

## Lista de Exercícios 7: Resolução

Guilherme de Abreu (nUSP: 12543033)

Hélio Cardoso (nUSP: 10310227)

Laura Camargos (nUSP: 13692334)

Sandy Dutra (nUSP: 12544570)

Theo dos Santos (nUSP: 10691331)

14 de abril de 2025

**Exercício 1:** Criar uma tabela sintática para a gramática abaixo.

$S \rightarrow AS'$

$S' \rightarrow S \mid \lambda$

$A \rightarrow a \mid b \mid \lambda$

**Resolução:**

Iniciamos pela construção da Tabela de Primeiros e Seguidores:

Regra	Primeiro	Seguidor
$S \rightarrow AS'$	$a, b, +, \lambda$	$\lambda$
$S' \rightarrow S \mid \lambda$	$+, \lambda$	$\lambda$
$A \rightarrow a \mid b \mid \lambda$	$a, b, \lambda$	$+, \lambda$

Com base nesta, determinamos as regras de transição para obtenção dos símbolos terminais a partir dos terminais, que orienta a análise preditiva não recursiva:

	<b>a</b>	<b>b</b>	<b>+</b>	<b><math>\lambda</math></b>
$S$	$S \rightarrow AS'$	$S \rightarrow AS'$	$S \rightarrow AS'$	$S \rightarrow AS'$
$S'$	$S' \rightarrow +S$		$S' \rightarrow +S$	$S' \rightarrow \lambda$
$A$	$A \rightarrow a$	$A \rightarrow b$		$A \rightarrow \lambda$

**Exercício 2:** Construa a tabela sintática para a gramática abaixo e reconheça a cadeia  $id + id \cdot id$  utilizando análise sintática preditiva não recursiva. A cada passo, mostre o estado da pilha, da cadeia de entrada e a regra utilizada na derivação. Mostre a implementação deste analisador sintático.

$E \rightarrow TE'$   
 $E' \rightarrow +TE' \mid \lambda$   
 $T \rightarrow FT'$   
 $T' \rightarrow \cdot FT' \mid \lambda$   
 $F \rightarrow \mid id$

**Resolução:**

Seguindo os passos vistos no exercício anterior, temos:

Regra	Primeiro	Seguidor
$E \rightarrow TE'$	$(, id$	$), \lambda$
$E' \rightarrow +TE' \mid \lambda$	$+, \lambda$	$), \lambda$
$T \rightarrow FT'$	$(, id$	$+, \lambda$
$T' \rightarrow \cdot FT' \mid \lambda$	$\cdot, \lambda$	$+, \lambda$
$F \rightarrow (E) \mid id$	$(, id$	$\cdot, \lambda$

E a seguinte tabela sintática:

	$($	$)$	$+$	$\cdot$	<b>id</b>	$\lambda$
$E$	$\rightarrow TE'$				$\rightarrow TE'$	
$E'$		$\rightarrow \lambda$	$\rightarrow +TE'$			$\rightarrow \lambda$
$T$	$\rightarrow FT'$		$\rightarrow FT'$		$\rightarrow FT'$	
$T'$			$\rightarrow \lambda$	$\rightarrow FT'$		$\rightarrow \lambda$
$F$	$\rightarrow (E)$				$\rightarrow id$	

O que nos leva ao seguinte processo de análise sintática não recursivo:

Stack	String	Regra
$\lambda E$	$id + id \cdot id \lambda$	$E \rightarrow TE'$
$\lambda E' T$	$id + id \cdot id \lambda$	$T \rightarrow FT'$
$\lambda E' T' F$	$id + id \cdot id \lambda$	$F \rightarrow id$
$\lambda E' T'$	$+id \cdot id \lambda$	$T' \rightarrow \lambda$
$\lambda E'$	$+id \cdot id \lambda$	$E' \rightarrow +TE'$
$\lambda E' T$	$id \cdot id \lambda$	$T \rightarrow FT'$
$\lambda E' T' F$	$id \cdot id \lambda$	$F \rightarrow id$
$\lambda E' T'$	$\cdot id \lambda$	$T' \rightarrow \cdot FT'$
$\lambda E' T' F$	$id \lambda$	$F \rightarrow id$
$\lambda E' T'$	$\lambda$	$T' \rightarrow \lambda$
$\lambda E'$	$\lambda$	$E' \rightarrow \lambda$
$\lambda$	$\lambda$	Sucesso