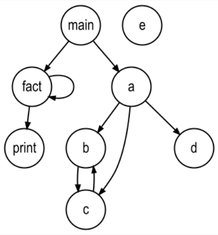
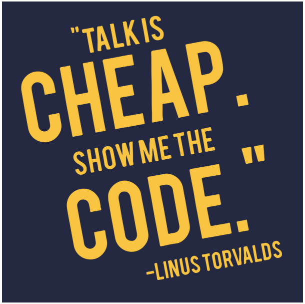
(1. ANTLR 4)

hfwei@nju.edu.cn

20230324

Cymbol.g4


main
e
fact
a
print
b
d
c


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

IfStat.g4

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

IfStat.g4

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

```
stat : matched_stat | open_stat ;

matched_stat : 'if' expr 'then' matched_stat 'else' matched_stat
             | expr
             ;

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

Left-Factoring

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

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

expr : ID ;
```

,

, ANTLR 4

Expr.g4

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

Expr.g4

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

$(-, \quad \hat{})$

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

ANTLR 4 ()

```
expr : expr '-' term
     | term
     ;

term : term '*' factor
     | factor
     ;

factor : DIGIT ;
```

()

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

ANTLR 4 ()

```
expr : expr '-' term
     | term
     ;

term : term '*' factor
     | factor
     ;

factor : DIGIT ;
```

()

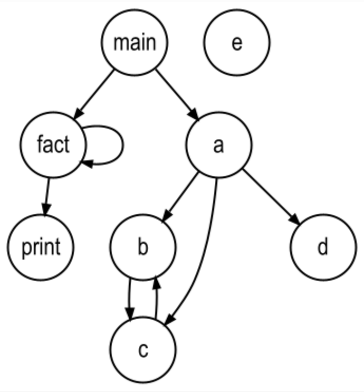
```
expr :
   | expr '*' expr
   | expr '-' expr
   | DIGIT
   ;
```

ANTLR 4 ()

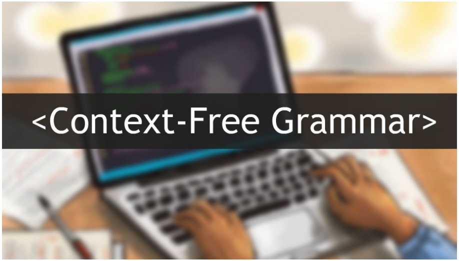
```
expr : term expr_prime ;
expr_prime : '-' term expr_prime
           |
           ;

term : factor term_prime ;
term_prime : '*' factor term_prime
           |
           ;

factor : DIGIT ;
```

()

Call Graphs



Calculator


<Context-Free Grammar>

Definition (Context-Free Grammar (CFG);)

G $G = (T, N, P, S)$:

- T (Terminal) , ;
- N (Non-terminal) ;
- P (Production) ;

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

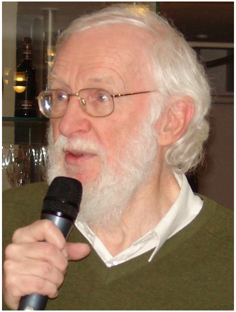
/ (Head) A:

/ (Body) α: , ϵ

- S (Start) $S \in N$

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

John Backus
(1924 ∼ 2007)



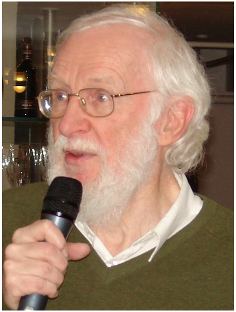
Peter Naur
(1928 ∼ 2016)

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)

: G $L(G)$

: G $L(G)$

?

(Derivation)

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

,

(Derivation)

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

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

(Derivation)

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

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

$$E \implies -E :$$

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

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

(Derivation)

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

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

$$E \implies -E :$$

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

$$E \xRightarrow{*} -(\mathbf{id} + E) :$$

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

Definition (Sentential Form;)

$S \overset{*}{\Rightarrow} \alpha, \;\; \alpha \in (T \cup N)^*, \;\; \alpha \;\; G$

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

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

Definition (Sentential Form;)

$S \overset{*}{\Rightarrow} \alpha, \;\; \alpha \in (T \cup N)^*, \;\; \alpha \;\; G$

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

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

Definition (Sentence;)

$S \overset{*}{\Rightarrow} w, \;\; w \in T^*, \;\; w \;\; G$

Definition (G $L(G)$)

G $L(G)$

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

G :

- **Membership :** $x \in T^*, x \in L(G)$?
- $L(G)$?

$$x \in T^*, x \in L(G)?$$

$$(, xG)$$

$x \in T^*, x \in L(G)?$

$(, xG)$

:

,

$L(G)$?

$$\boxed{\begin{aligned} S &\to SS \\ S &\to (S) \\ S &\to () \\ S &\to \epsilon \end{aligned}}$$

$$L(G) =$$

$$\boxed{\begin{aligned} &S \to SS \\ &S \to (S) \\ &S \to () \\ &S \to \epsilon \end{aligned}}$$

$$L(G) = \{\}$$

$$\boxed{\begin{aligned} S &\to SS \\ S &\to (S) \\ S &\to () \\ S &\to \epsilon \end{aligned}}$$

$$L(G) = \{\}$$

$$\boxed{\begin{aligned} S &\to aSb \\ S &\to \epsilon \end{aligned}}$$

$$L(G) =$$

$$\boxed{\begin{aligned} &S \to SS \\ &S \to (S) \\ &S \to () \\ &S \to \epsilon \end{aligned}}$$

$$L(G) = \{\}$$

$$\boxed{\begin{aligned} &S \to aSb \\ &S \to \epsilon \end{aligned}}$$

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

$$\Sigma = \{a, b\} \quad \text{(Palindrome)}$$

$$\Sigma = \{a, b\} \quad \text{(Palindrome)}$$

$$\boxed{\begin{aligned} S &\to aSa \\ S &\to bSb \\ S &\to a \\ S &\to b \\ S &\to \epsilon \end{aligned}}$$

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

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

$$\boxed{\begin{aligned} S &\to bSbb \mid A \\ A &\to aA \mid \epsilon \end{aligned}}$$

$$\{x \in \{a, b\}^* \mid x \quad a, b \ \}$$

$$\{x \in \{a,b\}^* \mid x \quad a, b \ \}$$

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

$$\{x \in \{a, b\}^* \mid x \quad a, b \ \}$$

$$\{x \in \{a,b\}^* \mid x \quad a, b \ \}$$

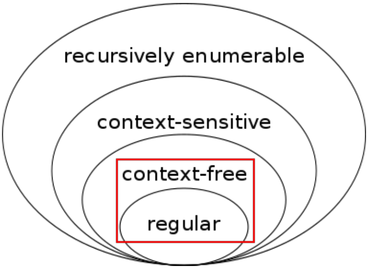
$$\boxed{\begin{aligned} S &\to T \mid U \\ T &\to VaT \mid VaV \\ U &\to VbU \mid VbV \\ V &\to aVbV \mid bVaV \mid \epsilon \end{aligned}}$$

$$\{x \in \{a,b\}^* \mid x \quad a, b \ \}$$

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

Let me know if you have a proof.

?

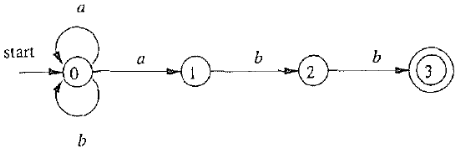

recursively enumerable
context-sensitive
context-free
regular

$$r \quad L(r)$$

$$r = (a|b)^* abb$$

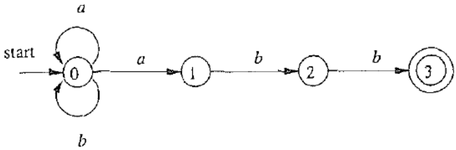
$$r \quad L(r)$$

$$r = (a|b)^* abb$$



$$r \quad L(r)$$

$$r = (a|b)^* abb$$



$$A_0 \rightarrow aA_0 \mid bA_0 \mid aA_1$$

$$A_1 \rightarrow bA_2$$

$$A_2 \rightarrow bA_3$$

$$A_3 \rightarrow \epsilon$$

, $\delta(A_i, \epsilon) = A_j$, $A_i \rightarrow A_j$

$$\boxed{\begin{aligned} S &\to aSb \\ S &\to \epsilon \end{aligned}}$$

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

Theorem

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

Theorem

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

Theorem

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

$$r\colon\ L(r) = L$$

Theorem

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

$$r\colon\ L(r) = L$$

$$D(r)\colon\ L(D(r)) = L;\quad k$$

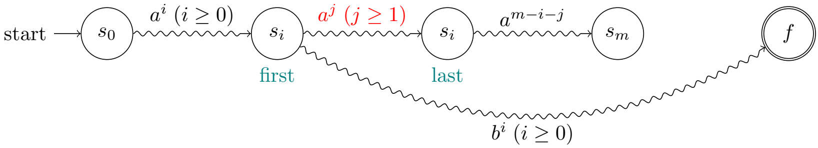
Theorem

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

$$r:\ L(r) = L$$

$$D(r):\ L(D(r)) = L;\ \ k$$

$$\boxed{a^m (m > k)}$$



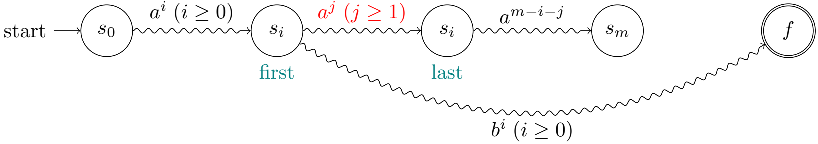
Theorem

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

$$r\colon\ L(r) = L$$

$$D(r)\colon\ L(D(r)) = L;\ \ k$$

$$\boxed{a^m (m > k)}$$



$$D(r)\ \ a^{i+j} b^i;\ \mathbf{!}$$

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

Pumping Lemma for Regular Languages

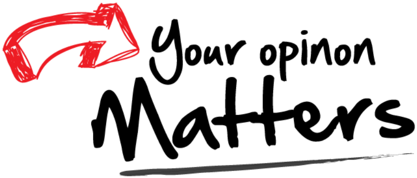
$$L = \{a^n b^n \mid n \geq 0\}$$

Pumping Lemma for Regular Languages

$$L = \{a^n b^n c^n \mid n \geq 0\}$$

Pumping Lemma for Context-free Languages


Thank
You!



Office 926

hfwei@nju.edu.cn