

Compiladores

(IF688)

Leopoldo Teixeira
imt@cin.ufpe.br | [@leopoldomt](https://twitter.com/leopoldomt)

Gramáticas LL(1)

- Parsers preditivos podem ser criados para classe de gramáticas LL(1)
- Esta classe é rica o suficiente para cobrir a maioria das construções de linguagens de programação
- Temos de tomar cuidado especial ao escrever tais gramáticas, pois gramáticas com ambiguidade e recursão à esquerda não são LL(1)

Gramática LL(1)

- G é LL(1) sse, quando $A \rightarrow \alpha \mid \beta$ são duas produções distintas de G e as condições abaixo são satisfeitas:
 - Para nenhum terminal a , α e β geram palavras iniciadas em a
 - No máximo um de α e β deriva a palavra vazia
 - Se β gera ϵ por meio de derivação sucessiva, α não pode derivar palavras iniciadas com terminais de $\text{FOLLOW}(A)$
 - Se α gera ϵ por meio de derivação sucessiva, β não pode derivar palavras iniciadas com terminais de $\text{FOLLOW}(A)$

Parsing Tables

- Vários métodos de parsing usam tabelas de referência para decidir qual produção aplicar
- Além disso, podemos utilizar tais tabelas para demonstrar que uma gramática não é LL(1)
- Com base nos conjuntos FIRST e FOLLOW, construímos a tabela $M[A, \mathbf{a}]$, onde A é um não terminal e \mathbf{a} um símbolo terminal (incluindo \$)

Parsing Table

$S \rightarrow \mathbf{aABe}$

$A \rightarrow \mathbf{bK}$

$K \rightarrow \mathbf{bcK}$

$K \rightarrow \boldsymbol{\varepsilon}$

$B \rightarrow \mathbf{d}$

	FIRST	FOLLOW
S	{ a }	{ \$ }
A	{ b }	{ d }
K	{ b, ε }	{ d }
B	{ d }	{ e }

	a	b	c	d	e	\$
S						
A						
K						
B						

Predictive Parsing Table

- Para cada produção $A \rightarrow \alpha$ de G
 - Para todo $\mathbf{a} \in \text{FIRST}(\alpha)$, adicione $A \rightarrow \alpha$ em $M[A, \mathbf{a}]$
 - Se $\epsilon \in \text{FIRST}(\alpha)$, então, para todo $\mathbf{b} \in \text{FOLLOW}(A)$, adicione $A \rightarrow \alpha$ em $M[A, \mathbf{b}]$
 - a regra acima leva em conta também o símbolo $\$$
- Posições em branco na tabela são **error**

Parsing Table

$S \rightarrow \mathbf{aABe}$

$A \rightarrow \mathbf{bK}$

$K \rightarrow \mathbf{bcK}$

$K \rightarrow \boldsymbol{\varepsilon}$

$B \rightarrow \mathbf{d}$

	FIRST	FOLLOW
S	{ a }	{ \$ }
A	{ b }	{ d }
K	{ b, ε }	{ d }
B	{ d }	{ e }

	a	b	c	d	e	\$
S						
A						
K						
B						

Parsing Table

$S \rightarrow \mathbf{aABe}$

$A \rightarrow \mathbf{bK}$

$K \rightarrow \mathbf{bcK}$

$K \rightarrow \boldsymbol{\varepsilon}$

$B \rightarrow \mathbf{d}$

	FIRST	FOLLOW
<i>S</i>	{ a }	{ \$ }
<i>A</i>	{ b }	{ d }
<i>K</i>	{ b, ε }	{ d }
<i>B</i>	{ d }	{ e }

	a	b	c	d	e	\$
<i>S</i>	$S \rightarrow \mathbf{aABe}$					
<i>A</i>						
<i>K</i>						
<i>B</i>						

Parsing Table

$$S \rightarrow \mathbf{aABe}$$

$$A \rightarrow \mathbf{bK}$$

$$K \rightarrow \mathbf{bcK}$$

$$K \rightarrow \mathbf{\varepsilon}$$

$$B \rightarrow \mathbf{d}$$

	FIRST	FOLLOW
<i>S</i>	{ a }	{ \$ }
<i>A</i>	{ b }	{ d }
<i>K</i>	{ b, ε }	{ d }
<i>B</i>	{ d }	{ e }

	a	b	c	d	e	\$
<i>S</i>	$S \rightarrow \mathbf{aABe}$					
<i>A</i>		$A \rightarrow \mathbf{bK}$				
<i>K</i>						
<i>B</i>						

Parsing Table

$$S \rightarrow \mathbf{aABe}$$

$$A \rightarrow \mathbf{bK}$$

$$K \rightarrow \mathbf{bcK}$$

$$K \rightarrow \boldsymbol{\varepsilon}$$

$$B \rightarrow \mathbf{d}$$

	FIRST	FOLLOW
<i>S</i>	{ a }	{ \$ }
<i>A</i>	{ b }	{ d }
<i>K</i>	{ b, ε }	{ d }
<i>B</i>	{ d }	{ e }

	a	b	c	d	e	\$
<i>S</i>	$S \rightarrow \mathbf{aABe}$					
<i>A</i>		$A \rightarrow \mathbf{bK}$				
<i>K</i>		$K \rightarrow \mathbf{bcK}$				
<i>B</i>						

Parsing Table

$$S \rightarrow \mathbf{aABe}$$

$$A \rightarrow \mathbf{bK}$$

$$K \rightarrow \mathbf{bcK}$$

$$K \rightarrow \boldsymbol{\varepsilon}$$

$$B \rightarrow \mathbf{d}$$

	FIRST	FOLLOW
<i>S</i>	{ a }	{ \$ }
<i>A</i>	{ b }	{ d }
<i>K</i>	{ b, ε }	{ d }
<i>B</i>	{ d }	{ e }

	a	b	c	d	e	\$
<i>S</i>	$S \rightarrow \mathbf{aABe}$					
<i>A</i>		$A \rightarrow \mathbf{bK}$				
<i>K</i>		$K \rightarrow \mathbf{bcK}$		$K \rightarrow \boldsymbol{\varepsilon}$		
<i>B</i>						

Parsing Table

$$S \rightarrow \mathbf{aABe}$$

$$A \rightarrow \mathbf{bK}$$

$$K \rightarrow \mathbf{bcK}$$

$$K \rightarrow \boldsymbol{\varepsilon}$$

$$B \rightarrow \mathbf{d}$$

	FIRST	FOLLOW
<i>S</i>	{ a }	{ \$ }
<i>A</i>	{ b }	{ d }
<i>K</i>	{ b, ε }	{ d }
<i>B</i>	{ d }	{ e }

	a	b	c	d	e	\$
<i>S</i>	$S \rightarrow \mathbf{aABe}$					
<i>A</i>		$A \rightarrow \mathbf{bK}$				
<i>K</i>		$K \rightarrow \mathbf{bcK}$		$K \rightarrow \boldsymbol{\varepsilon}$		
<i>B</i>				$B \rightarrow \mathbf{d}$		

Exemplo

Expressões Aritméticas

$$\begin{array}{lll} E & \rightarrow & T E' \\ E' & \rightarrow & +T E' \mid \epsilon \\ T & \rightarrow & F T' \\ T' & \rightarrow & *F T' \mid \epsilon \\ F & \rightarrow & (E) \mid \text{id} \end{array}$$

FIRST

$E \rightarrow T E'$
 $E' \rightarrow +T E' \mid \varepsilon$
 $T \rightarrow F T'$
 $T' \rightarrow *F T' \mid \varepsilon$
 $F \rightarrow (E) \mid \text{id}$

	FIRST	FOLLOW
E		
E'		
T		
T'		
F		

FIRST(E') | FIRST(T') | FIRST(F')

$E \rightarrow T E'$
 $E' \rightarrow +T E' \mid \epsilon$
 $T \rightarrow F T'$
 $T' \rightarrow *F T' \mid \epsilon$
 $F \rightarrow (E) \mid id$

	FIRST	FOLLOW
E		
E'		
T		
T'		
F		

FIRST(E) | FIRST(T)

$E \rightarrow TE'$
 $E' \rightarrow +TE' \mid \epsilon$
 $T \rightarrow FT'$
 $T' \rightarrow *FT' \mid \epsilon$
 $F \rightarrow (E) \mid \text{id}$

	FIRST	FOLLOW
E		
E'	{ +, ϵ }	
T		
T'	{ *, ϵ }	
F	{ (, id }	

FOLLOW(E)

$E \rightarrow TE'$
 $E' \rightarrow +TE' \mid \epsilon$
 $T \rightarrow FT'$
 $T' \rightarrow *FT' \mid \epsilon$
 $F \rightarrow (E) \mid id$

	FIRST	FOLLOW
E	{ (, id }	
E'	{ +, ϵ }	
T	{ (, id }	
T'	{ *, ϵ }	
F	{ (, id }	

FOLLOW(E)

$E \rightarrow T E'$
 $E' \rightarrow + T E' \mid \epsilon$
 $T \rightarrow F T'$
 $T' \rightarrow * F T' \mid \epsilon$
 $F \rightarrow (E) \mid id$

	FIRST	FOLLOW
E	{ (, id }	{ \$ }
E'	{ +, ϵ }	
T	{ (, id }	
T'	{ *, ϵ }	
F	{ (, id }	

FOLLOW(E')

$E \rightarrow T E'$
 $E' \rightarrow +T E' \mid \epsilon$
 $T \rightarrow F T'$
 $T' \rightarrow *F T' \mid \epsilon$
 $F \rightarrow (E) \mid id$

	FIRST	FOLLOW
E	{ (, id }	{ \$,) }
E'	{ +, ϵ }	
T	{ (, id }	
T'	{ *, ϵ }	
F	{ (, id }	

FOLLOW(T)

$E \rightarrow T E'$
 $E' \rightarrow + T E' \mid \epsilon$
 $T \rightarrow F T'$
 $T' \rightarrow * F T' \mid \epsilon$
 $F \rightarrow (E) \mid id$

	FIRST	FOLLOW
E	{ (, id }	{ \$,) }
E'	{ +, ϵ }	{ \$,) }
T	{ (, id }	
T'	{ *, ϵ }	
F	{ (, id }	

FOLLOW(T)

$E \rightarrow T E'$
 $E' \rightarrow +T E' \mid \epsilon$
 $T \rightarrow F T'$
 $T' \rightarrow *F T' \mid \epsilon$
 $F \rightarrow (E) \mid id$

	FIRST	FOLLOW
E	{ (, id }	{ \$,) }
E'	{ +, ϵ }	{ \$,) }
T	{ (, id }	{ + }
T'	{ *, ϵ }	
F	{ (, id }	

FOLLOW(T')

$E \rightarrow T E'$
 $E' \rightarrow + T E' \mid \epsilon$
 $T \rightarrow F T'$
 $T' \rightarrow * F T' \mid \epsilon$
 $F \rightarrow (E) \mid id$

	FIRST	FOLLOW
E	{ (, id }	{ \$,) }
E'	{ +, ϵ }	{ \$,) }
T	{ (, id }	{ +, \$,) }
T'	{ *, ϵ }	
F	{ (, id }	

FOLLOW(F)

$E \rightarrow TE'$
 $E' \rightarrow +TE' \mid \epsilon$
 $T \rightarrow F T'$
 $T' \rightarrow *F T' \mid \epsilon$
 $F \rightarrow (E) \mid id$

	FIRST	FOLLOW
E	{ (, id }	{ \$,) }
E'	{ +, ϵ }	{ \$,) }
T	{ (, id }	{ +, \$,) }
T'	{ *, ϵ }	{ +, \$,) }
F	{ (, id }	

FOLLOW(F)

$E \rightarrow TE'$
 $E' \rightarrow +TE' \mid \epsilon$
 $T \rightarrow FT'$
 $T' \rightarrow *FT' \mid \epsilon$
 $F \rightarrow (E) \mid id$

	FIRST	FOLLOW
E	{ (, id }	{ \$,) }
E'	{ +, ϵ }	{ \$,) }
T	{ (, id }	{ +, \$,) }
T'	{ *, ϵ }	{ +, \$,) }
F	{ (, id }	{ * }

FIRST e FOLLOW

$E \rightarrow T E'$
 $E' \rightarrow +T E' \mid \epsilon$
 $T \rightarrow F T'$
 $T' \rightarrow *F T' \mid \epsilon$
 $F \rightarrow (E) \mid \text{id}$

	FIRST	FOLLOW
E	{ (, id }	{ \$,) }
E'	{ +, ϵ }	{ \$,) }
T	{ (, id }	{ +, \$,) }
T'	{ *, ϵ }	{ +, \$,) }
F	{ (, id }	{ *, +, \$,) }

Predictive Parsing Table

	Símbolos de entrada (terminais e \$)					
Não terminais	id	+	*	()	\$
<i>E</i>						
<i>E'</i>						
<i>T</i>						
<i>T'</i>						
<i>F</i>						

E

→ T E'

E'

→ + T E' | ε

T

→ F T'

T'

→ * F T' | ε

F

→ (E) | id

	FIRST	FOLLOW
E	{ (, id }	{ \$,) }
E'	{ +, ε }	{ \$,) }
T	{ (, id }	{ +, \$,) }
T'	{ *, ε }	{ +, \$,) }
F	{ (, id }	{ *, +, \$,) }

	id	+	*	()	\$
<i>E</i>						
<i>E'</i>						
<i>T</i>						
<i>T'</i>						
<i>F</i>						

E

→ T E'

E'

→ + T E' | ε

T

→ F T'

T'

→ * F T' | ε

F

→ (E) | id

	FIRST	FOLLOW
E	{ (, id }	{ \$,) }
E'	{ +, ε }	{ \$,) }
T	{ (, id }	{ +, \$,) }
T'	{ *, ε }	{ +, \$,) }
F	{ (, id }	{ *, +, \$,) }

	id	+	*	()	\$
E						
E'						
T						
T'						
F						

E

→ T E'

E'

→ + T E' | ε

T

→ F T'

T'

→ * F T' | ε

F

→ (E) | id

	FIRST	FOLLOW
E	{ (, id }	{ \$,) }
E'	{ +, ε }	{ \$,) }
T	{ (, id }	{ +, \$,) }
T'	{ *, ε }	{ +, \$,) }
F	{ (, id }	{ *, +, \$,) }

	id	+	*	()	\$
E	E → T E'			E → T E'		
E'						
T						
T'						
F						

E

→ T E'

E'

→ + T E' | ε

T

→ F T'

T'

→ * F T' | ε

F

→ (E) | id

	FIRST	FOLLOW
E	{ (, id }	{ \$,) }
E'	{ +, ε }	{ \$,) }
T	{ (, id }	{ +, \$,) }
T'	{ *, ε }	{ +, \$,) }
F	{ (, id }	{ *, +, \$,) }

	id	+	*	()	\$
E	E → T E'			E → T E'		
E'						
T						
T'						
F						

E

→ T E'

E'

→ + T E' | ε

T

→ F T'

T'

→ * F T' | ε

F

→ (E) | id

	FIRST	FOLLOW
E	{ (, id }	{ \$,) }
E'	{ +, ε }	{ \$,) }
T	{ (, id }	{ +, \$,) }
T'	{ *, ε }	{ +, \$,) }
F	{ (, id }	{ *, +, \$,) }

	id	+	*	()	\$
E	E → T E'			E → T E'		
E'		E' → + T E'			E' → ε	E' → ε
T						
T'						
F						

E

→ T E'

E'

→ + T E' | ε

T

→ F T'

T'

→ * F T' | ε

F

→ (E) | id

	FIRST	FOLLOW
E	{ (, id }	{ \$,) }
E'	{ +, ε }	{ \$,) }
T	{ (, id }	{ +, \$,) }
T'	{ *, ε }	{ +, \$,) }
F	{ (, id }	{ *, +, \$,) }

	id	+	*	()	\$
<i>E</i>	E → T E'			E → T E'		
<i>E'</i>		E' → + T E'			E' → ε	E' → ε
<i>T</i>						
<i>T'</i>						
<i>F</i>						

E

→ T E'

E'

→ + T E' | ε

T

→ F T'

T'

→ * F T' | ε

F

→ (E) | id

	FIRST	FOLLOW
E	{ (, id }	{ \$,) }
E'	{ +, ε }	{ \$,) }
T	{ (, id }	{ +, \$,) }
T'	{ *, ε }	{ +, \$,) }
F	{ (, id }	{ *, +, \$,) }

	id	+	*	()	\$
E	E → T E'			E → T E'		
E'		E' → + T E'			E' → ε	E' → ε
T	T → F T'			T → F T'		
T'						
F						

E

→ T E'

E'

→ + T E' | ε

T

→ F T'

T'

→ * F T' | ε

F

→ (E) | id

	FIRST	FOLLOW
E	{ (, id }	{ \$,) }
E'	{ +, ε }	{ \$,) }
T	{ (, id }	{ +, \$,) }
T'	{ *, ε }	{ +, \$,) }
F	{ (, id }	{ *, +, \$,) }

	id	+	*	()	\$
E	E → T E'			E → T E'		
E'		E' → + T E'			E' → ε	E' → ε
T	T → F T'			T → F T'		
T'						
F						

E

→ T E'

E'

→ + T E' | ε

T

→ F T'

T'

→ * F T' | ε

F

→ (E) | id

	FIRST	FOLLOW
E	{ (, id }	{ \$,) }
E'	{ +, ε }	{ \$,) }
T	{ (, id }	{ +, \$,) }
T'	{ *, ε }	{ +, \$,) }
F	{ (, id }	{ *, +, \$,) }

	id	+	*	()	\$
E	E → T E'			E → T E'		
E'		E' → + T E'			E' → ε	E' → ε
T	T → F T'			T → F T'		
T'		T' → ε	T' → * F T'		T' → ε	T' → ε
F						

E

→ T E'

E'

→ + T E' | ε

T

→ F T'

T'

→ * F T' | ε

F

→ (E) | id

	FIRST	FOLLOW
E	{ (, id }	{ \$,) }
E'	{ +, ε }	{ \$,) }
T	{ (, id }	{ +, \$,) }
T'	{ *, ε }	{ +, \$,) }
F	{ (, id }	{ *, +, \$,) }

	id	+	*	()	\$
E	E → T E'			E → T E'		
E'		E' → + T E'			E' → ε	E' → ε
T	T → F T'			T → F T'		
T'		T' → ε	T' → * F T'		T' → ε	T' → ε
F						

E

→ T E'

E'

→ + T E' | ε

T

→ F T'

T'

→ * F T' | ε

F

→ (E) | id

	FIRST	FOLLOW
E	{ (, id }	{ \$,) }
E'	{ +, ε }	{ \$,) }
T	{ (, id }	{ +, \$,) }
T'	{ *, ε }	{ +, \$,) }
F	{ (, id }	{ *, +, \$,) }

	id	+	*	()	\$
E	E → T E'			E → T E'		
E'		E' → + T E'			E' → ε	E' → ε
T	T → F T'			T → F T'		
T'		T' → ε	T' → * F T'		T' → ε	T' → ε
F	F → id			F → (E)		

Quais as características de gramáticas LL(1) serão refletidas nestas tabelas?

Exemplo Anterior

$S \rightarrow \mathbf{aABe}$

$A \rightarrow A\mathbf{bc}$

$A \rightarrow \mathbf{b}$

$B \rightarrow \mathbf{d}$

	FIRST	FOLLOW
<i>S</i>		
<i>A</i>		
<i>B</i>		

	a	b	c	d	e	\$
<i>S</i>						
<i>A</i>						
<i>B</i>						

FIRST e FOLLOW

$S \rightarrow \mathbf{aABe}$

$A \rightarrow A\mathbf{bc}$

$A \rightarrow \mathbf{b}$

$B \rightarrow \mathbf{d}$

	FIRST	FOLLOW
<i>S</i>	{ a }	{ \$ }
<i>A</i>	{ b }	{ b, d }
<i>B</i>	{ d }	{ e }

	a	b	c	d	e	\$
<i>S</i>						
<i>A</i>						
<i>B</i>						

Parsing Table

$S \rightarrow \mathbf{aABe}$

$A \rightarrow A\mathbf{bc}$

$A \rightarrow \mathbf{b}$

$B \rightarrow \mathbf{d}$

	FIRST	FOLLOW
S	{ a }	{ \$ }
A	{ b }	{ b, d }
B	{ d }	{ e }

	a	b	c	d	e	\$
S	$S \rightarrow \mathbf{aABe}$					
A		$A \rightarrow A\mathbf{bc}$ $A \rightarrow \mathbf{b}$				
B				$B \rightarrow \mathbf{d}$		

Exemplo - Fatoração à Esquerda

$S \rightarrow i E t S S' \mid a$

$S' \rightarrow e S \mid \epsilon$

$E \rightarrow b$

	FIRST	FOLLOW
S		
S'		
E		

FIRST(S)

$S \rightarrow i E t S S' \mid a$

$S' \rightarrow e S \mid \epsilon$

$E \rightarrow b$

	FIRST	FOLLOW
S	{ i, a }	
S'		
E		

FIRST(S')

$S \rightarrow i E t S S' \mid a$

$S' \rightarrow e S \mid \epsilon$

$E \rightarrow b$

	FIRST	FOLLOW
S	{ i, a }	
S'	{ e, ϵ }	
E		

FIRST(E)

$S \rightarrow i E t S S' \mid a$

$S' \rightarrow e S \mid \epsilon$

$E \rightarrow b$

	FIRST	FOLLOW
S	{ i, a }	
S'	{ e, ϵ }	
E	{ b }	

FOLLOW(S)

$S \rightarrow i E t S S' \mid a$
 $S' \rightarrow e S \mid \epsilon$
 $E \rightarrow b$

	FIRST	FOLLOW
S	{ i, a }	{ \$, e }
S'	{ e, ϵ }	
E	{ b }	

FOLLOW(S')

$S \rightarrow i E t S S' \mid a$
 $S' \rightarrow e S \mid \epsilon$
 $E \rightarrow b$

	FIRST	FOLLOW
S	{ i, a }	{ \$, e }
S'	{ e, ϵ }	{ \$, e }
E	{ b }	

FOLLOW(E)

$S \rightarrow i E t S S' \mid a$
 $S' \rightarrow e S \mid \epsilon$
 $E \rightarrow b$

	FIRST	FOLLOW
S	{ i, a }	{ \$, e }
S'	{ e, ϵ }	{ \$, e }
E	{ b }	{ t }

$S \rightarrow iEtSS' \mid a$
 $S' \rightarrow eS \mid \epsilon$
 $E \rightarrow b$

	FIRST	FOLLOW
S	{ i, a }	{ \$, e }
S'	{ e, ε }	{ \$, e }
E	{ b }	{ t }

	a	b	e	i	t	\$
S						
S'						
E						

$S \rightarrow iEtSS' \mid a$
 $S' \rightarrow eS \mid \epsilon$
 $E \rightarrow b$

	FIRST	FOLLOW
S	{ i , a }	{ \$, e }
S'	{ e , ε }	{ \$, e }
E	{ b }	{ t }

	a	b	e	i	t	\$
S				$S \rightarrow iEtSS'$		
S'						
E						

$S \rightarrow iEtSS' \mid a$
 $S' \rightarrow eS \mid \epsilon$
 $E \rightarrow b$

	FIRST	FOLLOW
S	{ i, a }	{ \$, e }
S'	{ e, ε }	{ \$, e }
E	{ b }	{ t }

	a	b	e	i	t	\$
S	$S \rightarrow a$			$S \rightarrow iEtSS'$		
S'						
E						

$S \rightarrow iEtSS' \mid a$
 $S' \rightarrow eS \mid \epsilon$
 $E \rightarrow b$

	FIRST	FOLLOW
S	{ i, a }	{ \$, e }
S'	{ e, ε }	{ \$, e }
E	{ b }	{ t }

	a	b	e	i	t	\$
S	$S \rightarrow a$			$S \rightarrow iEtSS'$		
S'			$S' \rightarrow eS$			
E						

$S \rightarrow iEtSS' \mid a$
 $S' \rightarrow eS \mid \epsilon$
 $E \rightarrow b$

	FIRST	FOLLOW
S	{ i, a }	{ \$, e }
S'	{ e, ε }	{ \$, e }
E	{ b }	{ t }

	a	b	e	i	t	\$
S	$S \rightarrow a$			$S \rightarrow iEtSS'$		
S'			$S' \rightarrow \epsilon$ $S' \rightarrow eS$			$S' \rightarrow \epsilon$
E						

$$\begin{aligned}
 S &\rightarrow iEtSS' \mid a \\
 S' &\rightarrow eS \mid \epsilon \\
 E &\rightarrow b
 \end{aligned}$$

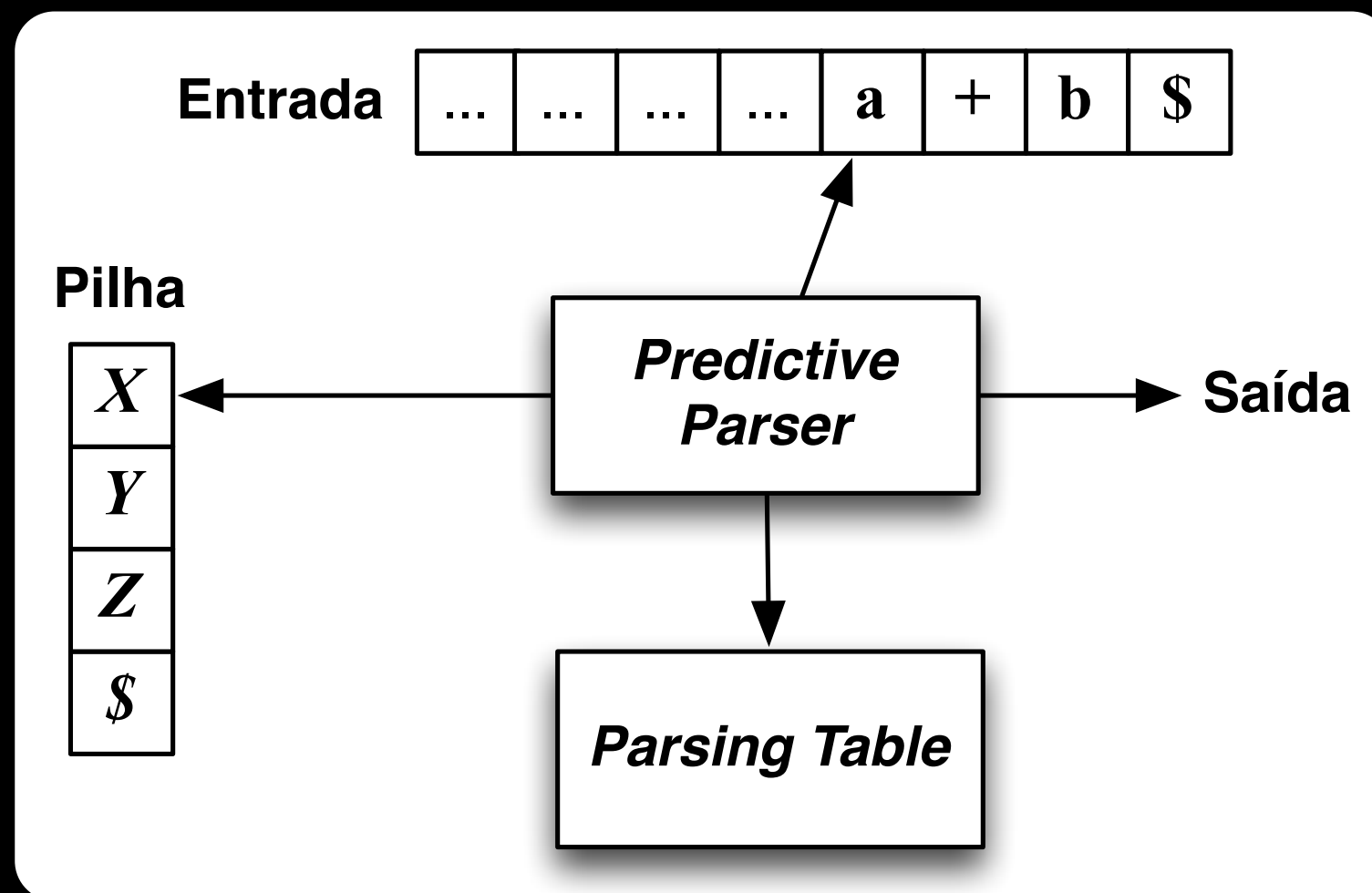
	FIRST	FOLLOW
S	{ i, a }	{ \$, e }
S'	{ e, ε }	{ \$, e }
E	{ b }	{ t }

	a	b	e	i	t	\$
S	S → a			S → iEtSS'		
S'			S' → ε S' → eS			S' → ε
E		E → b				

Non-Recursive Predictive Parsing

- Um parser preditivo não recursivo pode ser construído mantendo uma pilha explicitamente
 - ao invés de implicitamente por meio das chamadas recursivas
- Se w é a entrada que foi casada até o momento, a pilha vai manter uma sequência de símbolos da gramática tais que $S \rightarrow^* w\alpha$

Table-driven Predictive Parsing



1. $E \rightarrow T E'$
2. $E' \rightarrow + T E'$
3. $E' \rightarrow \epsilon$
4. $T \rightarrow F T'$
5. $T' \rightarrow * F T'$
6. $T' \rightarrow \epsilon$
7. $F \rightarrow (E)$
8. $F \rightarrow id$

	id	+	*	()	\$
<i>E</i>	1			1		
<i>E'</i>		2			3	3
<i>T</i>	4			4		
<i>T'</i>		6	5		6	6
<i>F</i>	8			7		

Pilha	Entrada (buffer)
E\$	id + id * id\$
T E'\$	id + id * id\$
F T' E' \$	id + id * id\$
id T' E' \$	id + id * id\$
T' E' \$	+ id * id\$
E' \$	+ id * id\$
+ T E' \$	+ id * id\$
T E' \$	id * id\$
F T' E' \$	id * id\$
id T' E' \$	id * id\$
T' E' \$	* id\$
* F T' E' \$	* id\$
F T' E' \$	id\$
id T' E' \$	id\$
T' E' \$	\$
E' \$	\$
\$	\$

Inicialmente o parser tem $w\$$ na entrada e o símbolo inicial S no topo da pilha, acima de $\$$

let a = primeiro token de w

let X = top()

while ($X \neq \$$) {

if ($X = a$) pop(); **let** a = nextToken();

else if (X é terminal) error();

else if ($M[X,a] == \text{error}$) error();

else if ($M[X,a] == X \rightarrow Y_1 Y_2 \dots Y_k$) {

 pop();

 push(Y_k); push(Y_{k-1}); ... push(Y_2); push(Y_1);

 }

let X = top()

}

Por que se importar
com $LL(1)$?

LL(1) é *real-world*

- Linguagens de programação podem ser expressas por meio de gramáticas LL(1)
- Exemplos
 - LISP
 - Python
 - JavaScript

LL(1) é simples

- Pode ser implementada manualmente por meio de tabelas como as vistas anteriormente
- Utilizando a técnica *recursive descent*
 - uma função para cada não-terminal
 - funções chamam outras funções de acordo com o *lookahead*

LL(1) é eficiente

- Parsers guiados por tabela ou recursive-descent são eficientes
- Podem fazer o parsing em tempo $O(n f(G))$
 - n é o tamanho da string
 - $f(G)$ é uma função de acordo com o tamanho da gramática (geralmente uma constante)

Gramática LL(1)?

$S \rightarrow B c \mid D B$

$B \rightarrow a b \mid c S$

$D \rightarrow d \mid \varepsilon$

	FIRST	FOLLOW
S		
B		
D		

Gramática LL(1)?

$S \rightarrow B c \mid D B$

$B \rightarrow a b \mid c S$

$D \rightarrow d \mid \varepsilon$

	FIRST	FOLLOW
S		
B		
D	{ d, ε }	

Gramática LL(1)?

$S \rightarrow B c \mid D B$

$B \rightarrow a b \mid c S$

$D \rightarrow d \mid \varepsilon$

	FIRST	FOLLOW
S		
B	{ a , c }	
D	{ d , ε }	

Gramática LL(1)?

$S \rightarrow B c \mid D B$

$B \rightarrow a b \mid c S$

$D \rightarrow d \mid \varepsilon$

	FIRST	FOLLOW
S	{ a , c }	
B	{ a , c }	
D	{ d , ε }	

Gramática LL(1)?

$S \rightarrow B c \mid D B$

$B \rightarrow a b \mid c S$

$D \rightarrow d \mid \varepsilon$

	FIRST	FOLLOW
S	{ a , c , d }	
B	{ a , c }	
D	{ d , ε }	

Gramática LL(1)?

$S \rightarrow B c \mid D B$

$B \rightarrow a b \mid c S$

$D \rightarrow d \mid \varepsilon$

	FIRST	FOLLOW
S	$\{ a, c, d \}$	$\{ \$ \}$
B	$\{ a, c \}$	
D	$\{ d, \varepsilon \}$	

Gramática LL(1)?

$S \rightarrow B c \mid D B$

$B \rightarrow a b \mid c S$

$D \rightarrow d \mid \varepsilon$

	FIRST	FOLLOW
S	$\{ a, c, d \}$	$\{ \$ \}$
B	$\{ a, c \}$	$\{ c \}$
D	$\{ d, \varepsilon \}$	

Gramática LL(1)?

$S \rightarrow B c \mid D B$

$B \rightarrow a b \mid c S$

$D \rightarrow d \mid \varepsilon$

	FIRST	FOLLOW
S	$\{ a, c, d \}$	$\{ \$ \}$
B	$\{ a, c \}$	$\{ c, \$ \}$
D	$\{ d, \varepsilon \}$	

Gramática LL(1)?

$S \rightarrow B c \mid D B$

$B \rightarrow a b \mid c S$

$D \rightarrow d \mid \varepsilon$

	FIRST	FOLLOW
S	$\{ a, c, d \}$	$\{ \$ \}$
B	$\{ a, c \}$	$\{ c, \$ \}$
D	$\{ d, \varepsilon \}$	$\{ a, c \}$

Gramática LL(1)?

$S \rightarrow B c \mid D B$

$B \rightarrow a b \mid c S$

$D \rightarrow d \mid \varepsilon$

	FIRST	FOLLOW
S	$\{ a, c, d \}$	$\{ \$, c \}$
B	$\{ a, c \}$	$\{ c, \$ \}$
D	$\{ d, \varepsilon \}$	$\{ a, c \}$

$S \rightarrow B\ c \mid D\ B$
 $B \rightarrow a\ b \mid c\ S$
 $D \rightarrow d \mid \epsilon$

	FIRST	FOLLOW
S	{ a, c, d }	{ \$, c }
B	{ a, c }	{ c, \$ }
D	{ d, ε }	{ a, c }

	a	b	c	d	\$
<i>S</i>					
<i>B</i>					
<i>D</i>					

$S \rightarrow B\ c \mid D\ B$
 $B \rightarrow a\ b \mid c\ S$
 $D \rightarrow d \mid \epsilon$

	FIRST	FOLLOW
S	{ a, c, d }	{ \$, c }
B	{ a, c }	{ c, \$ }
D	{ d, ε }	{ a, c }

	a	b	c	d	\$
<i>S</i>	$S \rightarrow B\ c$		$S \rightarrow B\ c$		
<i>B</i>					
<i>D</i>					

$$\begin{aligned}
 S &\rightarrow B\ c \mid D\ B \\
 B &\rightarrow a\ b \mid c\ S \\
 D &\rightarrow d \mid \varepsilon
 \end{aligned}$$

	FIRST	FOLLOW
S	{ a, c, d }	{ \$, c }
B	{ a, c }	{ c, \$ }
D	{ d, ε }	{ a, c }

	a	b	c	d	\$
<i>S</i>	$S \rightarrow B\ c$ $S \rightarrow D\ B$		$S \rightarrow B\ c$ $S \rightarrow D\ B$	$S \rightarrow D\ B$	
<i>B</i>					
<i>D</i>					

$$\begin{aligned}
 S &\rightarrow B\ c \mid D\ B \\
 B &\rightarrow a\ b \mid c\ S \\
 D &\rightarrow d \mid \varepsilon
 \end{aligned}$$

	FIRST	FOLLOW
S	{ a, c, d }	{ \$, c }
B	{ a, c }	{ c, \$ }
D	{ d, ε }	{ a, c }

	a	b	c	d	\$
S	$S \rightarrow B\ c$ $S \rightarrow D\ B$		$S \rightarrow B\ c$ $S \rightarrow D\ B$	$S \rightarrow D\ B$	
B	$B \rightarrow a\ b$				
D					

$$\begin{aligned}
 S &\rightarrow B\ c \mid D\ B \\
 B &\rightarrow a\ b \mid c\ S \\
 D &\rightarrow d \mid \varepsilon
 \end{aligned}$$

	FIRST	FOLLOW
S	{ a, c, d }	{ \$, c }
B	{ a, c }	{ c, \$ }
D	{ d, ε }	{ a, c }

	a	b	c	d	\$
<i>S</i>	$S \rightarrow B\ c$ $S \rightarrow D\ B$		$S \rightarrow B\ c$ $S \rightarrow D\ B$	$S \rightarrow D\ B$	
<i>B</i>	$B \rightarrow a\ b$		$B \rightarrow c\ S$		
<i>D</i>					

$$\begin{aligned}
 S &\rightarrow B\ c \mid D\ B \\
 B &\rightarrow a\ b \mid c\ S \\
 D &\rightarrow d \mid \varepsilon
 \end{aligned}$$

	FIRST	FOLLOW
S	{ a, c, d }	{ \$, c }
B	{ a, c }	{ c, \$ }
D	{ d, ε }	{ a, c }

	a	b	c	d	\$
<i>S</i>	$S \rightarrow B\ c$ $S \rightarrow D\ B$		$S \rightarrow B\ c$ $S \rightarrow D\ B$	$S \rightarrow D\ B$	
<i>B</i>	$B \rightarrow a\ b$		$B \rightarrow c\ S$		
<i>D</i>				$D \rightarrow d$	

$S \rightarrow B\ c \mid D\ B$
 $B \rightarrow a\ b \mid c\ S$
 $D \rightarrow d \mid \epsilon$

	FIRST	FOLLOW
S	{ a, c, d }	{ \$, c }
B	{ a, c }	{ c, \$ }
D	{ d, ε }	{ a, c }

	a	b	c	d	\$
S	$S \rightarrow B\ c$ $S \rightarrow D\ B$		$S \rightarrow B\ c$ $S \rightarrow D\ B$	$S \rightarrow D\ B$	
B	$B \rightarrow a\ b$		$B \rightarrow c\ S$		
D	$D \rightarrow \epsilon$		$D \rightarrow \epsilon$	$D \rightarrow d$	

$stmt \rightarrow \text{if } expr \text{ then } stmt$
 $stmt \rightarrow \text{while } expr \text{ do } stmt$
 $stmt \rightarrow expr;$
 $expr \rightarrow term \mapsto id$
 $expr \rightarrow \text{zero? } term$
 $expr \rightarrow \text{not } expr$
 $expr \rightarrow ++ id$
 $expr \rightarrow -- id$
 $term \rightarrow id$
 $term \rightarrow \text{constant}$

`id \mapsto id;`

`while not zero? id do -- id;`

`if not zero? id then
 if not zero? id then
 constant \mapsto id;`