

# 语法分析

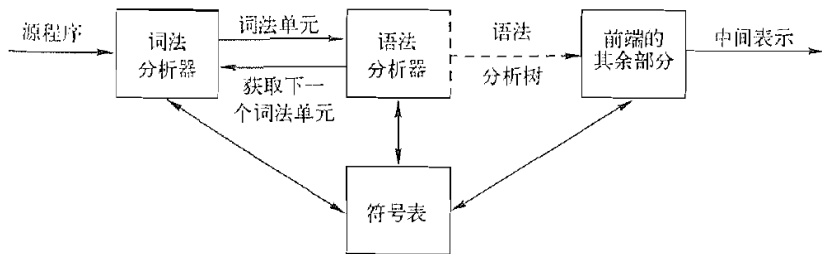
魏恒峰

hfwei@nju.edu.cn

2020 年 11 月 18 日



**输入:** 词法单元流 & 语言的语法规则



**输出:** 语法分析树 (Syntax Tree)

## 语法分析举例

$\langle \text{Stmt} \rangle \rightarrow \langle \text{Id} \rangle = \langle \text{Expr} \rangle ;$
$\langle \text{Stmt} \rangle \rightarrow \{ \langle \text{StmtList} \rangle \}$
$\langle \text{Stmt} \rangle \rightarrow \text{if} ( \langle \text{Expr} \rangle ) \langle \text{Stmt} \rangle$
$\langle \text{StmtList} \rangle \rightarrow \langle \text{Stmt} \rangle$
$\langle \text{StmtList} \rangle \rightarrow \langle \text{StmtList} \rangle \langle \text{Stmt} \rangle$
$\langle \text{Expr} \rangle \rightarrow \langle \text{Id} \rangle$
$\langle \text{Expr} \rangle \rightarrow \langle \text{Num} \rangle$
$\langle \text{Expr} \rangle \rightarrow \langle \text{Expr} \rangle \langle \text{Optr} \rangle \langle \text{Expr} \rangle$
$\langle \text{Id} \rangle \rightarrow x$
$\langle \text{Id} \rangle \rightarrow y$
$\langle \text{Num} \rangle \rightarrow 0$
$\langle \text{Num} \rangle \rightarrow 1$
$\langle \text{Num} \rangle \rightarrow 9$
$\langle \text{Optr} \rangle \rightarrow >$
$\langle \text{Optr} \rangle \rightarrow +$

		$\langle \text{Stmt} \rangle$
if (	$\langle \text{Expr} \rangle$	$\langle \text{Stmt} \rangle$
if (	$\langle \text{Expr} \rangle \langle \text{Optr} \rangle \langle \text{Expr} \rangle$	$\langle \text{Stmt} \rangle$
if (	$\langle \text{Id} \rangle \langle \text{Optr} \rangle \langle \text{Expr} \rangle$	$\langle \text{Stmt} \rangle$
if (	$x \langle \text{Optr} \rangle \langle \text{Expr} \rangle$	$\langle \text{Stmt} \rangle$
if (	$x > \langle \text{Expr} \rangle$	$\langle \text{Stmt} \rangle$
if (	$x > \langle \text{Num} \rangle$	$\langle \text{Stmt} \rangle$
if (	$x > 9$	$\langle \text{Stmt} \rangle$
if (	$x > 9$	{ $\langle \text{StmtList} \rangle$ }
if (	$x > 9$	{ $\langle \text{StmtList} \rangle \langle \text{Stmt} \rangle$ }
if (	$x > 9$	{ $\langle \text{Stmt} \rangle$ }
if (	$x > 9$	{ $\langle \text{Id} \rangle = \langle \text{Expr} \rangle ;$ }
if (	$x > 9$	{ $x = \langle \text{Expr} \rangle ;$ }
if (	$x > 9$	{ $x = \langle \text{Num} \rangle ;$ }
if (	$x > 9$	{ $x = 0 ;$ }
if (	$x > 9$	{ $x = 0 ; \langle \text{Id} \rangle = \langle \text{Expr} \rangle ;$ }
if (	$x > 9$	{ $x = 0 ; y = \langle \text{Expr} \rangle$ }
if (	$x > 9$	{ $x = 0 ; y = \langle \text{Expr} \rangle \langle \text{Optr} \rangle \langle \text{Expr} \rangle$ }
if (	$x > 9$	{ $x = 0 ; y = \langle \text{Id} \rangle \langle \text{Optr} \rangle \langle \text{Expr} \rangle$ }
if (	$x > 9$	{ $x = 0 ; y = y \langle \text{Optr} \rangle \langle \text{Expr} \rangle$ }
if (	$x > 9$	{ $x = 0 ; y = y + \langle \text{Expr} \rangle$ }
if (	$x > 9$	{ $x = 0 ; y = y + \langle \text{Num} \rangle$ }
if (	$x > 9$	{ $x = 0 ; y = y + 1 ;$ }

## 语法分析阶段的主题之一: 上下文无关文法

```

<Stmt> → <Id> = <Expr> ;
<Stmt> → { <StmtList> }
<Stmt> → if ( <Expr> ) <Stmt>
<StmtList> → <Stmt>
<StmtList> → <StmtList> <Stmt>
<Expr> → <Id>
<Expr> → <Num>
<Expr> → <Expr> <Optr> <Expr>
  <Id> → x
  <Id> → y
  <Num> → 0
  <Num> → 1
  <Num> → 9
  <Optr> → >
  <Optr> → +

```

## 语法分析阶段的主题之二: 构建语法分析树

$\langle \text{Stmt} \rangle$			
if (	$\langle \text{Expr} \rangle$	)	$\langle \text{Stmt} \rangle$
if (	$\langle \text{Expr} \rangle$ $\langle \text{Optr} \rangle$ $\langle \text{Expr} \rangle$	)	$\langle \text{Stmt} \rangle$
if (	$\langle \text{Id} \rangle$ $\langle \text{Optr} \rangle$ $\langle \text{Expr} \rangle$	)	$\langle \text{Stmt} \rangle$
if (	x $\langle \text{Optr} \rangle$ $\langle \text{Expr} \rangle$	)	$\langle \text{Stmt} \rangle$
if (	x > $\langle \text{Expr} \rangle$	)	$\langle \text{Stmt} \rangle$
if (	x > $\langle \text{Num} \rangle$	)	$\langle \text{Stmt} \rangle$
if (	x > 9	)	$\langle \text{Stmt} \rangle$
if (	x > 9	{	$\langle \text{StmtList} \rangle$ }
if (	x > 9	{	$\langle \text{StmtList} \rangle$ $\langle \text{Stmt} \rangle$ }
if (	x > 9	{	$\langle \text{Stmt} \rangle$ $\langle \text{Stmt} \rangle$ }
if (	x > 9	{	$\langle \text{Id} \rangle = \langle \text{Expr} \rangle ;$ $\langle \text{Stmt} \rangle$ }
if (	x > 9	{	x = $\langle \text{Expr} \rangle ;$ $\langle \text{Stmt} \rangle$ }
if (	x > 9	{	x = $\langle \text{Num} \rangle ;$ $\langle \text{Stmt} \rangle$ }
if (	x > 9	{	x = 0 $\langle \text{Stmt} \rangle$ }
if (	x > 9	{	x = 0 ; $\langle \text{Id} \rangle = \langle \text{Expr} \rangle ;$ }
if (	x > 9	{	x = 0 ; y = $\langle \text{Expr} \rangle ;$ }
if (	x > 9	{	x = 0 ; y = $\langle \text{Expr} \rangle \langle \text{Optr} \rangle \langle \text{Expr} \rangle ;$ }
if (	x > 9	{	x = 0 ; y = $\langle \text{Id} \rangle \langle \text{Optr} \rangle \langle \text{Expr} \rangle ;$ }
if (	x > 9	{	x = 0 ; y = y $\langle \text{Optr} \rangle \langle \text{Expr} \rangle ;$ }
if (	x > 9	{	x = 0 ; y = y + $\langle \text{Expr} \rangle ;$ }
if (	x > 9	{	x = 0 ; y = y + $\langle \text{Num} \rangle ;$ }
if (	x > 9	{	x = 0 ; y = y + 1 ; }

### 语法分析阶段的主题之三: 错误恢复



报错、**恢复**、继续分析



## <Context-Free Grammar>

上下文无关文法

## Definition (Context-Free Grammar (CFG); 上下文无关文法)

上下文无关文法  $G$  是一个四元组  $G = (T, N, P, S)$ :

- ▶  $T$  是**终结符号** (Terminal) 集合, 对应于词法分析器产生的词法单元;
- ▶  $N$  是**非终结符号** (Non-terminal) 集合;
- ▶  $P$  是**产生式** (Production) 集合;

$$\alpha \in N \longrightarrow \beta \in (T \cup N)^*$$

头部/左部 (Head): **唯一**的非终结符

体部/右部 (Body): 终结符与非终结符构成的串, 也可以是空串  $\epsilon$

- ▶  $S$  为**开始** (Start) 符号。要求  $S \in N$  且唯一。



$$G = (\{S\}, \{(\,,\,)\}, P, S)$$

$$S \rightarrow SS,$$

$$S \rightarrow (S),$$

$$S \rightarrow ()$$

$$G = (\{S\}, \{a, b\}, P, S)$$

$$S \rightarrow aSb$$

$$S \rightarrow \varepsilon$$

$stmt \rightarrow$     **if** *expr* **then** *stmt* **else** *stmt*  
              |    **if** *stmt* **then** *stmt*  
              |    **begin** *stmtList* **end**  
 $stmtList \rightarrow$    *stmt*; *stmtList* | *stmt*

关于语句块与条件语句的文法

**约定:** 如果没有明确指定, 第一个产生式的头部就是开始符号

## 关于终结符号的约定

1) 下述符号是终结符号:

- ① 在字母表里排在前面的小写字母, 比如  $a$ 、 $b$ 、 $c$ 。
- ② 运算符号, 比如  $+$ 、 $*$  等。
- ③ 标点符号, 比如括号、逗号等。
- ④ 数字  $0$ 、 $1$ 、 $\dots$ 、 $9$ 。
- ⑤ **黑体字符串**, 比如 **id** 或 **if**。每个这样的字符串表示一个终结符号。

## 关于非终结符号的约定

2) 下述符号是非终结符号:

- ① 在字母表中排在前面的大写字母, 比如  $A$ 、 $B$ 、 $C$ 。
- ② 字母  $S$ 。它出现时通常表示开始符号。
- ③ 小写、斜体的名字, 比如  $expr$  或  $stmt$ 。

## 推导 (Derivation)

$$E \rightarrow E + E \mid E * E \mid ( E ) \mid \mathbf{id}$$

推导即是將某个产生式的左边**替换**成它的右边

每一步推导需要选择替换哪个非终结符号, 以及使用哪个产生式

## 推导 (Derivation)

$$E \rightarrow E + E \mid E * E \mid ( E ) \mid \mathbf{id}$$

$$E \Rightarrow -E \Rightarrow -(E) \Rightarrow -(E + E) \Rightarrow -(\mathbf{id} + E) \Rightarrow -(\mathbf{id} + \mathbf{id})$$

## 推导 (Derivation)

$$E \rightarrow E + E \mid E * E \mid ( E ) \mid \mathbf{id}$$

$$E \Rightarrow -E \Rightarrow -(E) \Rightarrow -(E + E) \Rightarrow -(\mathbf{id} + E) \Rightarrow -(\mathbf{id} + \mathbf{id})$$

$E \Rightarrow -E$  : 经过一步推导得出

$E \xRightarrow{+} -(\mathbf{id} + E)$  : 经过一步或多步推导得出

$E \xRightarrow{*} -(\mathbf{id} + E)$  : 经过零步或多步推导得出



## 推导 (Derivation)

$$E \rightarrow E + E \mid E * E \mid ( E ) \mid \mathbf{id}$$

$$E \Rightarrow -E \Rightarrow -(E) \Rightarrow -(E + E) \Rightarrow -(\mathbf{id} + E) \Rightarrow -(\mathbf{id} + \mathbf{id})$$

$E \Rightarrow -E$  : 经过一步推导得出

$E \xRightarrow{+} -(\mathbf{id} + E)$  : 经过一步或多步推导得出

$E \xRightarrow{*} -(\mathbf{id} + E)$  : 经过零步或多步推导得出

$$E \Rightarrow -E \Rightarrow -(E) \Rightarrow -(E + E) \Rightarrow -(E + \mathbf{id}) \Rightarrow -(\mathbf{id} + \mathbf{id})$$

## Definition (Sentential Form; 句型)

如果  $S \xRightarrow{*} \alpha$ , 且  $\alpha \in (T \cup N)^*$ , 则称  $\alpha$  是文法  $G$  的一个**句型**。

$$E \rightarrow E + E \mid E * E \mid ( E ) \mid \mathbf{id}$$

$$E \Rightarrow -E \Rightarrow -(E) \Rightarrow -(E+E) \Rightarrow -(\mathbf{id} + E) \Rightarrow -(\mathbf{id} + \mathbf{id})$$

### Definition (Sentential Form; 句型)

如果  $S \xRightarrow{*} \alpha$ , 且  $\alpha \in (T \cup N)^*$ , 则称  $\alpha$  是文法  $G$  的一个**句型**。

$$E \rightarrow E + E \mid E * E \mid ( E ) \mid \mathbf{id}$$

$$E \Rightarrow -E \Rightarrow -(E) \Rightarrow -(E+E) \Rightarrow -(\mathbf{id} + E) \Rightarrow -(\mathbf{id} + \mathbf{id})$$

### Definition (Sentence; 句子)

如果  $S \xRightarrow{*} w$ , 且  $w \in T^*$ , 则称  $w$  是文法  $G$  的一个**句子**。

Definition (文法  $G$  生成的语言  $L(G)$ )

文法  $G$  的**语言**  $L(G)$  是它能推导出的**所有句子**构成的集合。

$$w \in L(G) \iff S \xRightarrow{*} w$$

$$S \rightarrow SS,$$

$$S \rightarrow (S),$$

$$S \rightarrow ()$$

$$L(G) =$$

$$S \rightarrow SS,$$

$$S \rightarrow (S),$$

$$S \rightarrow ()$$

$$S \rightarrow aSb$$

$$S \rightarrow \varepsilon$$

$L(G) = \{\text{长度} \geq 2 \text{ 的已匹配括号串}\}$

$L(G) =$

$$S \rightarrow SS,$$

$$S \rightarrow (S),$$

$$S \rightarrow ()$$

$L(G) = \{\text{长度} \geq 2 \text{ 的已匹配括号串}\}$

$$S \rightarrow aSb$$

$$S \rightarrow \varepsilon$$

$$L(G) = \{a^n b^n \mid n \geq 0\}$$

字母表  $\Sigma = \{a, b\}$  上的所有回文串 (Palindrome) 构成的语言



字母表  $\Sigma = \{a, b\}$  上的所有回文串 (Palindrome) 构成的语言

$$\begin{array}{l} P \rightarrow \epsilon \\ P \rightarrow 0 \\ P \rightarrow 1 \end{array}$$

$$P \rightarrow 0P0$$

$$P \rightarrow 1P1$$

字母表  $\Sigma = \{a, b\}$  上的所有回文串 (Palindrome) 构成的语言

$$\begin{array}{l} P \rightarrow \epsilon \\ P \rightarrow 0 \\ P \rightarrow 1 \end{array}$$

$$P \rightarrow 0P0$$

$$P \rightarrow 1P1$$

$$P \rightarrow \epsilon \mid 0 \mid 1 \mid 0P0 \mid 1P1$$

## 最左 (leftmost) 推导与最右 (rightmost) 推导

$$E \rightarrow E + E \mid E * E \mid ( E ) \mid \mathbf{id}$$

$$E \xRightarrow{\text{lm}} -E \xRightarrow{\text{lm}} -(E) \xRightarrow{\text{lm}} -(E + E) \xRightarrow{\text{lm}} -(\mathbf{id} + E) \xRightarrow{\text{lm}} -(\mathbf{id} + \mathbf{id})$$

## 最左 (leftmost) 推导与最右 (rightmost) 推导

$$E \rightarrow E + E \mid E * E \mid ( E ) \mid \mathbf{id}$$

$$E \xRightarrow{\text{lm}} -E \xRightarrow{\text{lm}} -(E) \xRightarrow{\text{lm}} -(E + E) \xRightarrow{\text{lm}} -(\mathbf{id} + E) \xRightarrow{\text{lm}} -(\mathbf{id} + \mathbf{id})$$

$E \xRightarrow{\text{lm}} -E$  : 经过一步最左推导得出

$E \xRightarrow{+}_{\text{lm}} -(\mathbf{id} + E)$  : 经过一步或多步最左推导得出

$E \xRightarrow{*}_{\text{lm}} -(\mathbf{id} + E)$  : 经过零步或多步最左推导得出

## 最左 (leftmost) 推导与最右 (rightmost) 推导

$$E \rightarrow E + E \mid E * E \mid ( E ) \mid \mathbf{id}$$

$$E \xRightarrow{\text{lm}} -E \xRightarrow{\text{lm}} -(E) \xRightarrow{\text{lm}} -(E + E) \xRightarrow{\text{lm}} -(\mathbf{id} + E) \xRightarrow{\text{lm}} -(\mathbf{id} + \mathbf{id})$$

$$E \xRightarrow{\text{lm}} -E : \text{经过一步最左推导得出}$$

$$E \xRightarrow[\text{lm}]{+} -(\mathbf{id} + E) : \text{经过一步或多步最左推导得出}$$

$$E \xRightarrow[\text{lm}]{*} -(\mathbf{id} + E) : \text{经过零步或多步最左推导得出}$$

$$E \xRightarrow{\text{rm}} -E \xRightarrow{\text{rm}} -(E) \xRightarrow{\text{rm}} -(E + E) \xRightarrow{\text{rm}} -(E + \mathbf{id}) \xRightarrow{\text{rm}} -(\mathbf{id} + \mathbf{id})$$

## Definition (Left-sentential Form; 最左句型)

如果  $S \xRightarrow[\text{lm}]{*} \alpha$ , 且  $\alpha \in (T \cup N)^*$ , 则称  $\alpha$  是文法  $G$  的一个**最左句型**。

$$E \xRightarrow{\text{lm}} -E \xRightarrow{\text{lm}} -(E) \xRightarrow{\text{lm}} -(E + E) \xRightarrow{\text{lm}} -(\text{id} + E) \xRightarrow{\text{lm}} -(\text{id} + \text{id})$$

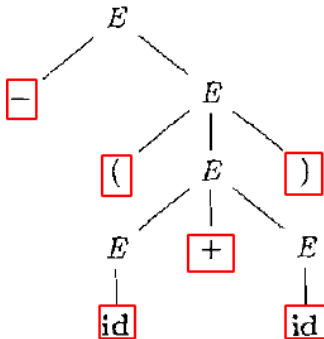
## Definition (Right-sentential Form; 最右句型)

如果  $S \xRightarrow[\text{rm}]{*} \alpha$ , 且  $\alpha \in (T \cup N)^*$ , 则称  $\alpha$  是文法  $G$  的一个**最右句型**。

$$E \xRightarrow{\text{rm}} -E \xRightarrow{\text{rm}} -(E) \xRightarrow{\text{rm}} -(E + E) \xRightarrow{\text{rm}} -(E + \text{id}) \xRightarrow{\text{rm}} -(\text{id} + \text{id})$$

## 语法分析树

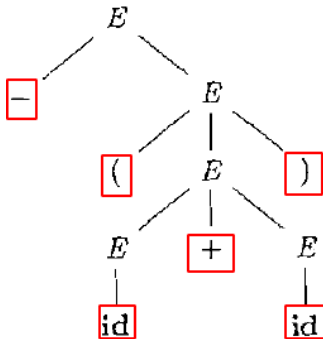
语法分析树是静态的, 它不关心动态的推导顺序



一棵语法分析树对应多个推导

## 语法分析树

语法分析树是静态的, 它不关心动态的推导顺序



一棵语法分析树对应多个推导

但是, 一棵语法分析树与**最左 (最右) 推导**一一对应



Thank  
You!



Office 926

hfwei@nju.edu.cn