
Análise Sintática LL(1)

Fim de Arquivo

Criar um novo não terminal como símbolo inicial

$$E \rightarrow E + T$$

$$E \rightarrow E - T$$

$$E \rightarrow T$$

$$T \rightarrow T * F$$

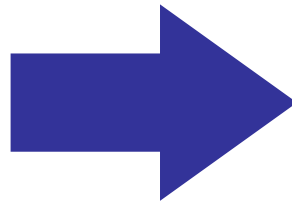
$$T \rightarrow T / F$$

$$T \rightarrow F$$

$$F \rightarrow \text{id}$$

$$F \rightarrow \text{num}$$

$$F \rightarrow (E)$$



$$S \rightarrow E \$$$

$$E \rightarrow E + T$$

$$E \rightarrow E - T$$

$$E \rightarrow T$$

$$T \rightarrow T * F$$

$$T \rightarrow T / F$$

$$T \rightarrow F$$

$$F \rightarrow \text{id}$$

$$F \rightarrow \text{num}$$

$$F \rightarrow (E)$$

Análise Descendente (Predictive Parsing)

$$S \rightarrow E \$$$

$$E \rightarrow E + T$$

$$E \rightarrow E - T$$

$$E \rightarrow T$$

$$T \rightarrow T * F$$

$$T \rightarrow T / F$$

$$T \rightarrow F$$

$$F \rightarrow \text{id}$$

$$F \rightarrow \text{num}$$

$$F \rightarrow (E)$$

Vamos aplicar a mesma técnica para essa outra gramática ...

Análise Descendente (Predictive Parsing)

```
void S() { E(); eat(EOF); }
```

```
void E() {  
    switch (tok) {  
        case ?: E(); eat(PLUS); T(); break;  
        case ?: E(); eat(MINUS); T(); break;  
        case ?: T(); break;  
        default: error(); }  
}
```

```
void T() {  
    switch (tok) {  
        case ?: T(); eat(TIMES); F(); break;  
        case ?: T(); eat(DIV); F(); break;  
        case ?: F(); break;  
        default: error(); }  
}
```

Funciona ???

- Como seria a execução para 1*2-3+4 ?
- E para 1*2-3?

$S \rightarrow E \$$
 $E \rightarrow E + T$
 $E \rightarrow E - T$
 $E \rightarrow T$
 $T \rightarrow T * F$
 $T \rightarrow T / F$
 $T \rightarrow F$
 $F \rightarrow \text{id}$
 $F \rightarrow \text{num}$
 $F \rightarrow (E)$

Análise Descendente (Predictive Parsing)

Como decidir entre E+T, E-T e T na função que implementa o não-terminal E?

- Tanto E como T podem derivar cadeias começando com **id**, **num** ou **(**
- E se fosse possível olhar um número $k > 1$ de símbolos para frente na entrada?

Análise Descendente (Predictive Parsing)

Como decidir entre $E+T$, $E-T$ e T na função que implementa o não-terminal E ?

- Tanto E como T podem derivar cadeias começando com **id**, **num** ou **(**
- E se fosse possível olhar um número $k > 1$ de símbolos para frente na entrada?

Essas cadeias podem ter tamanho arbitrário: O problema permanece

Análise descendente recursiva (preditiva) só funciona onde o primeiro símbolo terminal de cada sub-expressão permite escolher a produção adequada a ser utilizada na derivação

Análise Descendente

$S \rightarrow E \$$

$E \rightarrow F + T$

$E \rightarrow G + T$

$F \rightarrow \text{id}$

$G \rightarrow \text{num}$

$T \rightarrow \text{num}$

Como seria a análise da cadeia **num+num\$** ?

Análise Descendente

$S \rightarrow E \$$

$E \rightarrow F + T$

$E \rightarrow G + T$

$F \rightarrow \text{id}$

$G \rightarrow \text{num}$

$T \rightarrow \text{num}$

Como seria a análise da cadeia $\text{num+num\$}$?

$S \rightarrow E \$ \rightarrow G+T\$ \rightarrow \text{num}+T\$ \rightarrow \text{num+num\$}$

Análise Descendente LL(1)

$S \rightarrow E \$$

$E \rightarrow F + T$

$E \rightarrow G + T$

$F \rightarrow \text{id}$

$G \rightarrow \text{num}$

$T \rightarrow \text{num}$

	id	num	+	\$
S	$S \rightarrow E \$$	$S \rightarrow E \$$		
E	$E \rightarrow F + T$	$E \rightarrow G + T$		
F	$F \rightarrow \text{id}$			
G		$G \rightarrow \text{num}$		
T		$T \rightarrow \text{num}$		

Como seria a análise da cadeia $\text{num}+\text{num}\$$?

$S \rightarrow E \$ \rightarrow G+T\$ \rightarrow \text{num}+T\$ \rightarrow \text{num}+\text{num}\$$

Construindo um Predictive Parser LL(1)

- Precisa-se saber os terminais que podem ser gerados por um Não-Terminal
Conjunto **FIRST**
- Precisa-se saber os terminais que podem vir após um Não-Terminal
Conjunto **FOLLOW**
- Precisa-se saber se um Não-Terminal pode gerar a cadeia vazia
Nullable

Conjunto FIRST

- Dada uma cadeia γ de terminais e não terminais $\text{FIRST}(\gamma)$ é o conjunto de todos os terminais que podem iniciar uma cadeia derivada de γ .

- Exemplo usando gramática ao lado:

$\text{FIRST}((E)\$) = \{ (\}$

$\text{FIRST}(\text{num}*F\$) = \{ \text{num} \}$

$\text{FIRST}(\text{id}*F\$) = \{ \text{id} \}$

$S \rightarrow E \$$
$E \rightarrow E + T$
$E \rightarrow E - T$
$E \rightarrow T$
$T \rightarrow T * F$
$T \rightarrow T / F$
$T \rightarrow F$
$F \rightarrow \text{id}$
$F \rightarrow \text{num}$
$F \rightarrow (E)$

Conjunto FIRST

- Dada uma cadeia γ de terminais e não terminais $\text{FIRST}(\gamma)$ é o conjunto de todos os terminais que podem iniciar uma cadeia derivada de γ .

- Exemplo usando gramática ao lado:

$$\gamma = T * F$$

$$\text{FIRST}(\gamma) = \{ \text{id}, \text{num}, (\}$$

$S \rightarrow E \$$
$E \rightarrow E + T$
$E \rightarrow E - T$
$E \rightarrow T$
$T \rightarrow T * F$
$T \rightarrow T / F$
$T \rightarrow F$
$F \rightarrow \text{id}$
$F \rightarrow \text{num}$
$F \rightarrow (E)$

Predictive Parsing

Se uma gramática tem produções da forma:

$$X \rightarrow \gamma_1$$

$$X \rightarrow \gamma_2$$

- Caso os conjuntos $\text{FIRST}(\gamma_1)$ e $\text{FIRST}(\gamma_2)$ tenham intersecção, então a gramática não pode ser analisada com um *predictive parser*

Por que?

A função recursiva não vai saber que caso executar

Calculando FIRST

$Z \rightarrow d$

$Z \rightarrow X Y Z$

$Y \rightarrow$

$Y \rightarrow c$

$X \rightarrow Y$

$X \rightarrow a$

- Como seria para $\gamma = XYZ$?

- Pode-se simplesmente fazer $\text{FIRST}(XYZ) = \text{FIRST}(X)$?

Nullable

Nullable(X) é verdadeiro se X pode derivar a cadeia vazia.

$Z \rightarrow d$

$Z \rightarrow X Y Z$

$Y \rightarrow$

$Y \rightarrow c$

$X \rightarrow Y$

$X \rightarrow a$

Nullable(Y) = yes

Nullable(X) = yes

Nullable(Z) = no

Follow

$\text{FOLLOW}(X)$ é o conjunto de terminais que podem imediatamente seguir X

$t \in \text{FOLLOW}(X)$ se existe alguma derivação contendo Xt

Cuidado com derivações da forma $XYZt$, onde Y e Z podem ser vazios

$Z \rightarrow d$

$Z \rightarrow X Y Z$

$Y \rightarrow$

$Y \rightarrow c$

$X \rightarrow Y$

$X \rightarrow a$

$\text{FOLLOW}(Y) = \{d, a, c\}$

$\text{FOLLOW}(Z) = \{ \}$

FIRST, FOLLOW e Nullable

- $\text{Nullable}(X)$ é verdadeiro se X pode derivar a cadeia vazia
- $\text{FIRST}(\gamma)$ é o conjunto de terminais que podem iniciar cadeias derivadas de γ
- $\text{FOLLOW}(X)$ é o conjunto de terminais que podem imediatamente seguir X
 - $t \in \text{FOLLOW}(X)$ se existe alguma derivação contendo Xt
 - Cuidado com derivações da forma $XYZt$, onde Y e Z podem ser vazios

FIRST, FOLLOW e Nullable

Initialize FIRST and FOLLOW to all empty sets, and Nullable to all false.

```
for each terminal symbol  $Z$   $FIRST[Z] \leftarrow \{Z\}$ 
repeat
  for each production  $X \rightarrow Y_1 Y_2 \dots Y_k$ 
    if  $Y_1 \dots Y_k$  are all Nullable (or if  $k = 0$ ) then  $Nullable[X] \leftarrow \text{true}$ 
    for each  $i$  from 1 to  $k$ , each  $j$  from  $i + 1$  to  $k$ 
      if  $Y_1 \dots Y_{i-1}$  are all Nullable (or if  $i = 1$ )
        then  $FIRST[X] \leftarrow FIRST[X] \cup FIRST[Y_i]$ 
      if  $Y_{i+1} \dots Y_k$  are all Nullable (or if  $i = k$ )
        then  $FOLLOW[Y_i] \leftarrow FOLLOW[Y_i] \cup FOLLOW[X]$ 
      if  $Y_{i+1} \dots Y_{j-1}$  are all Nullable (or if  $i + 1 = j$ )
        then  $FOLLOW[Y_i] \leftarrow FOLLOW[Y_i] \cup FIRST[Y_j]$ 
  until FIRST, FOLLOW, and Nullable did not change in this iteration.
```

Generalizando para cadeias: FIRST

- $\text{FIRST}(X \gamma) = \text{FIRST}[X]$, if not nullable[X]
- $\text{FIRST}(X \gamma) = \text{FIRST}[X] \cup \text{FIRST}(\gamma)$, if nullable[X]
- A cadeia γ é *Nullable* se cada símbolo em γ é *Nullable*

Generalizando para cadeias: FOLLOW

- Se houver uma produção $A \rightarrow \alpha B \beta$, então, tudo em $\text{FIRST}(\beta)$ irá para $\text{FOLLOW}(B)$
- Se houver uma produção $A \rightarrow \alpha B$, ou uma produção $A \rightarrow \alpha B \beta$ onde $\text{FIRST}(\beta)$ é Nullable, então tudo em $\text{FOLLOW}(A)$ irá para $\text{FOLLOW}(B)$

Construindo um Predictive Parser LL(1)

- Cada função relativa a um não-terminal precisa conter uma cláusula para cada produção
- A escolha da produção adequada é baseada no próximo *token*
- Isto é feito através da *predictive parsing table*
- Dada uma produção $X \rightarrow \gamma$
- Para cada terminal $T \in \text{FIRST}(\gamma)$
 - Coloque a produção $X \rightarrow \gamma$ na linha X , coluna T .
- Se γ é *nullable*:
 - Coloque a produção na linha X , coluna T para cada $T \in \text{FOLLOW}[X]$.

Exemplo

$Z \rightarrow d$

$Z \rightarrow X Y Z$

$Y \rightarrow$

$Y \rightarrow c$

$X \rightarrow Y$

$X \rightarrow a$

	nullable	FIRST	FOLLOW
X			
Y			
Z			

Exemplo

$Z \rightarrow d$


$Z \rightarrow X Y Z$

$Y \rightarrow$

$Y \rightarrow c$

$X \rightarrow Y$

$X \rightarrow a$



	nullable	FIRST	FOLLOW
X			
Y			
Z			

Exemplo

$Z \rightarrow d$


$Z \rightarrow X Y Z$

$Y \rightarrow$

$Y \rightarrow c$

$X \rightarrow Y$

$X \rightarrow a$



	nullable	FIRST	FOLLOW
X	sim		
Y			
Z			

Exemplo

$Z \rightarrow d$


$Z \rightarrow X Y Z$

$Y \rightarrow$

$Y \rightarrow c$

$X \rightarrow Y$

$X \rightarrow a$



	nullable	FIRST	FOLLOW
X	sim		
Y	sim		
Z			

Exemplo

$Z \rightarrow d$

$Z \rightarrow X Y Z$

$Y \rightarrow$

$Y \rightarrow c$

$X \rightarrow Y$

$X \rightarrow a$



	nullable	FIRST	FOLLOW
X	sim		
Y	sim		
Z	não		

Exemplo

$Z \rightarrow d$

$Z \rightarrow X Y Z$

$Y \rightarrow$

$Y \rightarrow c$

$X \rightarrow Y$

$X \rightarrow a$



	nullable	FIRST	FOLLOW
X	sim		
Y	sim		
Z	não		

Exemplo

$Z \rightarrow d$

$Z \rightarrow X Y Z$

$Y \rightarrow$

$Y \rightarrow c$

$X \rightarrow Y$

$X \rightarrow a$



	nullable	FIRST	FOLLOW
X	sim		
Y	sim		
Z	não	d	

Exemplo

$Z \rightarrow d$

$Z \rightarrow X Y Z$

$Y \rightarrow$

$Y \rightarrow c$

$X \rightarrow Y$

$X \rightarrow a$



	nullable	FIRST	FOLLOW
X	sim		
Y	sim	c	
Z	não	d	

Exemplo

$Z \rightarrow d$

$Z \rightarrow X Y Z$

$Y \rightarrow$

$Y \rightarrow c$

$X \rightarrow Y$

$X \rightarrow a$



	nullable	FIRST	FOLLOW
X	sim	c	
Y	sim	c	
Z	não	d	

Exemplo

$Z \rightarrow d$

$Z \rightarrow X Y Z$

$Y \rightarrow$

$Y \rightarrow c$

$X \rightarrow Y$

$X \rightarrow a$



	nullable	FIRST	FOLLOW
X	sim	$c a$	
Y	sim	c	
Z	não	d	

Exemplo

$Z \rightarrow d$

$Z \rightarrow X Y Z$

$Y \rightarrow$

$Y \rightarrow c$

$X \rightarrow Y$

$X \rightarrow a$



	nullable	FIRST	FOLLOW
X	sim	$c a$	
Y	sim	c	
Z	não	$d c a$	

Exemplo

$Z \rightarrow d$

$Z \rightarrow X Y Z$

$Y \rightarrow$

$Y \rightarrow c$

$X \rightarrow Y$

$X \rightarrow a$



	nullable	FIRST	FOLLOW
X	sim	c a	
Y	sim	c	
Z	não	d c a	

Exemplo

$Z \rightarrow d$

$Z \rightarrow X Y Z$

$Y \rightarrow$

$Y \rightarrow c$

$X \rightarrow Y$

$X \rightarrow a$



	nullable	FIRST	FOLLOW
X	sim	c a	c
Y	sim	c	
Z	não	d c a	

Exemplo

$Z \rightarrow d$

$Z \rightarrow X Y Z$

$Y \rightarrow$

$Y \rightarrow c$

$X \rightarrow Y$

$X \rightarrow a$



	nullable	FIRST	FOLLOW
X	sim	c a	c a d
Y	sim	c	
Z	não	d c a	

Exemplo

$Z \rightarrow d$

$Z \rightarrow X Y Z$

$Y \rightarrow$

$Y \rightarrow c$

$X \rightarrow Y$

$X \rightarrow a$



	nullable	FIRST	FOLLOW
X	sim	$c a$	$c a d$
Y	sim	c	$d c a$
Z	não	$d c a$	

Exemplo

$Z \rightarrow d$

$Z \rightarrow X Y Z$

$Y \rightarrow$

$Y \rightarrow c$

$X \rightarrow Y$

$X \rightarrow a$

	nullable	FIRST	FOLLOW
X	sim	a c	a c d
Y	sim	c	a c d
Z	não	a c d	

Exemplo

	nullable	FIRST	FOLLOW
$Z \rightarrow d$	sim	a c	a c d
$Z \rightarrow X Y Z$	sim	c	a c d
$Y \rightarrow$	não	a c d	
$Y \rightarrow c$			
$X \rightarrow Y$			
$X \rightarrow a$			

Construindo um Predictive Parser LL(1)

Criar uma tabela com:

- uma linha para cada não-terminal X
- uma coluna para cada terminal t

Para cada produção $X \rightarrow \gamma$

- Para cada terminal $t \in \text{FIRST}(\gamma)$
 - Coloque a produção $X \rightarrow \gamma$ na linha X , coluna t
- Se γ é *nullable*:
 - Coloque a produção na linha X , coluna t para cada $t \in \text{FOLLOW}[X]$.

Exemplo

$Z \rightarrow d$
 $Z \rightarrow X Y Z$
 $Y \rightarrow$
 $Y \rightarrow c$
 $X \rightarrow Y$
 $X \rightarrow a$

	nullable	FIRST	FOLLOW
X	sim	a c	a c d
Y	sim	c	a c d
Z	não	a c d	

	a	c	d
X			
Y			
Z			

Exemplo

$Z \rightarrow d$
 $Z \rightarrow X Y Z$
 $Y \rightarrow$
 $Y \rightarrow c$
 $X \rightarrow Y$
 $X \rightarrow a$

	nullable	FIRST	FOLLOW
X	sim	a c	a c d
Y	sim	c	a c d
Z	não	a c d	

	a	c	d
X			
Y			
Z			$Z \rightarrow d$

Exemplo

$Z \rightarrow d$
 $Z \rightarrow X Y Z$
 $Y \rightarrow$
 $Y \rightarrow c$
 $X \rightarrow Y$
 $X \rightarrow a$

	nullable	FIRST	FOLLOW
X	sim	a c	a c d
Y	sim	c	a c d
Z	não	a c d	

	a	c	d
X			
Y			
Z	$Z \rightarrow XYZ$	$Z \rightarrow XYZ$	$Z \rightarrow d$

Exemplo

$Z \rightarrow d$
 $Z \rightarrow X Y Z$
 $Y \rightarrow$
 $Y \rightarrow c$
 $X \rightarrow Y$
 $X \rightarrow a$

	nullable	FIRST	FOLLOW
X	sim	a c	a c d
Y	sim	c	a c d
Z	não	a c d	

	a	c	d
X			
Y			
Z	$Z \rightarrow XYZ$	$Z \rightarrow XYZ$	$Z \rightarrow d$ $Z \rightarrow XYZ$

Exemplo

$Z \rightarrow d$
 $Z \rightarrow X Y Z$
 $Y \rightarrow$
 $Y \rightarrow c$
 $X \rightarrow Y$
 $X \rightarrow a$

	nullable	FIRST	FOLLOW
X	sim	a c	a c d
Y	sim	c	a c d
Z	não	a c d	

	a	c	d
X			
Y	$Y \rightarrow$	$Y \rightarrow$	$Y \rightarrow$
Z	$Z \rightarrow XYZ$	$Z \rightarrow XYZ$	$Z \rightarrow d$ $Z \rightarrow XYZ$

Exemplo

$Z \rightarrow d$
 $Z \rightarrow X Y Z$
 $Y \rightarrow$
 $Y \rightarrow c$
 $X \rightarrow Y$
 $X \rightarrow a$

	nullable	FIRST	FOLLOW
X	sim	a c	a c d
Y	sim	c	a c d
Z	não	a c d	

	a	c	d
X			
Y	$Y \rightarrow$	$Y \rightarrow$ $Y \rightarrow c$	$Y \rightarrow$
Z	$Z \rightarrow XYZ$	$Z \rightarrow XYZ$	$Z \rightarrow d$ $Z \rightarrow XYZ$

Exemplo

$Z \rightarrow d$
 $Z \rightarrow X Y Z$
 $Y \rightarrow$
 $Y \rightarrow c$
 $X \rightarrow Y$
 $X \rightarrow a$

	nullable	FIRST	FOLLOW
X	sim	a c	a c d
Y	sim	c	a c d
Z	não	a c d	

	a	c	d
X		$X \rightarrow Y$	
Y	$Y \rightarrow$	$Y \rightarrow$ $Y \rightarrow c$	$Y \rightarrow$
Z	$Z \rightarrow XYZ$	$Z \rightarrow XYZ$	$Z \rightarrow d$ $Z \rightarrow XYZ$

Exemplo

$Z \rightarrow d$
 $Z \rightarrow X Y Z$
 $Y \rightarrow$
 $Y \rightarrow c$
 $X \rightarrow Y$
 $X \rightarrow a$

	nullable	FIRST	FOLLOW
X	sim	a c	a c d
Y	sim	c	a c d
Z	não	a c d	

	a	c	d
X	$X \rightarrow Y$	$X \rightarrow Y$	$X \rightarrow Y$
Y	$Y \rightarrow$	$Y \rightarrow$ $Y \rightarrow c$	$Y \rightarrow$
Z	$Z \rightarrow XYZ$	$Z \rightarrow XYZ$	$Z \rightarrow d$ $Z \rightarrow XYZ$

Exemplo

$Z \rightarrow d$
 $Z \rightarrow X Y Z$
 $Y \rightarrow$
 $Y \rightarrow c$
 $X \rightarrow Y$
 $X \rightarrow a$

	nullable	FIRST	FOLLOW
X	sim	a c	a c d
Y	sim	c	a c d
Z	não	a c d	

	a	c	d
X	$X \rightarrow a$ $X \rightarrow Y$	$X \rightarrow Y$	$X \rightarrow Y$
Y	$Y \rightarrow$	$Y \rightarrow$ $Y \rightarrow c$	$Y \rightarrow$
Z	$Z \rightarrow XYZ$	$Z \rightarrow XYZ$	$Z \rightarrow d$ $Z \rightarrow XYZ$

Exemplo

$Z \rightarrow d$
 $Z \rightarrow X Y Z$
 $Y \rightarrow$
 $Y \rightarrow c$
 $X \rightarrow Y$
 $X \rightarrow a$

	nullable	FIRST	FOLLOW
X	sim	a c	a c d
Y	sim	c	a c d
Z	não	a c d	

Funciona ???

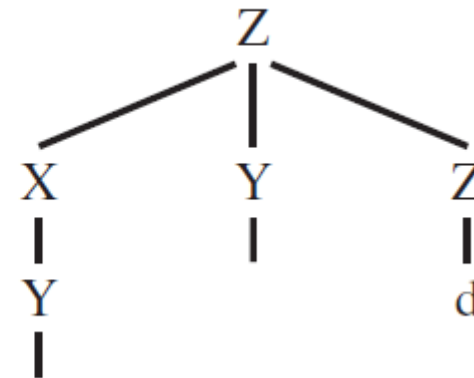
	a	c	d
X	$X \rightarrow a$ $X \rightarrow Y$	$X \rightarrow Y$	$X \rightarrow Y$
Y	$Y \rightarrow$	$Y \rightarrow$ $Y \rightarrow c$	$Y \rightarrow$
Z	$Z \rightarrow XYZ$	$Z \rightarrow XYZ$	$Z \rightarrow d$ $Z \rightarrow XYZ$

Construindo um Predictive Parser LL(1)

Não Funciona!! Por quê?

- A gramática é ambígua
- Note que algumas células da tabela do *predictive parser* têm mais de uma entrada!
- Isso sempre acontece com gramáticas ambíguas!

Z
|
d



Construindo um Predictive Parser LL(1)

- Linguagens cujas tabelas não possuam entradas duplicadas são denominadas de LL(1)
 - *Left to right parsing, leftmost derivation, 1-symbol lookahead*
- A definição de conjuntos FIRST pode ser generalizada para os primeiros k tokens de uma string
 - Gera-se uma tabela onde as linhas são os não-terminais e as colunas são todas as seqüências possíveis de k terminais
- Isso é raramente feito devido ao tamanho explosivo das tabelas geradas
- Gramáticas analisáveis com tabelas LL(k) são chamadas LL(k)
- Nenhuma gramática ambígua é LL(k) para nenhum k !

Exemplo: Gramática LL(1)

$$E \rightarrow TE'$$

$$E' \rightarrow +TE'$$

$$E' \rightarrow$$

$$T \rightarrow FT'$$

$$T' \rightarrow *FT'$$

$$T' \rightarrow$$

$$F \rightarrow (E)$$

$$F \rightarrow id$$

Exemplo: Gramática LL(1)

$S \rightarrow E\$$ ← inserção do fim de arquivo

$E \rightarrow TE'$

$E' \rightarrow +TE'$

$E' \rightarrow$

$T \rightarrow FT'$

$T' \rightarrow *FT'$

$T' \rightarrow$

$F \rightarrow (E)$

$F \rightarrow id$

Exemplo: Gramática LL(1)

$S \rightarrow E\$$

$E \rightarrow TE'$

$E' \rightarrow +TE'$

$E' \rightarrow$

$T \rightarrow FT'$

$T' \rightarrow *FT'$

$T' \rightarrow$

$F \rightarrow (E)$

$F \rightarrow id$

	Nullable	FIRST	FOLLOW
E	N	(id) \$
E'	S	+) \$
T	N	(id	+) \$
T'	S	*	+) \$
F	N	(id	* +) \$
S	N	(id	

Exemplo: Gramática LL(1)

$$S \rightarrow E\$$$

$$E \rightarrow TE'$$

$$E' \rightarrow +TE'$$

$$E' \rightarrow$$

$$T \rightarrow FT'$$

$$T' \rightarrow *FT'$$

$$T' \rightarrow$$

$$F \rightarrow (E)$$

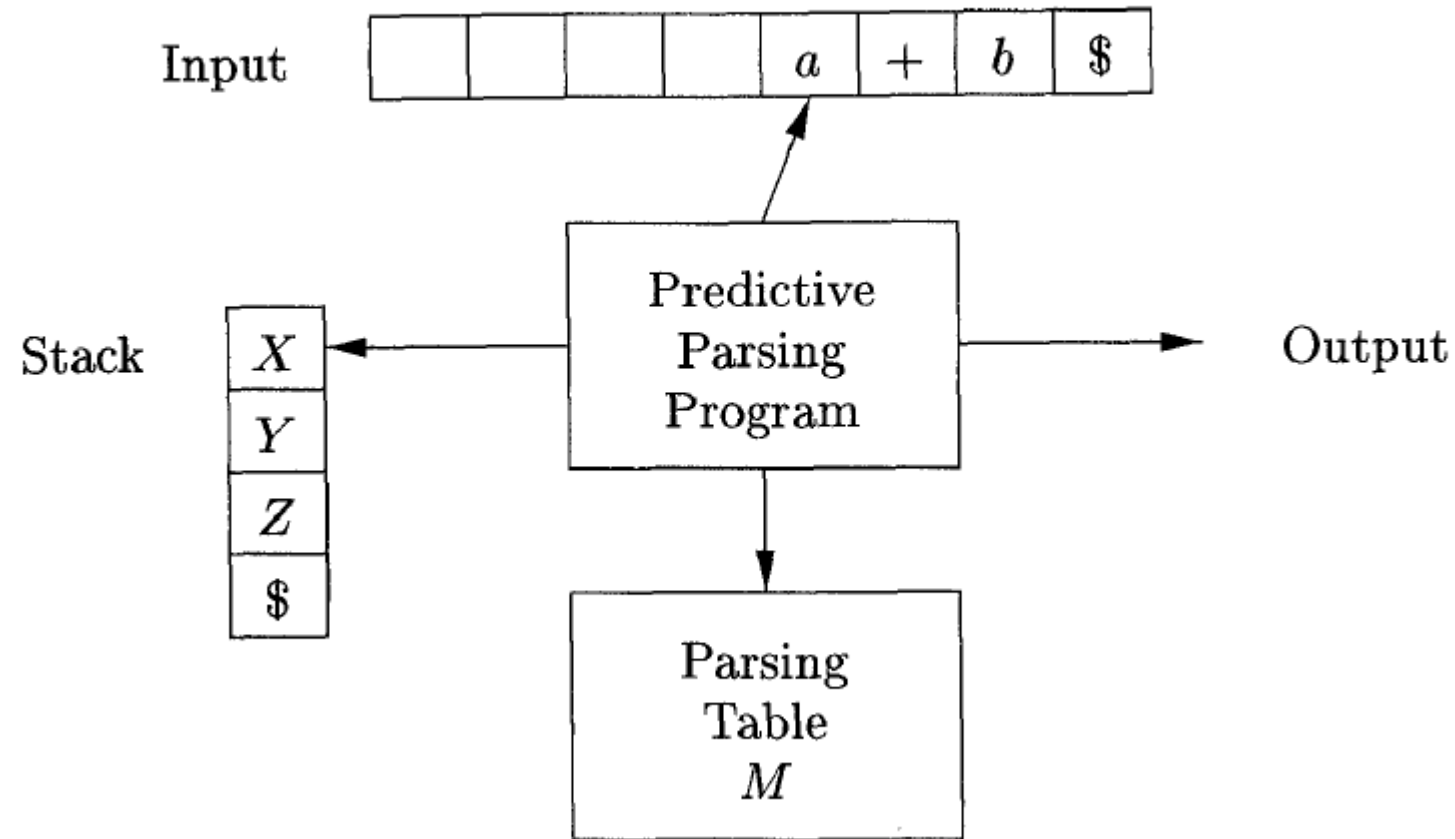
$$F \rightarrow id$$

	Nullable	FIRST	FOLLOW
E	N	(id) \$
E'	S	+) \$
T	N	(id	+) \$
T'	S	*	+) \$
F	N	(id	* +) \$
S	N	(id	

NON - TERMINAL	INPUT SYMBOL					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow$	$E' \rightarrow$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow$	$T' \rightarrow *FT'$		$T' \rightarrow$	$T' \rightarrow$
F	$F \rightarrow id$			$F \rightarrow (E)$		
S	$S \rightarrow E\$$			$S \rightarrow E\$$		

Análise Sintática LL(1)

TOP-DOWN PARSING



Análise Sintática LL(1)

$$S \rightarrow E\$$$

$$E \rightarrow TE'$$

$$E' \rightarrow +TE'$$

$$E' \rightarrow$$

$$T \rightarrow FT'$$

$$T' \rightarrow *FT'$$

$$T' \rightarrow$$

$$F \rightarrow (E)$$

$$F \rightarrow id$$

NON - TERMINAL	INPUT SYMBOL					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow$	$E' \rightarrow$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow$	$T' \rightarrow *FT'$		$T' \rightarrow$	$T' \rightarrow$
F	$F \rightarrow id$			$F \rightarrow (E)$		
S	$S \rightarrow E\$$			$S \rightarrow E\$$		

A cadeia abaixo pertence a linguagem gerada pela gramática?

id+id*id\$

Exemplo: Gramática LL(1)

Matched	Stack	Input	Action
		s	id + id * id \$

NON - TERMINAL	INPUT SYMBOL					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow$	$E' \rightarrow$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow$	$T' \rightarrow *FT'$		$T' \rightarrow$	$T' \rightarrow$
F	$F \rightarrow \text{id}$			$F \rightarrow (E)$		
S	$S \rightarrow ES$			$S \rightarrow ES$		

Exemplo: Gramática LL(1)

Matched	Stack	Input	Action
	S	id + id * id \$	
	E\$	id + id * id \$	S -> E\$

NON - TERMINAL	INPUT SYMBOL					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow$	$E' \rightarrow$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow$	$T' \rightarrow *FT'$		$T' \rightarrow$	$T' \rightarrow$
F	$F \rightarrow \text{id}$			$F \rightarrow (E)$		
S	$S \rightarrow ES$			$S \rightarrow ES$		

Exemplo: Gramática LL(1)

NON - TERMINAL	INPUT SYMBOL					
	id	+	*	()	\$
<i>E</i>	$E \rightarrow TE'$			$E \rightarrow TE'$		
<i>E'</i>		$E' \rightarrow +TE'$			$E' \rightarrow$	$E' \rightarrow$
<i>T</i>	$T \rightarrow FT'$			$T \rightarrow FT'$		
<i>T'</i>		$T' \rightarrow$	$T' \rightarrow *FT'$		$T' \rightarrow$	$T' \rightarrow$
<i>F</i>	$F \rightarrow \text{id}$			$F \rightarrow (E)$		
<i>S</i>	$S \rightarrow ES$			$S \rightarrow ES$		

Matched	Stack	Input	Action
	S	id + id * id \$	
	E\$	id + id * id \$	S -> E\$
	TE' \$	id + id * id \$	E -> TE'

Exemplo: Gramática LL(1)

NON - TERMINAL	INPUT SYMBOL					
	id	+	*	()	\$
<i>E</i>	$E \rightarrow TE'$			$E \rightarrow TE'$		
<i>E'</i>		$E' \rightarrow +TE'$			$E' \rightarrow$	$E' \rightarrow$
<i>T</i>	$T \rightarrow FT'$			$T \rightarrow FT'$		
<i>T'</i>		$T' \rightarrow$	$T' \rightarrow *FT'$		$T' \rightarrow$	$T' \rightarrow$
<i>F</i>	$F \rightarrow \text{id}$			$F \rightarrow (E)$		
<i>S</i>	$S \rightarrow ES$			$S \rightarrow ES$		

Matched	Stack	Input	Action
	S	id + id * id \$	
	E\$	id + id * id \$	S -> E\$
	TE'\$	id + id * id \$	E -> TE'
	FT'E'\$	id + id * id \$	T -> FT'

Exemplo: Gramática LL(1)

NON - TERMINAL	INPUT SYMBOL					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow$	$E' \rightarrow$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow$	$T' \rightarrow *FT'$		$T' \rightarrow$	$T' \rightarrow$
F	$F \rightarrow \text{id}$			$F \rightarrow (E)$		
S	$S \rightarrow ES$			$S \rightarrow ES$		

Matched	Stack	Input	Action
	S	id + id * id \$	
	E\$	id + id * id \$	S -> E\$
	TE'\$	id + id * id \$	E -> TE'
	FT'E'\$	id + id * id \$	T -> FT'
	idT'E'\$	id + id * id \$	F -> id

Exemplo: Gramática LL(1)

NON - TERMINAL	INPUT SYMBOL					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow$	$E' \rightarrow$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow$	$T' \rightarrow *FT'$		$T' \rightarrow$	$T' \rightarrow$
F	$F \rightarrow \text{id}$			$F \rightarrow (E)$		
S	$S \rightarrow ES$			$S \rightarrow ES$		

Matched	Stack	Input	Action
	S	id + id * id \$	
	E\$	id + id * id \$	S -> E\$
	TE'\$	id + id * id \$	E -> TE'
	FT'E'\$	id + id * id \$	T -> FT'
id	idT'E'\$	id + id * id \$	F -> id
	T'E'\$	+ id * id \$	match id

Exemplo: Gramática LL(1)

NON - TERMINAL	INPUT SYMBOL					
	id	+	*	()	\$
<i>E</i>	$E \rightarrow TE'$			$E \rightarrow TE'$		
<i>E'</i>		$E' \rightarrow +TE'$			$E' \rightarrow$	$E' \rightarrow$
<i>T</i>	$T \rightarrow FT'$			$T \rightarrow FT'$		
<i>T'</i>		$T' \rightarrow$	$T' \rightarrow *FT'$		$T' \rightarrow$	$T' \rightarrow$
<i>F</i>	$F \rightarrow \text{id}$			$F \rightarrow (E)$		
<i>S</i>	$S \rightarrow ES$			$S \rightarrow ES$		

Matched	Stack	Input	Action
	S	id + id * id \$	
	E\$	id + id * id \$	S -> E\$
	TE'\$	id + id * id \$	E -> TE'
	FT'E'\$	id + id * id \$	T -> FT'
id	idT'E'\$	id + id * id \$	F -> id
id	T'E'\$	+ id * id \$	match id
	E'\$	+ id * id \$	T' ->

Exemplo: Gramática LL(1)

NON - TERMINAL	INPUT SYMBOL					
	id	+	*	()	\$
<i>E</i>	$E \rightarrow TE'$			$E \rightarrow TE'$		
<i>E'</i>		$E' \rightarrow +TE'$			$E' \rightarrow$	$E' \rightarrow$
<i>T</i>	$T \rightarrow FT'$			$T \rightarrow FT'$		
<i>T'</i>		$T' \rightarrow$	$T' \rightarrow *FT'$		$T' \rightarrow$	$T' \rightarrow$
<i>F</i>	$F \rightarrow \text{id}$			$F \rightarrow (E)$		
<i>S</i>	$S \rightarrow ES$			$S \rightarrow ES$		

Matched	Stack	Input	Action
	S	id + id * id \$	
	E\$	id + id * id \$	S -> E\$
	TE'\$	id + id * id \$	E -> TE'
	FT'E'\$	id + id * id \$	T -> FT'
	idTE'\$	id + id * id \$	F -> id
id	T'E'\$	+ id * id \$	match id
id	E'\$	+ id * id \$	T' ->
id	+TE'\$	+ id * id \$	E' -> +TE'

Exemplo: Gramática LL(1)

NON - TERMINAL	INPUT SYMBOL					
	id	+	*	()	\$
<i>E</i>	$E \rightarrow TE'$			$E \rightarrow TE'$		
<i>E'</i>		$E' \rightarrow +TE'$			$E' \rightarrow$	$E' \rightarrow$
<i>T</i>	$T \rightarrow FT'$			$T \rightarrow FT'$		
<i>T'</i>		$T' \rightarrow$	$T' \rightarrow *FT'$		$T' \rightarrow$	$T' \rightarrow$
<i>F</i>	$F \rightarrow \text{id}$			$F \rightarrow (E)$		
<i>S</i>	$S \rightarrow ES$			$S \rightarrow ES$		

Matched	Stack	Input	Action
	S	id + id * id \$	
	E\$	id + id * id \$	S -> E\$
	TE'\$	id + id * id \$	E -> TE'
	FT'E'\$	id + id * id \$	T -> FT'
	idTE'\$	id + id * id \$	F -> id
id	T'E'\$	+ id * id \$	match id
id	E'\$	+ id * id \$	T' ->
id	+TE'\$	+ id * id \$	E' -> +TE'
id +	TE'\$	id * id \$	match +

Exemplo: Gramática LL(1)

NON - TERMINAL	INPUT SYMBOL					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow$	$E' \rightarrow$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow$	$T' \rightarrow *FT'$		$T' \rightarrow$	$T' \rightarrow$
F	$F \rightarrow \text{id}$			$F \rightarrow (E)$		
S	$S \rightarrow ES$			$S \rightarrow ES$		

Matched	Stack	Input	Action
	S	id + id * id \$	
	E\$	id + id * id \$	S -> E\$
	TE'\$	id + id * id \$	E -> TE'
	FT'E'\$	id + id * id \$	T -> FT'
	idTE'\$	id + id * id \$	F -> id
id	T'E'\$	+ id * id \$	match id
id	E'\$	+ id * id \$	T' ->
id	+TE'\$	+ id * id \$	E' -> +TE'
id +	TE'\$	id * id \$	match +
id +	FT'E'\$	id * id \$	T -> FT'

Exemplo: Gramática LL(1)

NON - TERMINAL	INPUT SYMBOL					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow$	$E' \rightarrow$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow$	$T' \rightarrow *FT'$		$T' \rightarrow$	$T' \rightarrow$
F	$F \rightarrow \text{id}$			$F \rightarrow (E)$		
S	$S \rightarrow ES$			$S \rightarrow ES$		

Matched	Stack	Input	Action
	S	id + id * id \$	
	E\$	id + id * id \$	S -> E\$
	TE'\$	id + id * id \$	E -> TE'
	FT'E'\$	id + id * id \$	T -> FT'
	idTE'\$	id + id * id \$	F -> id
id	T'E'\$	+ id * id \$	match id
id	E'\$	+ id * id \$	T' ->
id	+TE'\$	+ id * id \$	E' -> +TE'
id +	TE'\$	id * id \$	match +
id +	FT'E'\$	id * id \$	T -> FT'
id +	idTE'\$	id * id \$	F -> id

Exemplo: Gramática LL(1)

NON - TERMINAL	INPUT SYMBOL					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow$	$E' \rightarrow$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow$	$T' \rightarrow *FT'$		$T' \rightarrow$	$T' \rightarrow$
F	$F \rightarrow \text{id}$			$F \rightarrow (E)$		
S	$S \rightarrow ES$			$S \rightarrow ES$		

Matched	Stack	Input	Action
	S	id + id * id \$	
	E\$	id + id * id \$	S -> E\$
	TE'\$	id + id * id \$	E -> TE'
	FT'E'\$	id + id * id \$	T -> FT'
	idTE'\$	id + id * id \$	F -> id
id	T'E'\$	+ id * id \$	match id
id	E'\$	+ id * id \$	T' ->
id	+TE'\$	+ id * id \$	E' -> +TE'
id +	TE'\$	id * id \$	match +
id +	FT'E'\$	id * id \$	T -> FT'
id +	idTE'\$	id * id \$	F -> id
id + id	T'E'\$	* id \$	match id

Exemplo: Gramática LL(1)

NON - TERMINAL	INPUT SYMBOL					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow$	$E' \rightarrow$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow$	$T' \rightarrow *FT'$		$T' \rightarrow$	$T' \rightarrow$
F	$F \rightarrow \text{id}$			$F \rightarrow (E)$		
S	$S \rightarrow ES$			$S \rightarrow ES$		

Matched	Stack	Input	Action
	S	id + id * id \$	
	E\$	id + id * id \$	S -> E\$
	TE'\$	id + id * id \$	E -> TE'
	FT'E'\$	id + id * id \$	T -> FT'
	idTE'\$	id + id * id \$	F -> id
id	T'E'\$	+ id * id \$	match id
id	E'\$	+ id * id \$	T' ->
id	+TE'\$	+ id * id \$	E' -> +TE'
id +	TE'\$	id * id \$	match +
id +	FT'E'\$	id * id \$	T -> FT'
id +	idTE'\$	id * id \$	F -> id
id + id	T'E'\$	* id \$	match id
id + id	*FT'E'\$	* id \$	T' -> *FT'

Exemplo: Gramática LL(1)

NON - TERMINAL	INPUT SYMBOL					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow$	$E' \rightarrow$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow$	$T' \rightarrow *FT'$		$T' \rightarrow$	$T' \rightarrow$
F	$F \rightarrow \text{id}$			$F \rightarrow (E)$		
S	$S \rightarrow ES$			$S \rightarrow ES$		

Matched	Stack	Input	Action
	S	id + id * id \$	
	E\$	id + id * id \$	S -> E\$
	TE'\$	id + id * id \$	E -> TE'
	FT'E'\$	id + id * id \$	T -> FT'
	idTE'\$	id + id * id \$	F -> id
id	T'E'\$	+ id * id \$	match id
id	E'\$	+ id * id \$	T' ->
id	+TE'\$	+ id * id \$	E' -> +TE'
id +	TE'\$	id * id \$	match +
id +	FT'E'\$	id * id \$	T -> FT'
id +	idTE'\$	id * id \$	F -> id
id + id	T'E'\$	* id \$	match id
id + id	*FT'E'\$	* id \$	T' -> *FT'
id + id *	FT'E'\$	id \$	match *

Exemplo: Gramática LL(1)

NON - TERMINAL	INPUT SYMBOL					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow$	$E' \rightarrow$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow$	$T' \rightarrow *FT'$		$T' \rightarrow$	$T' \rightarrow$
F	$F \rightarrow \text{id}$			$F \rightarrow (E)$		
S	$S \rightarrow ES$			$S \rightarrow ES$		

Matched	Stack	Input	Action
	S	id + id * id \$	
	E\$	id + id * id \$	S -> E\$
	TE'\$	id + id * id \$	E -> TE'
	FT'E'\$	id + id * id \$	T -> FT'
	idTE'\$	id + id * id \$	F -> id
id	T'E'\$	+ id * id \$	match id
id	E'\$	+ id * id \$	T' ->
id	+TE'\$	+ id * id \$	E' -> +TE'
id +	TE'\$	id * id \$	match +
id +	FT'E'\$	id * id \$	T -> FT'
id +	idTE'\$	id * id \$	F -> id
id + id	T'E'\$	* id \$	match id
id + id	*FT'E'\$	* id \$	T' -> *FT'
id + id *	FT'E'\$	id \$	match *
id + id *	idTE'\$	id \$	F -> id

Exemplo: Gramática LL(1)

NON - TERMINAL	INPUT SYMBOL					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow$	$E' \rightarrow$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow$	$T' \rightarrow *FT'$		$T' \rightarrow$	$T' \rightarrow$
F	$F \rightarrow \text{id}$			$F \rightarrow (E)$		
S	$S \rightarrow ES$			$S \rightarrow ES$		

Matched	Stack	Input	Action
	S	id + id * id \$	
	E\$	id + id * id \$	S -> E\$
	TE'\$	id + id * id \$	E -> TE'
	FT'E'\$	id + id * id \$	T -> FT'
	idTE'\$	id + id * id \$	F -> id
id	T'E'\$	+ id * id \$	match id
id	E'\$	+ id * id \$	T' ->
id	+TE'\$	+ id * id \$	E' -> +TE'
id +	TE'\$	id * id \$	match +
id +	FT'E'\$	id * id \$	T -> FT'
id +	idTE'\$	id * id \$	F -> id
id + id	T'E'\$	* id \$	match id
id + id	*FT'E'\$	* id \$	T' -> *FT'
id + id *	FT'E'\$	id \$	match *
id + id *	idTE'\$	id \$	F -> id
id + id * id	T'E'\$	\$	match id

Exemplo: Gramática LL(1)

NON - TERMINAL	INPUT SYMBOL					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow$	$E' \rightarrow$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow$	$T' \rightarrow *FT'$		$T' \rightarrow$	$T' \rightarrow$
F	$F \rightarrow \text{id}$			$F \rightarrow (E)$		
S	$S \rightarrow ES$			$S \rightarrow ES$		

Matched	Stack	Input	Action
	S	id + id * id \$	
	E\$	id + id * id \$	S -> E\$
	TE'\$	id + id * id \$	E -> TE'
	FT'E'\$	id + id * id \$	T -> FT'
	idT'E'\$	id + id * id \$	F -> id
id	T'E'\$	+ id * id \$	match id
id	E'\$	+ id * id \$	T' ->
id	+TE'\$	+ id * id \$	E' -> +TE'
id +	TE'\$	id * id \$	match +
id +	FT'E'\$	id * id \$	T -> FT'
id +	idT'E'\$	id * id \$	F -> id
id + id	T'E'\$	* id \$	match id
id + id	*FT'E'\$	* id \$	T' -> *FT'
id + id *	FT'E'\$	id \$	match *
id + id *	idT'E'\$	id \$	F -> id
id + id * id	T'E'\$	\$	match id
id + id * id	E'\$	\$	T' ->

Exemplo: Gramática LL(1)

NON - TERMINAL	INPUT SYMBOL					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow$	$E' \rightarrow$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow$	$T' \rightarrow *FT'$		$T' \rightarrow$	$T' \rightarrow$
F	$F \rightarrow \text{id}$			$F \rightarrow (E)$		
S	$S \rightarrow ES$			$S \rightarrow ES$		

Matched	Stack	Input	Action
	S	id + id * id \$	
	E\$	id + id * id \$	S -> E\$
	TE'\$	id + id * id \$	E -> TE'
	FT'E'\$	id + id * id \$	T -> FT'
	idT'E'\$	id + id * id \$	F -> id
id	T'E'\$	+ id * id \$	match id
id	E'\$	+ id * id \$	T' ->
id	+TE'\$	+ id * id \$	E' -> +TE'
id +	TE'\$	id * id \$	match +
id +	FT'E'\$	id * id \$	T -> FT'
id +	idT'E'\$	id * id \$	F -> id
id + id	T'E'\$	* id \$	match id
id + id	*FT'E'\$	* id \$	T' -> *FT'
id + id *	FT'E'\$	id \$	match *
id + id *	idT'E'\$	id \$	F -> id
id + id * id	T'E'\$	\$	match id
id + id * id	E'\$	\$	T' ->
id + id * id	\$	\$	E' ->

Exemplo: Gramática LL(1)

NON - TERMINAL	INPUT SYMBOL					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow$	$E' \rightarrow$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow$	$T' \rightarrow *FT'$		$T' \rightarrow$	$T' \rightarrow$
F	$F \rightarrow id$			$F \rightarrow (E)$		
S	$S \rightarrow ES$			$S \rightarrow ES$		

Matched	Stack	Input	Action
	S	id + id * id \$	
	E\$	id + id * id \$	S -> E\$
	TE'\$	id + id * id \$	E -> TE'
	FT'E'\$	id + id * id \$	T -> FT'
	idT'E'\$	id + id * id \$	F -> id
id	T'E'\$	+ id * id \$	match id
id	E'\$	+ id * id \$	T' ->
id	+TE'\$	+ id * id \$	E' -> +TE'
id +	TE'\$	id * id \$	match +
id +	FT'E'\$	id * id \$	T -> FT'
id +	idT'E'\$	id * id \$	F -> id
id + id	T'E'\$	* id \$	match id
id + id	*FT'E'\$	* id \$	T' -> *FT'
id + id *	FT'E'\$	id \$	match *
id + id *	idT'E'\$	id \$	F -> id
id + id * id	T'E'\$	\$	match id
id + id * id	E'\$	\$	T' ->
id + id * id	\$	\$	E' ->
id + id * id \$			match \$

Exemplo: Gramática LL(1)

NON - TERMINAL	INPUT SYMBOL					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow$	$E' \rightarrow$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow$	$T' \rightarrow *FT'$		$T' \rightarrow$	$T' \rightarrow$
F	$F \rightarrow id$			$F \rightarrow (E)$		
S	$S \rightarrow ES$			$S \rightarrow ES$		

Matched	Stack	Input	Action
	S	id + id * id \$	
	E\$	id + id * id \$	S -> E\$
	TE'\$	id + id * id \$	E -> TE'
	FT'E'\$	id + id * id \$	T -> FT'
	idT'E'\$	id + id * id \$	F -> id
id	T'E'\$	+ id * id \$	match id
id	E'\$	+ id * id \$	T' ->
id	+TE'\$	+ id * id \$	E' -> +TE'
id +	TE'\$	id * id \$	match +
id +	FT'E'\$	id * id \$	T -> FT'
id +	idT'E'\$	id * id \$	F -> id
id + id	T'E'\$	* id \$	match id
id + id	*FT'E'\$	* id \$	T' -> *FT'
id + id *	FT'E'\$	id \$	match *
id + id *	idT'E'\$	id \$	F -> id
id + id * id	T'E'\$	\$	match id
id + id * id	E'\$	\$	T' ->
id + id * id	\$	\$	E' ->
id + id * id \$			match \$

✓ Cadeia Aceita

Análise Sintática LL(1)

$$S \rightarrow E\$$$

$$E \rightarrow TE'$$

$$E' \rightarrow +TE'$$

$$E' \rightarrow$$

$$T \rightarrow FT'$$

$$T' \rightarrow *FT'$$

$$T' \rightarrow$$

$$F \rightarrow (E)$$

$$F \rightarrow id$$

NON - TERMINAL	INPUT SYMBOL					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow$	$E' \rightarrow$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow$	$T' \rightarrow *FT'$		$T' \rightarrow$	$T' \rightarrow$
F	$F \rightarrow id$			$F \rightarrow (E)$		
S	$S \rightarrow E\$$			$S \rightarrow E\$$		

A cadeia abaixo pertence a linguagem gerada pela gramática?

id+\$

Exemplo: Gramática LL(1)

Matched	Stack	Input	Action
		s	id + \$

NON - TERMINAL	INPUT SYMBOL					
	id	+	*	()	\$
<i>E</i>	$E \rightarrow TE'$			$E \rightarrow TE'$		
<i>E'</i>		$E' \rightarrow +TE'$			$E' \rightarrow$	$E' \rightarrow$
<i>T</i>	$T \rightarrow FT'$			$T \rightarrow FT'$		
<i>T'</i>		$T' \rightarrow$	$T' \rightarrow *FT'$		$T' \rightarrow$	$T' \rightarrow$
<i>F</i>	$F \rightarrow \text{id}$			$F \rightarrow (E)$		
<i>S</i>	$S \rightarrow ES$			$S \rightarrow ES$		

Exemplo: Gramática LL(1)

Matched	Stack	Input	Action
		S	
	E\$	id + \$	S -> E\$

NON - TERMINAL	INPUT SYMBOL					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow$	$E' \rightarrow$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow$	$T' \rightarrow *FT'$		$T' \rightarrow$	$T' \rightarrow$
F	$F \rightarrow \text{id}$			$F \rightarrow (E)$		
S	$S \rightarrow ES$			$S \rightarrow ES$		

Exemplo: Gramática LL(1)

NON - TERMINAL	INPUT SYMBOL					
	id	+	*	()	\$
<i>E</i>	$E \rightarrow TE'$			$E \rightarrow TE'$		
<i>E'</i>		$E' \rightarrow +TE'$			$E' \rightarrow$	$E' \rightarrow$
<i>T</i>	$T \rightarrow FT'$			$T \rightarrow FT'$		
<i>T'</i>		$T' \rightarrow$	$T' \rightarrow *FT'$		$T' \rightarrow$	$T' \rightarrow$
<i>F</i>	$F \rightarrow \text{id}$			$F \rightarrow (E)$		
<i>S</i>	$S \rightarrow ES$			$S \rightarrow ES$		

Matched	Stack	Input	Action
	S	id + \$	
	E\$	id + \$	S -> E\$
	TE' \$	id + \$	E -> TE'

Exemplo: Gramática LL(1)

NON - TERMINAL	INPUT SYMBOL					
	id	+	*	()	\$
<i>E</i>	$E \rightarrow TE'$			$E \rightarrow TE'$		
<i>E'</i>		$E' \rightarrow +TE'$			$E' \rightarrow$	$E' \rightarrow$
<i>T</i>	$T \rightarrow FT'$			$T \rightarrow FT'$		
<i>T'</i>		$T' \rightarrow$	$T' \rightarrow *FT'$		$T' \rightarrow$	$T' \rightarrow$
<i>F</i>	$F \rightarrow \text{id}$			$F \rightarrow (E)$		
<i>S</i>	$S \rightarrow ES$			$S \rightarrow ES$		

Matched	Stack	Input	Action
	S	id + \$	
	E\$	id + \$	S -> E\$
	TE'\$	id + \$	E -> TE'
	FT'E'\$	id + \$	T -> FT'

Exemplo: Gramática LL(1)

NON - TERMINAL	INPUT SYMBOL					
	id	+	*	()	\$
<i>E</i>	$E \rightarrow TE'$			$E \rightarrow TE'$		
<i>E'</i>		$E' \rightarrow +TE'$			$E' \rightarrow$	$E' \rightarrow$
<i>T</i>	$T \rightarrow FT'$			$T \rightarrow FT'$		
<i>T'</i>		$T' \rightarrow$	$T' \rightarrow *FT'$		$T' \rightarrow$	$T' \rightarrow$
<i>F</i>	$F \rightarrow \text{id}$			$F \rightarrow (E)$		
<i>S</i>	$S \rightarrow ES$			$S \rightarrow ES$		

Matched	Stack	Input	Action
	S	id + \$	
	E\$	id + \$	S -> E\$
	TE'\$	id + \$	E -> TE'
	FT'E'\$	id + \$	T -> FT'
	idT'E'\$	id + \$	F -> id

Exemplo: Gramática LL(1)

NON - TERMINAL	INPUT SYMBOL					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow$	$E' \rightarrow$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow$	$T' \rightarrow *FT'$		$T' \rightarrow$	$T' \rightarrow$
F	$F \rightarrow \text{id}$			$F \rightarrow (E)$		
S	$S \rightarrow ES$			$S \rightarrow ES$		

Matched	Stack	Input	Action
	S	id + \$	
	E\$	id + \$	S -> E\$
	TE'\$	id + \$	E -> TE'
	FT'E'\$	id + \$	T -> FT'
id	idT'E'\$	id + \$	F -> id
	T'E'\$	+ \$	match id

Exemplo: Gramática LL(1)

NON - TERMINAL	INPUT SYMBOL					
	id	+	*	()	\$
<i>E</i>	$E \rightarrow TE'$			$E \rightarrow TE'$		
<i>E'</i>		$E' \rightarrow +TE'$			$E' \rightarrow$	$E' \rightarrow$
<i>T</i>	$T \rightarrow FT'$			$T \rightarrow FT'$		
<i>T'</i>		$T' \rightarrow$	$T' \rightarrow *FT'$		$T' \rightarrow$	$T' \rightarrow$
<i>F</i>	$F \rightarrow \text{id}$			$F \rightarrow (E)$		
<i>S</i>	$S \rightarrow ES$			$S \rightarrow ES$		

Matched	Stack	Input	Action
	S	id + \$	
	E\$	id + \$	S -> E\$
	TE'\$	id + \$	E -> TE'
	FT'E'\$	id + \$	T -> FT'
id	idT'E'\$	id + \$	F -> id
id	T'E'\$	+ \$	match id
	E'\$	+ \$	T' ->

Exemplo: Gramática LL(1)

NON - TERMINAL	INPUT SYMBOL					
	id	+	*	()	\$
<i>E</i>	$E \rightarrow TE'$			$E \rightarrow TE'$		
<i>E'</i>		$E' \rightarrow +TE'$			$E' \rightarrow$	$E' \rightarrow$
<i>T</i>	$T \rightarrow FT'$			$T \rightarrow FT'$		
<i>T'</i>		$T' \rightarrow$	$T' \rightarrow *FT'$		$T' \rightarrow$	$T' \rightarrow$
<i>F</i>	$F \rightarrow \text{id}$			$F \rightarrow (E)$		
<i>S</i>	$S \rightarrow ES$			$S \rightarrow ES$		

Matched	Stack	Input	Action
	S	id + \$	
	E\$	id + \$	S -> E\$
	TE'\$	id + \$	E -> TE'
	FT'E'\$	id + \$	T -> FT'
	idTE'\$	id + \$	F -> id
id	T'E'\$	+ \$	match id
id	E'\$	+ \$	T' ->
id	+TE'\$	+ \$	E' -> +TE'

Exemplo: Gramática LL(1)

NON - TERMINAL	INPUT SYMBOL					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow$	$E' \rightarrow$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow$	$T' \rightarrow *FT'$		$T' \rightarrow$	$T' \rightarrow$
F	$F \rightarrow \text{id}$			$F \rightarrow (E)$		
S	$S \rightarrow ES$			$S \rightarrow ES$		

Matched	Stack	Input	Action
	S	$\text{id } + \$$	
	$E\$$	$\text{id } + \$$	$S \rightarrow E\$$
	$TE'\$$	$\text{id } + \$$	$E \rightarrow TE'$
	$FT'E'\$$	$\text{id } + \$$	$T \rightarrow FT'$
	$\text{id}TE'\$$	$\text{id } + \$$	$F \rightarrow \text{id}$
id	$T'E'\$$	$+ \$$	match id
id	$E'\$$	$+ \$$	$T' \rightarrow$
id	$+TE'\$$	$+ \$$	$E' \rightarrow +TE'$
$\text{id } +$	$TE'\$$	$\$$	match $+$

Exemplo: Gramática LL(1)

NON - TERMINAL	INPUT SYMBOL					
	id	+	*	()	\$
<i>E</i>	$E \rightarrow TE'$			$E \rightarrow TE'$		
<i>E'</i>		$E' \rightarrow +TE'$			$E' \rightarrow$	$E' \rightarrow$
<i>T</i>	$T \rightarrow FT'$			$T \rightarrow FT'$		
<i>T'</i>		$T' \rightarrow$	$T' \rightarrow *FT'$		$T' \rightarrow$	$T' \rightarrow$
<i>F</i>	$F \rightarrow \text{id}$			$F \rightarrow (E)$		
<i>S</i>	$S \rightarrow ES$			$S \rightarrow ES$		

Matched	Stack	Input	Action
	S	id + \$	
	E\$	id + \$	S -> E\$
	TE'\$	id + \$	E -> TE'
	FT'E'\$	id + \$	T -> FT'
	idTE'\$	id + \$	F -> id
id	T'E'\$	+ \$	match id
id	E'\$	+ \$	T' ->
id	+TE'\$	+ \$	E' -> +TE'
id +	TE'\$	\$	match +

✗ Cadeia Recusada

Lista de Exercícios

Lista 10

- Exercícios teóricos e de implementação