsilon$
- [10] $E3 \rightarrow NUM$
- [11] $\quad \quad \quad | \langle LPAR \rangle E \langle RPAR \rangle$
- [12] $NUM \rightarrow \langle UNUM \rangle$
- [13] $\quad \quad \quad | \langle SUB \rangle \langle UNUM \rangle$
- [14] $OP1 \rightarrow \langle ADD \rangle$
- [15] $\quad \quad \quad | \langle SUB \rangle$
- [16] $OP2 \rightarrow \langle MUL \rangle$
- [17] $\quad \quad \quad | \langle DIV \rangle$
- [18] $OP3 \rightarrow \langle POW \rangle$



语法解析树

练习：

- 将正则表达式CFG改写为LL(1)语法并写出解析查询表

```
[1] REGEX    → REGEX <OR> REGEX
[2]          | CONCAT
[3] CONCAT   → CONCAT CONCAT
[4]          | CLOSURE
[5] CLOSURE  → ITEM <STAR>
[6]          | ITEM
[7] ITEM     → <LPAR> REGEX <RPAR>
[8]          | <CHAR>
```