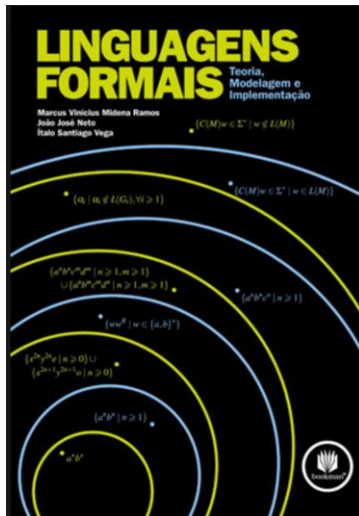


TEORIA: ANÁLISES DESCENDENTE LL(1)



Nossos **objetivos** nesta aula são:

- conhecer o processo de geração de tabelas LL(1)
- pratica com geração de tabelas de análise LL(1)



Para esta semana, usamos como referência as **Seções 4.1 (Gramáticas Livres de Contexto) até 4.4 (Ambigüidade)** do nosso livro da referência básica:

RAMOS, M.V.M., JOSÉ NETO, J., VEJA, I.S. **Linguagens Formais: Teoria, Modelagem e Implementação**. Porto Alegre: Bookman, 2009.

Não deixem de ler estas seções depois desta aula!

CONJUNTOS NULLABE, FIRST e FOLLOW

- Para se construir uma tabela de análise LL(1), vamos utilizar três procedimentos computacionais aplicados a uma produção:

nullable(X)	:	verifica se X produz a cadeia vazia (ϵ)
FIRST(X)	:	conjunto de terminais que iniciam as palavras derivadas de X
FOLLOW(X)	:	conjunto de terminais que podem imediatamente seguir X

Em particular, os conjuntos FIRST e FOLLOW ajudam a determinar qual regra será aplicada na ocorrência de um determinado terminal.

- **Exemplo** - vamos considerar a seguinte gramática:

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

Para esta gramática, teremos os seguintes valores para nullable, FIRST e FOLLOW:

	nullable	FIRST	FOLLOW
X	T	a c ϵ	a c d
Y	T	c ϵ	a c d
Z	F	a c d	\$

- Observe que somente **terminais** fazem parte dos conjuntos FIRST e FOLLOW. **O conjunto FIRST pode, eventualmente, incluir a cadeia vazia. FOLLOW nunca pode conter a cadeia vazia.**

Algoritmo iterativo para computar Nullable, FIRST e FOLLOW:

1. Inicializar nullable como F para todas as regras e FIRST/FOLLOW como conjuntos vazios
2. Para cada **símbolo terminal** Z
3. $FIRST[Z] = \{Z\}$
4. **Repita**
5. Para cada produção $X \rightarrow Y_1 \dots Y_k$
6. Se $Y_1 \dots Y_k$ são todos nullable (ou se $k=0$)
7. **nullable**[X] = T
8. Para cada i de 1 até k
9. Se $Y_1 \dots Y_{i-1}$ são todos nullable (ou se $i=1$)
10. **FIRST**[X] = **FIRST**[X] \cup **FIRST**[Y_i]
11. Se $Y_{i+1} \dots Y_k$ são todos nullable (ou se $i=k$)
12. **FOLLOW**[Y_i] = **FOLLOW**[Y_i] \cup **FOLLOW**[X]
13. Para cada j de i+1 até k
14. Se $Y_{i+1} \dots Y_{j-1}$ são todos nullable (ou se $i+1=k$)
15. **FOLLOW**[Y_j] = **FOLLOW**[Y_j] \cup **FIRST**[Y_j]
16. **Até que nullable, FIRST e FOLLOW não se alterem**

Para facilitar a execução deste algoritmo, utilizamos uma tabela como mostrado abaixo:

	nullable	FIRST	FOLLOW
X			
Y			
Z			

Na primeira coluna, colocamos todos os símbolos não-terminais e, nas demais colunas, os valores para nullable, FIRST e FOLLOW.

EXERCÍCIO TUTORIADO

Calcular os valores de nullable, FIRST e FOLLOW para a seguinte gramática:

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

	nullable	FIRST	FOLLOW
X			
Y			
Z			

EXERCÍCIO COM DISCUSSÃO EM DUPLAS

Calcular os valores de nullable, FIRST e FOLLOW para a seguinte gramática:

$S \rightarrow cAa\$$

$A \rightarrow cB \mid B$

$B \rightarrow bcB \mid \varepsilon$

	nullable	FIRST	FOLLOW
S			
A			
B			

TABELAS DE ANÁLISE LL (1)

Para se montar uma tabela preditiva descendente, colocamos os símbolos não-terminais nas linhas e os terminais nas colunas. **Colocamos a produção $X \rightarrow Y$ na linha X e coluna T se e somente se $T \in \text{FIRST}[Y]$. Se Y for nullable, colocamos também a produção $X \rightarrow Y$ na linha X e coluna T para cada $T \in \text{FOLLOW}[X]$.**

EXERCÍCIO TUTORIADO

Construa a tabela preditiva descendente para a gramática abaixo e seus conjuntos nullable, FIRST e FOLLOW:

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

	nullable	FIRST	FOLLOW
X	T	a c ϵ	a c d
Y	T	c ϵ	a c d
Z	F	a c d	\$

	a	c	d	\$
X				
Y				
Z				

Esta tabela apresenta um pequeno problema. Há entradas com mais de uma regra que pode ser aplicadas: **isto ocorre porque esta gramática é ambígua**. Para verificar isto, basta encontrar duas árvores para a entrada d. **Entradas não preenchidas significam ERRO.**

Gramáticas cujas tabelas preditivas descendentes **não contém entradas duplicadas** são chamadas de **gramáticas LL(1)**.

EXERCÍCIO COM DISCUSSÃO EM DUPLAS

Montar a tabela preditiva descendente para a gramática abaixo. Ela é uma tabela do tipo LL(1) ?

$S \rightarrow cAa\$$

$A \rightarrow cB \mid B$

$B \rightarrow bcB \mid \varepsilon$

EXERCÍCIOS EXTRA-CLASSE

1. Utilizando a tabela LL(1) construída em classe

	a	b	c	\$
S	ERRO	ERRO	$S \rightarrow cAa$	ERRO
A	$A \rightarrow B$	$A \rightarrow B$	$A \rightarrow cB$	ERRO
B	$B \rightarrow \epsilon$	$B \rightarrow bcB$	ERRO	ERRO

Mostre a análise para a entrada **cbca\$**:

PILHA	ENTRADA	AÇÃO

2. Considere-se a GLC mostrada abaixo:

$S \rightarrow E\$$

$E \rightarrow TE'$

$E' \rightarrow +TE'$

$E' \rightarrow -TE'$

$E' \rightarrow \varepsilon$

$T \rightarrow FT'$

$T' \rightarrow *FT'$

$T' \rightarrow /FT'$

$T' \rightarrow \varepsilon$

$F \rightarrow id$

$F \rightarrow num$

$F \rightarrow (E)$

- Construa os conjuntos nullable, FIRST e FOLLOW
- Construa a tabela preditiva descendente para esta gramática. Esta tabela é do tipo LL(1) ?