

Compiladores

Análise Sintática

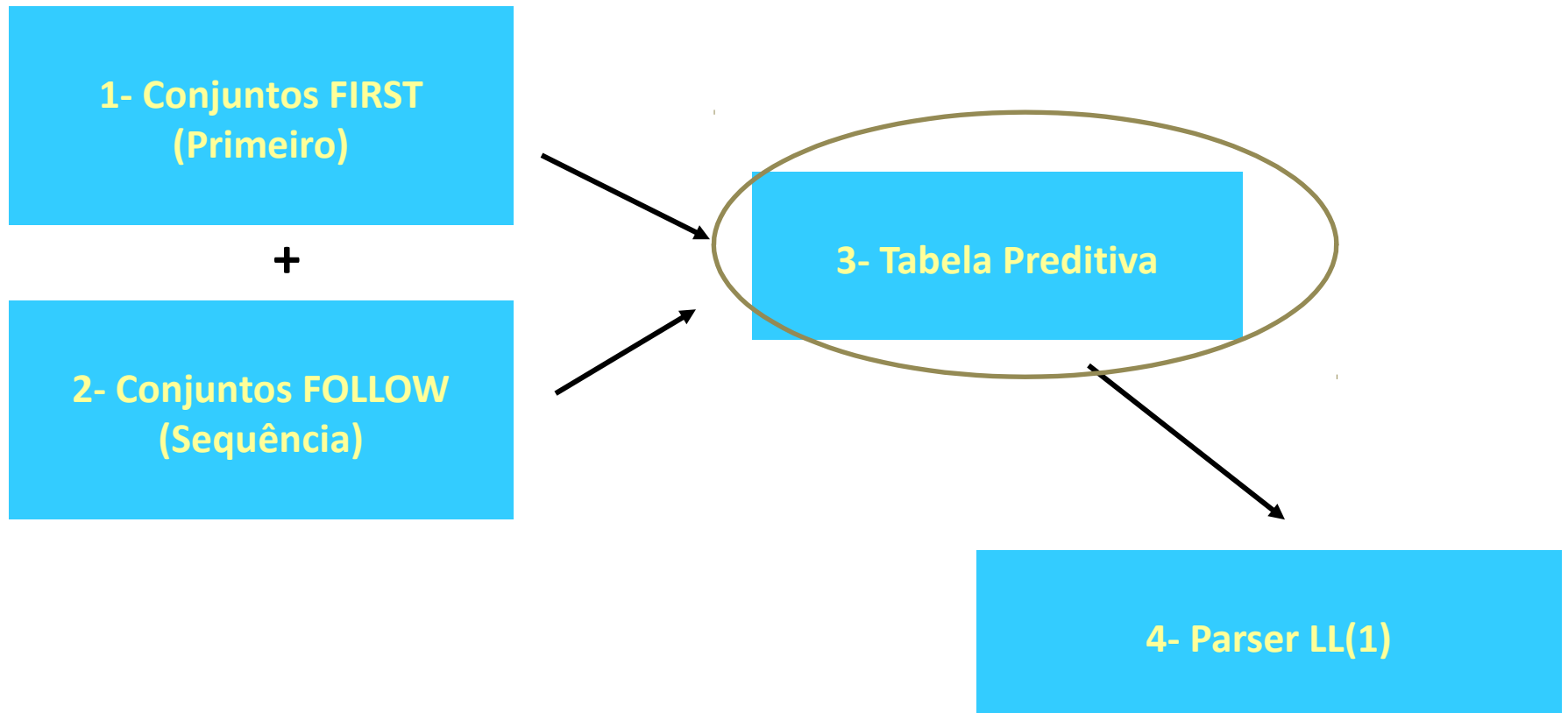
Análise Sintática LL(1) - continuação

Prof. Dr. Luiz Eduardo G. Martins
(adaptado por Profa Dra Ana Carolina Lorena)

UNIFESP

Análise Sintática LL(1)

- Estratégia de construção do *parser* LL(1)



Análise Sintática LL(1)

- Estratégia de construção do *parser* LL(1)

$S \rightarrow aSAb$

$S \rightarrow bAa$

$A \rightarrow bAb$

$A \rightarrow c$

Não-Terminal	Terminal		
	a	b	c
S	$S \rightarrow aSAb$	$S \rightarrow bAa$	
A		$A \rightarrow bAb$	$A \rightarrow c$

Para montar a tabela, é necessário saber que produção deve ser aplicada a partir de cada não terminal que possa gerar o terminal correspondente

Análise Sintática LL(1)

- Estratégia de construção do *parser* LL(1)

$S \rightarrow (S)S$

$S \rightarrow \varepsilon$

Não-Terminal	Terminal	
	()
S	$S \rightarrow (S)S$	$S \rightarrow \varepsilon$

Para montar a tabela, é necessário saber que produção deve ser aplicada a partir de cada não terminal que possa gerar o terminal correspondente

Análise Sintática LL(1)

- Estratégia de construção do *parser* LL(1)
 - ❑ A construção da tabela preditiva para gramáticas complexas não é tarefa trivial
 - ❑ Para auxiliar na construção da tabela preditiva, adotamos a construção de conjuntos Primeiros (*FIRST*) e de Sequência (*FOLLOW*)

Análise Sintática LL(1)

- Conjunto FIRST

- Dada uma gramática, deve ser criado um conjunto FIRST para cada símbolo da gramática, terminal ou não-terminal
- Computa os **terminais** que podem aparecer como os **primeiros símbolos** em uma **cadeia derivada de um terminal ou não-terminal** e também se um não-terminal pode derivar ϵ (desaparecer)
 - Para saber que terminais podem ser derivados a partir de um não terminal

Análise Sintática LL(1)

- Conjunto FIRST: definição

—Seja X um símbolo gramatical, o conjunto $\text{FIRST}(X)$ é composto por terminais, e possivelmente ϵ , definido da seguinte maneira:

- Se X for um terminal ou ϵ , então $\text{FIRST}(X) = \{X\}$
- Se X for um não-terminal, então para cada escolha de produção $X \rightarrow X_1X_2\dots X_n$ $\text{FIRST}(X)$ contém $\text{FIRST}(X_1) - \{\epsilon\}$
- Se para algum $i < n$, todos os conjuntos $\text{FIRST}(X_1), \dots, \text{FIRST}(X_i)$ contiverem ϵ , então $\text{FIRST}(X)$ conterá $\text{FIRST}(X_{i+1}) - \{\epsilon\}$
- Se todos os conjuntos $\text{FIRST}(X_1), \dots, \text{FIRST}(X_n)$ contiverem ϵ , então $\text{FIRST}(X)$ também conterá ϵ

Análise Sintática LL(1)

- Conjunto FIRST

—Observações sobre a construção dos conjuntos FIRST:

- O conjunto FIRST é formado apenas por símbolos terminais (e possivelmente ϵ)
- A definição dos conjuntos FIRST funciona “à esquerda”
- Um não-terminal A é anulável (ou seja, pode desaparecer) se e somente se $\text{FIRST}(A)$ contiver ϵ
 - Ou seja, se houver uma derivação $A \Rightarrow^* \epsilon$

Análise Sintática LL(1)

- Conjunto FIRST

- Algoritmo para construção do conjunto FIRST

```
for cada não-terminal  $A$  do Primeiro( $A$ ) := {};  
while houver alterações em algum Primeiro( $A$ ) do  
  for cada escolha de produção  $A \rightarrow X_1X_2...X_n$  do  
     $k := 1$  ; Continue := true ;  
    while Continue = true and  $k \leq n$  do  
      acrescente Primeiro( $X_k$ ) - { $\epsilon$ } a Primeiro( $A$ ) ;  
      if  $\epsilon$  não pertencer a Primeiro( $X_k$ ) then Continue := false ;  
       $k := k + 1$  ;  
    if Continue = true then acrescente  $\epsilon$  a Primeiro( $A$ ) ;
```

Figura 4.6 Algoritmo para a computação de Primeiro(A) para todos os não-terminais A .

Análise Sintática LL(1)

- Conjunto FIRST

–Exemplo (4.9 – Louden):

- Considere a gramática:

$\text{exp} \rightarrow \text{exp soma termo} \mid \text{termo}$

$\text{soma} \rightarrow + \mid -$

$\text{termo} \rightarrow \text{termo mult fator} \mid \text{fator}$

$\text{mult} \rightarrow *$

$\text{fator} \rightarrow (\text{exp}) \mid \text{NUM}$

Análise Sintática LL(1)

- Conjunto FIRST

Tabela 4.6 Computação dos conjuntos Primeiros para a gramática do Exemplo 4.9.

Regra gramatical	Passada 1	Passada 2	Passada 3
$exp \rightarrow exp$ soma termo			
$exp \rightarrow termo$			$Primeiro(exp) = \{ (, número \}$
$soma \rightarrow +$	$Primeiro(soma) = \{ + \}$		
$soma \rightarrow -$	$Primeiro(soma) = \{ +, - \}$		
$termo \rightarrow termo$ mult fator			
$termo \rightarrow fator$		$Primeiro(termo) = \{ (, número \}$	
$mult \rightarrow *$	$Primeiro(mult) = \{ * \}$		
$fator \rightarrow (exp)$	$Primeiro(fator) = \{ (\}$		
$fator \rightarrow número$	$Primeiro(fator) = \{ (, número \}$		

```

for cada não-terminal  $A$  do  $Primeiro(A) := \{ \}$ ;
while houver alterações em algum  $Primeiro(A)$  do
    for cada escolha de produção  $A \rightarrow X_1 X_2 \dots X_n$  do
        acrescente  $Primeiro(X_1)$  a  $Primeiro(A)$ ;
    
```

Análise Sintática LL(1)

- Conjunto FIRST

—Exemplo:

- Considere a gramática:

$\text{exp} \rightarrow \text{exp soma termo} \mid \text{termo}$

$\text{soma} \rightarrow + \mid -$

$\text{termo} \rightarrow \text{termo mult fator} \mid \text{fator}$

$\text{mult} \rightarrow *$

$\text{fator} \rightarrow (\text{exp}) \mid \text{NUM}$

$\text{Primeiro}(\text{exp}) = \{ (, \text{número} \}$

$\text{Primeiro}(\text{termo}) = \{ (, \text{número} \}$

$\text{Primeiro}(\text{fator}) = \{ (, \text{número} \}$

$\text{Primeiro}(\text{soma}) = \{ +, - \}$

$\text{Primeiro}(\text{mult}) = \{ * \}$

Análise Sintática LL(1)

- Conjunto FIRST

—Exemplo:

- Considere a gramática:

declaração \rightarrow if-decl | outra

if-decl \rightarrow if (exp) declaração else-parte

else-parte \rightarrow else declaração | ϵ

exp \rightarrow 0 | 1

Análise Sintática LL(1)

- Conjunto FIRST

—Exemplo:

Regra gramatical	Passada 1	Passada 2
declaração \rightarrow if-decl	$\text{FIRST}(\text{declaração}) = \text{FIRST}(\text{if-decl})$	$\text{FIRST}(\text{declaração}) = \{\text{if}, \text{outra}\}$
declaração \rightarrow outra	$\text{FIRST}(\text{declaração}) = \{\text{outra}\}$	
if-decl \rightarrow if (exp) declaração else-parte	$\text{FIRST}(\text{if-decl}) = \{\text{if}\}$	
else-parte \rightarrow else declaração	$\text{FIRST}(\text{else-parte}) = \{\text{else}\}$	
else-parte $\rightarrow \epsilon$	$\text{FIRST}(\text{else-parte}) = \{\text{else}, \epsilon\}$	
exp \rightarrow 0	$\text{FIRST}(\text{exp}) = \{0\}$	
exp \rightarrow 1	$\text{FIRST}(\text{exp}) = \{0, 1\}$	

Análise Sintática LL(1)

- Conjunto FIRST

—Exercício:

- Considere a gramática:

decl-sequência \rightarrow decl decl-seq'

decl-seq' \rightarrow ; decl-sequência | ϵ

decl \rightarrow s

Exercício 1: obter os conjuntos FIRST para essa gramática

Análise Sintática LL(1)

- Conjunto FOLLOW

- Necessário para a montagem da tabela preditiva, quando algum conjunto FIRST possui ϵ
- Dado um não-terminal A, o conjunto FOLLOW(A), composto por terminais e possivelmente \$, é definido como:

1. Se A for o símbolo inicial, então \$ pertence a FOLLOW(A);
2. Se houver uma produção $B \rightarrow \alpha A \gamma$, então $\text{FIRST}(\gamma) - \{\epsilon\}$ pertence a FOLLOW(A);
3. Se houver uma produção $B \rightarrow \alpha A \gamma$ tal que ϵ pertença a $\text{FIRST}(\gamma)$, então FOLLOW(A) contém FOLLOW(B);

OBS: o símbolo \$ indica final da entrada (se comporta como se fosse um *token* indicando fim de arquivo)

Análise Sintática LL(1)

- Conjunto FOLLOW

- Observações sobre a construção dos conjuntos FOLLOW:

- ✓ O conjunto FOLLOW é formado apenas por símbolos **terminais**
 - ✓ O símbolo \$ indica final da entrada (se comporta como se fosse um *token*)

- ✓ ϵ **nunca** é um elemento do conjunto FOLLOW

(ϵ foi usado nos conjuntos FIRST apenas para marcar as cadeias que podem desaparecer)

- ✓ Os conjuntos FOLLOW são definidos apenas para os símbolos **não-terminais**

- ✓ A definição dos conjuntos FOLLOW funciona “à direita” das produções

- ✓ Regras que não têm não-terminais à direita nada acrescentam à construção dos conjuntos FOLLOW

Análise Sintática LL(1)

- Conjunto FOLLOW

```
Seqüência(símbolo-inicial) := {$} ;  
for cada não-terminal A # símbolo-inicial do Seqüência(A) := { } ;  
while houver alterações em algum conjunto de Seqüência do  
  for cada produção  $A \rightarrow X_1X_2...X_n$  do  
    for each  $X_i$  que for não-terminal do  
      adicione Primeiro( $X_{i+1}X_{i+2}...X_n$ ) - { $\epsilon$ } a Seqüência( $X_i$ )  
      (* Nota: se  $i=n$ , então  $X_{i+1}X_{i+2}...X_n = \epsilon$  *)  
      if  $\epsilon$  estiver em Primeiro ( $X_{i+1}X_{i+2}...X_n$ ) then  
        adicione Seqüência(A) a Seqüência( $X_i$ )
```

Figura 4.8 Algoritmo para a computação de conjuntos de Seqüência.

Análise Sintática LL(1)

- Conjunto FOLLOW

- Exemplo (4.12 – Louden)

- Considere novamente a gramática:

$\text{exp} \rightarrow \text{exp soma termo} \mid \text{termo}$

$\text{soma} \rightarrow + \mid -$

$\text{termo} \rightarrow \text{termo mult fator} \mid \text{fator}$

$\text{mult} \rightarrow *$

$\text{fator} \rightarrow (\text{exp}) \mid \text{NUM}$

$\text{Primeiro}(\text{exp}) = \{ (, \text{número} \}$

$\text{Primeiro}(\text{termo}) = \{ (, \text{número} \}$

$\text{Primeiro}(\text{fator}) = \{ (, \text{número} \}$

$\text{Primeiro}(\text{soma}) = \{ +, - \}$

$\text{Primeiro}(\text{mult}) = \{ * \}$

Análise Sintática LL(1)

Tabela 4.8 Computação de conjuntos de Seqüência para a gramática do Exemplo 4.12.

Regra gramatical	Passada 1	Passada 2
$exp \rightarrow exp \text{ soma}$ <i>termo</i>	Seqüência(exp) = { \$, +, - } Seqüência($soma$) = { (, número } Seqüência($termo$) = { \$, +, - }	Seqüência($termo$) = { \$, +, -, *,) }
$exp \rightarrow termo$		
$termo \rightarrow termo \text{ mult}$ <i>fator</i>	Seqüência($termo$) = { \$, +, -, * } Seqüência($mult$) = { (, número } Seqüência($fator$) = { \$, +, -, * }	Seqüência($fator$) = { \$, +, -, *,) }
$termo \rightarrow fator$		
$fator \rightarrow (exp)$	Seqüência(exp) = { \$, +, -,) }	

```

Seqüência(símbolo-inicial) := { $ } ;
for cada não-terminal A # símbolo-inicial do Seqüência(A) := { } ;
while houver alterações em algum conjunto de Seqüência do
  for cada produção  $A \rightarrow X_1 X_2 \dots X_n$  do
    for each  $X_i$  que for não-terminal do
      adicione Primeiro( $X_{i+1} X_{i+2} \dots X_n$ ) - {  $\epsilon$  } a Seqüência( $X_i$ )
      (* Nota: se  $i=n$ , então  $X_{i+1} X_{i+2} \dots X_n = \epsilon$  *)
      if  $\epsilon$  estiver em Primeiro ( $X_{i+1} X_{i+2} \dots X_n$ ) then
        adicione Seqüência(A) a Seqüência( $X_i$ )

```

```

Primeiro( $exp$ ) = { (, número }
Primeiro( $termo$ ) = { (, número }
Primeiro( $fator$ ) = { (, número }
Primeiro( $soma$ ) = { +, - }
Primeiro( $mult$ ) = { * }

```

Análise Sintática LL(1)

- Conjunto FOLLOW

—Exemplo:

- Considere a gramática:

declaração \rightarrow if-decl | outra

if-decl \rightarrow if (exp) declaração else-parte

else-parte \rightarrow else declaração | ε

exp \rightarrow 0 | 1

$\text{FIRST}(\text{declaração}) = \{\text{if}, \text{outra}\}$

$\text{FIRST}(\text{if-decl}) = \{\text{if}\}$

$\text{FIRST}(\text{else-parte}) = \{\text{else}, \varepsilon\}$

$\text{FIRST}(\text{exp}) = \{0, 1\}$

Análise Sintática LL(1)

- Conjunto FOLLOW

—Exemplo:

- $\text{FOLLOW}(\text{declaração}) = \{\$, \text{else}\}$
- $\text{FOLLOW}(\text{if-decl}) = \{\$, \text{else}\}$
- $\text{FOLLOW}(\text{else-parte}) = \{\$, \text{else}\}$
- $\text{FOLLOW}(\text{exp}) = \{)\}$

Análise Sintática LL(1)

- Conjunto FOLLOW

—Exercício:

- Considere a gramática:

$\text{decl-sequência} \rightarrow \text{decl decl-seq}'$

$\text{decl-seq}' \rightarrow ; \text{decl-sequência} \mid \varepsilon$

$\text{decl} \rightarrow s$

Exercício 2: obter os conjuntos FOLLOW para essa gramática

Análise Sintática LL(1)

- Construção da **Tabela Preditiva LL(1)**

- Repetir os dois passos a seguir para **cada não-terminal A e escolha de produção $A \rightarrow \alpha$**

- Para cada terminal a em $\text{FIRST}(\alpha)$, adicione $A \rightarrow \alpha$ a $M[A, a]$
 - Equivalente a: se $A \rightarrow \alpha$ e existe derivação $\alpha \Rightarrow^* a\beta$, então $M[A, a] = A \rightarrow \alpha$

- (eu consigo gerar o símbolo a pela aplicação da regra $A \rightarrow \alpha$)*

- Se ϵ pertencer a $\text{FIRST}(\alpha)$, para cada elemento a de $\text{FOLLOW}(A)$ adicione $A \rightarrow \alpha$ a $M[A, a]$

- Equivalente a: se $A \rightarrow \alpha$ e existem derivações $\alpha \Rightarrow^* \epsilon$ e $S\$ \Rightarrow^* \beta A a \gamma$, então $M[A, a] = A \rightarrow \alpha$

- (eu consigo gerar o símbolo a pela aplicação da regra $A \rightarrow \alpha$, pois mesmo anulando A em alguma derivação, eu garanto que é possível aparecer o a depois)*

Análise Sintática LL(1)

- Exemplo de construção de tabela preditiva (4.15 – Louden)

$exp \rightarrow termo\ exp'$
 $exp' \rightarrow soma\ termo\ exp' \mid \varepsilon$
 $soma \rightarrow + \mid -$
 $termo \rightarrow fator\ termo'$
 $termo' \rightarrow mult\ fator\ termo' \mid \varepsilon$
 $mult \rightarrow *$
 $fator \rightarrow (exp) \mid \text{número}$

$\text{Primeiro}(exp) = \{ (, \text{número} \}$
 $\text{Primeiro}(exp') = \{ +, -, \varepsilon \}$
 $\text{Primeiro}(soma) = \{ +, - \}$
 $\text{Primeiro}(termo) = \{ (, \text{número} \}$
 $\text{Primeiro}(termo') = \{ *, \varepsilon \}$
 $\text{Primeiro}(mult) = \{ * \}$
 $\text{Primeiro}(fator) = \{ (, \text{número} \}$

$\text{Seqüência}(exp) = \{ \$,) \}$
 $\text{Seqüência}(exp') = \{ \$,) \}$
 $\text{Seqüência}(soma) = \{ (, \text{número} \}$
 $\text{Seqüência}(termo) = \{ \$,), +, - \}$
 $\text{Seqüência}(termo') = \{ \$,), +, - \}$
 $\text{Seqüência}(mult) = \{ (, \text{número} \}$
 $\text{Seqüência}(fator) = \{ \$,), +, -, * \}$

Análise Sintática LL(1)

Para cada não terminal A e produção $A \rightarrow \alpha$:

$\text{Primeiro}(exp) = \{ (, \text{número} \}$
 $\text{Primeiro}(exp') = \{ +, -, \epsilon \}$
 $\text{Primeiro}(soma) = \{ +, - \}$
 $\text{Primeiro}(termo) = \{ (, \text{número} \}$
 $\text{Primeiro}(termo') = \{ *, \epsilon \}$
 $\text{Primeiro}(mult) = \{ * \}$
 $\text{Primeiro}(fator) = \{ (, \text{número} \}$

$\text{Seqüência}(exp) = \{ \$,) \}$
 $\text{Seqüência}(exp') = \{ \$,) \}$
 $\text{Seqüência}(soma) = \{ (, \text{número} \}$
 $\text{Seqüência}(termo) = \{ \$,), +, - \}$
 $\text{Seqüência}(termo') = \{ \$,), +, - \}$
 $\text{Seqüência}(mult) = \{ (, \text{número} \}$
 $\text{Seqüência}(fator) = \{ \$,), +, -, * \}$

Para cada terminal a em $\text{FIRST}(\alpha)$, adicione $A \rightarrow \alpha$ a $M[A, a]$

Se ϵ pertencer a $\text{FIRST}(\alpha)$, para cada elemento a de $\text{FOLLOW}(A)$ adicione $A \rightarrow \alpha$ a $M[A, a]$

Tabela 4.4 Tabela de análise sintática LL(1) para a gramática da Figura 4.4.

$M[N, T]$	(número)	+	-	*	\$
exp	$exp \rightarrow$ $termo\ exp'$	$exp \rightarrow$ $termo\ exp'$					
exp'			$exp' \rightarrow \epsilon$	$exp' \rightarrow$ $soma$ $termo\ exp'$	$exp' \rightarrow$ $soma$ $termo\ exp'$		$exp' \rightarrow \epsilon$
$soma$				$soma \rightarrow$ +	$soma \rightarrow$ -		
$termo$	$termo \rightarrow$ $fator$ $termo'$	$termo \rightarrow$ $fator$ $termo'$					
$termo'$			$termo' \rightarrow$ ϵ	$termo' \rightarrow \epsilon$	$termo' \rightarrow \epsilon$	$termo' \rightarrow$ $mult$ $fator$ $termo'$	$termo' \rightarrow$ ϵ
$mult$						$mult \rightarrow$ *	
$fator$	$fator \rightarrow$ (exp)	$fator \rightarrow$ número					

$exp \rightarrow termo\ exp'$
 $exp' \rightarrow soma\ termo\ exp' \mid \epsilon$
 $soma \rightarrow + \mid -$
 $termo \rightarrow fator\ termo'$
 $termo' \rightarrow mult\ fator\ termo' \mid \epsilon$
 $mult \rightarrow *$
 $fator \rightarrow (exp) \mid \text{número}$

Análise Sintática LL(1)

- Tabela preditiva

—Exemplo:

- Considere a gramática:

declaração \rightarrow if-decl | outra

if-decl \rightarrow if (exp) declaração else-parte

else-parte \rightarrow else declaração | ε

exp \rightarrow 0 | 1

FIRST(declaração) = {if, outra}

FIRST(if-decl) = {if}

FIRST(else-parte) = {else, ε }

FIRST(exp) = {0, 1}

FOLLOW(declaração) = {\$, else}

FOLLOW(if-decl) = {\$, else}

FOLLOW(else-parte) = {\$, else}

FOLLOW(exp) = {)}

Análise Sintática LL(1)

- Tabela preditiva

M(N,T)	outra	if	else	()	0	1	\$
declaração	declaração → outra	declaração → if-decl						
if-decl		If-decl → if (exp) declaração else-parte						
else-parte			else-parte → else declaração else-parte → ϵ					else-parte → ϵ
exp						exp → 0	exp → 1	

Análise Sintática LL(1)

- Tabela preditiva

—Exemplo:

- É possível notar na tabela a presença de duas regras na célula $M[\text{else-parte}, \text{else}]$
 - Corresponde à **ambiguidade do else pendente**
 - Quando usar essa tabela, é preciso aplicar regra de eliminação de ambiguidade: sempre prefira a regra que gera a marca corrente de verificação à frente (aninhamento mais próximo)
 -
 - Usaremos agora a tabela para analisar uma cadeia (atenção às operações de substituição de não-terminais e de casamento de cadeias)

Análise Sintática LL(1)

- Exemplo: if(0) if(1) outra else outra

Pilha de análise sintática	Entrada	Ação
\$ decl	If (0) if (1) outra else outra \$	decl → if-decl
\$ if-decl	If (0) if (1) outra else outra \$	If-decl → if (exp) decl else-parte
\$ else-parte decl) exp (if	If (0) if (1) outra else outra \$	casa
\$ else-parte decl) exp ((0) if (1) outra else outra \$	casa
\$ else-parte decl) exp	0) if (1) outra else outra \$	exp → 0
\$ else-parte decl) 0	0) if (1) outra else outra \$	casa
\$ else-parte decl)) if (1) outra else outra \$	casa
\$ else-parte decl	if (1) outra else outra \$	decl → if-decl
\$ else-parte if-decl	if (1) outra else outra \$	If-decl → if (exp) decl else-parte
\$ else-parte else-parte decl) exp (if	if (1) outra else outra \$	casa

Análise Sintática LL(1)

Pilha de análise sintática	Entrada	Ação
\$ else-parte else-parte decl) exp ((1) outra else outra \$	casa
\$ else-parte else-parte decl) exp	1) outra else outra \$	exp \rightarrow 1
\$ else-parte else-parte decl) 1	1) outra else outra \$	casa
\$ else-parte else-parte decl)) outra else outra \$	casa
\$ else-parte else-parte decl	outra else outra \$	decl \rightarrow outra
\$ else-parte else-parte outra	outra else outra \$	casa
\$ else-parte else-parte	else outra \$	else-parte \rightarrow else decl
\$ else-parte decl else	else outra \$	casa
\$ else-parte decl	outra \$	decl \rightarrow outra
\$ else-parte outra	outra \$	casa
\$ else-parte	\$	else-parte $\rightarrow \epsilon$
\$	\$	aceita

Análise Sintática LL(1)

- Tabela preditiva

—Exercício:

- Considere a gramática:

$\text{decl-sequência} \rightarrow \text{decl decl-seq}'$

$\text{decl-seq}' \rightarrow ; \text{decl-sequência} \mid \varepsilon$

$\text{decl} \rightarrow s$

Exercício 3: obter a tabela de análise sintática LL(1) dessa gramática

Exercício 4: analisar a entrada $s;s$

Análise Sintática LL(1)

- Bibliografia consultada

LOUDEN, K. C. **Compiladores: princípios e práticas**. São Paulo: Pioneira Thompson Learning, 2004 (Cap. 4)

MERINO, M. **Notas de Aulas - Compiladores**, UNIMEP, 2006.