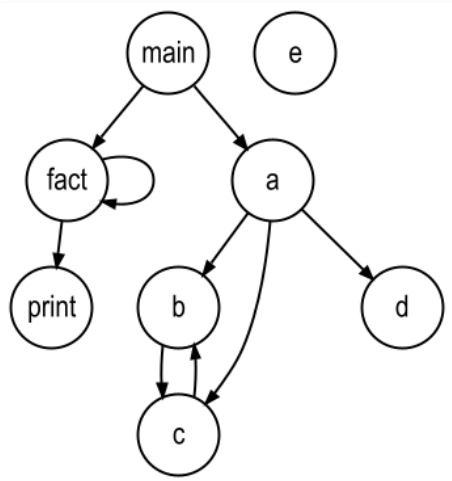
语法分析
(1. 上下文无关文法、ANTLR 4 语法分析器)

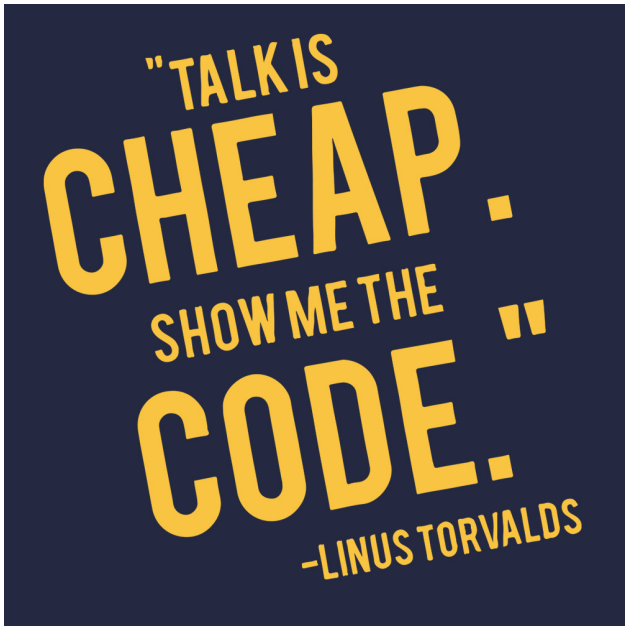
魏恒峰

hfwei@nju.edu.cn

2022 年 11 月 21 日

Cymbol.g4


main
e
fact
a
print
b
d
c


"TALK IS
CHEAP.
SHOW ME THE
CODE."
-LINUS TORVALDS


ON/AC
M+
M–
MRC
GT
OFF
7
8
9
÷
4
5
6
X
%
1
2
3
–
MU
0
00
.
+
=

IfStat.g4

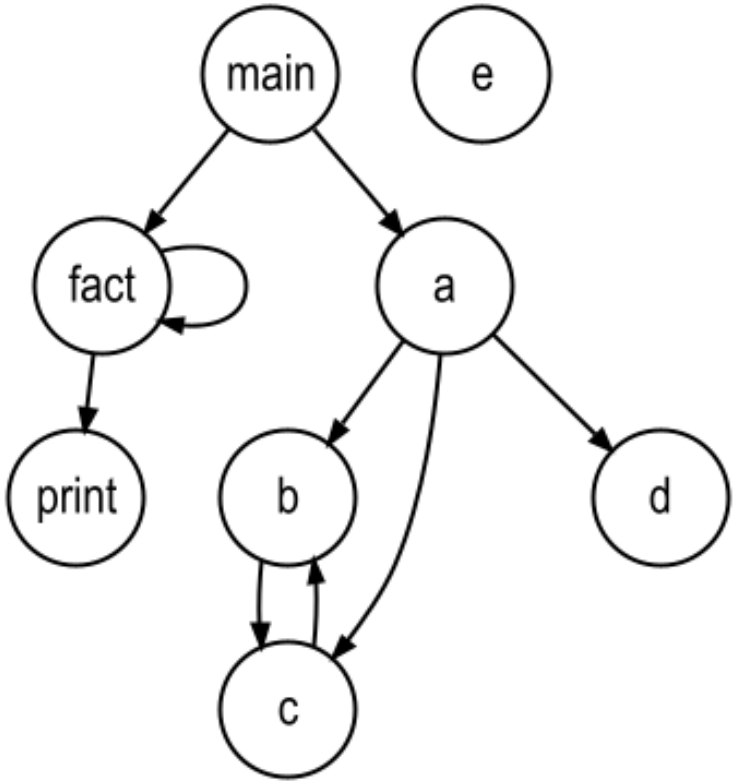
```
stat : 'if' expr 'then' stat
     | 'if' expr 'then' stat 'else' stat
     | expr
     ;
```

Expr.g4

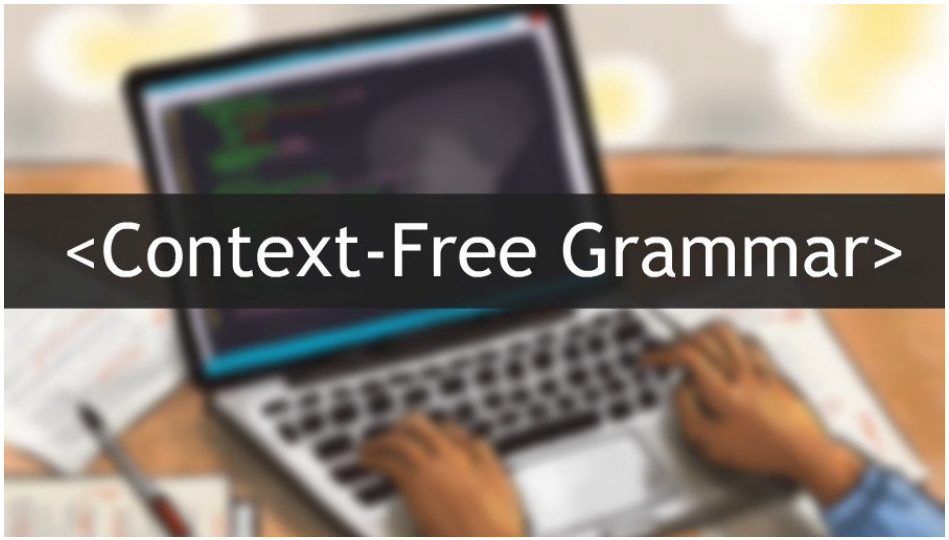
```
expr :
    | expr '*' expr
    | expr '+' expr
    | DIGIT
    ;
```

结合性、优先级

Call Graphs



Calculator



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;

$$\boxed{A \in N \longrightarrow \alpha \in (T \cup N)^*}$$

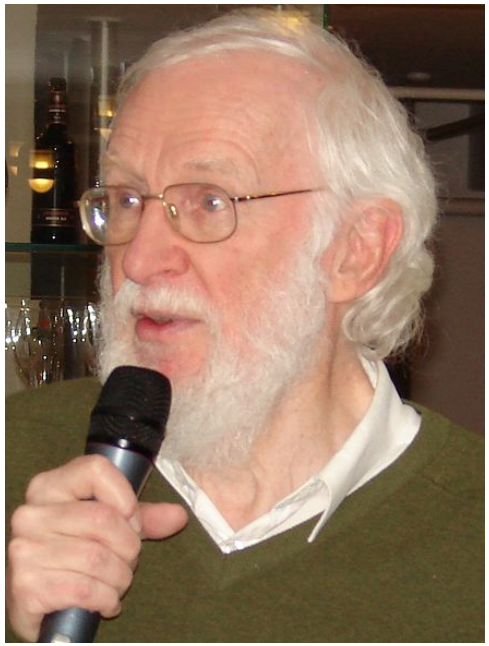
头部/左部 (Head) A: **单个**非终结符

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

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

[Extended] Backus-Naur form ([E]BNF)

John Backus
(1924 ∼ 2007)



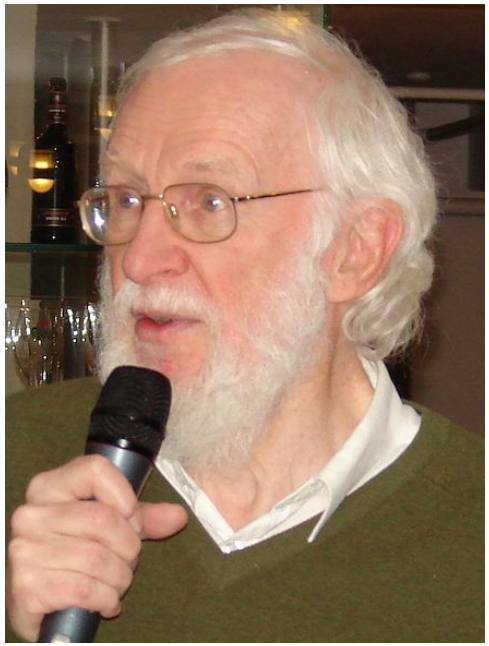
Peter Naur
(1928 ∼ 2016)

Niklaus Wirth (1934 ∼)

[Extended] Backus-Naur form ([E]BNF)

John Backus
(1924 ∼ 2007)

1977 (FORTRAN)



Peter Naur
(1928 ∼ 2016)

2005 (ALGOL60)

Niklaus Wirth (1934 ∼)

1984 (PLs; PASCAL)

语义: 上下文无关文法 G 定义了一个**语言** $L(G)$

语义: 上下文无关文法 G 定义了一个**语言** $L(G)$

语言是**串**的集合

串从何来?

推导 (Derivation)

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

推导即是将某个产生式的左边**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)

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

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

推导 (Derivation)

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

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

$E \implies -E$: 经过一步推导得出

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

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

推导 (Derivation)

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

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

$E \implies -E$: 经过一步推导得出

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

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

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

Definition (Sentential Form; 句型)

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

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

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

Definition (Sentential Form; 句型)

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

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

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

Definition (Sentence; 句子)

如果 $S \overset{*}{\Rightarrow} w$, 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。

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

关于文法 G 的**两个基本问题:**

- **Membership 问题:** 给定字符串 $x \in T^*$, $x \in L(G)$?
- $L(G)$ 究竟是什么?

给定字符串 $x \in T^*$, $x \in L(G)$?

(即, 检查 x 是否符合文法 G)

给定字符串 $x \in T^*$, $x \in L(G)$?

(即, 检查 x 是否符合文法 G)

这就是**语法分析器**的任务:

为输入的词法单元流寻找推导、**构建语法分析树**, 或者报错

$L(G)$ 是什么?

这是**程序设计语言设计者**需要考虑的问题

$$S \to SS$$

$$S \to (S)$$

$$S \to ()$$

$$S \to \epsilon$$

$L(G) =$

$$S \to SS$$

$$S \to (S)$$

$$S \to ()$$

$$S \to \epsilon$$

$L(G) = \{$良匹配括号串$\}$

$$S \to SS$$

$$S \to (S)$$

$$S \to ()$$

$$S \to \epsilon$$

$L(G) = \{$良匹配括号串$\}$

$$S \to aSb$$

$$S \to \epsilon$$

$L(G) =$

$$S \to SS$$

$$S \to (S)$$

$$S \to ()$$

$$S \to \epsilon$$

$L(G) = \{\text{良匹配括号串}\}$

$$S \to aSb$$

$$S \to \epsilon$$

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

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

$$S \to aSa$$

$$S \to bSb$$

$$S \to a$$

$$S \to b$$

$$S \to \epsilon$$

$$\{b^n a^m b^{2n} \mid n \geq 0, m \geq 0\}$$

$$\{b^n a^m b^{2n} \mid n \geq 0, m \geq 0\}$$

$$S \to bSbb \mid A$$

$$A \to aA \mid \epsilon$$

$$\{x \in \{a, b\}^* \mid x \text{ 中 } a, b \text{ 个数相同}\}$$

$$\{x \in \{a,b\}^* \mid x \text{ 中 } a,b \text{ 个数相同}\}$$

$$\boxed{V \to aVbV \mid bVaV \mid \epsilon}$$

$$\{x \in \{a, b\}^* \mid x \text{ 中 } a, b \text{ 个数不同}\}$$

$$\{x \in \{a,b\}^* \mid x \text{ 中 } a,b \text{ 个数不同}\}$$

$$S \to T \mid U$$

$$T \to VaT \mid VaV$$

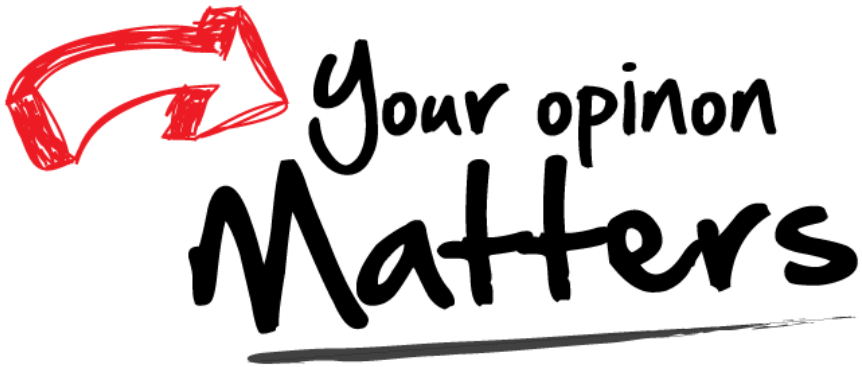
$$U \to VbU \mid VbV$$

$$V \to aVbV \mid bVaV \mid \epsilon$$

$$\{x \in \{a,b\}^* \mid x \text{ 中 } a,b \text{ 个数不同}\}$$

$$\begin{aligned}
S &\to T \mid U \\
T &\to VaT \mid VaV \\
U &\to VbU \mid VbV \\
V &\to aVbV \mid bVaV \mid \epsilon
\end{aligned}$$


Thank
You!



Office 926

hfwei@nju.edu.cn