

# Análise sintática

Função, interação com o compilador  
Análise descendente e ascendente  
Especificação e reconhecimento de cadeias de tokens válidas  
Implementação  
Tratamento de erros

Prof. Thiago A. S. Pardo  
taspardo@icmc.usp.br

1

# Análise sintática

- *Top-down* ou descendente
  - Com retrocesso: por tentativa e erro
  - Preditiva: para gramáticas LL(1)
    - Recursiva
    - Não recursiva

2

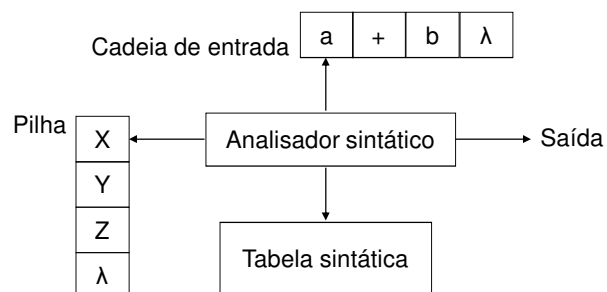
## ASD preditiva não recursiva

- **Alternativa** para os procedimentos recursivos
  - Gramáticas LL(1)
- O que é a recursividade? Como eliminá-la?

3

## ASD preditiva não recursiva

- **Idéia geral**
  - A recursão é substituída pelo uso de uma pilha onde os símbolos sendo expandidos são armazenados
  - Para determinar qual regra gramatical aplicar, consulta-se uma tabela sintática

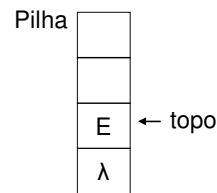


4

## ASD preditiva não recursiva

### ■ Funcionamento

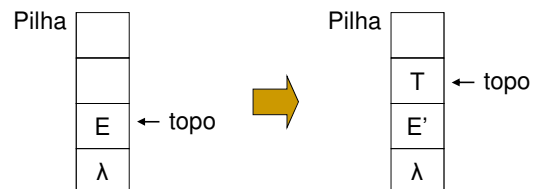
- Um símbolo não terminal a ser expandido é empilhado

$$\begin{aligned} \langle E \rangle &::= \langle T \rangle \langle E' \rangle \\ \langle E' \rangle &::= + \langle E \rangle \mid \lambda \\ \langle T \rangle &::= a \mid b \end{aligned}$$


5

## ASD preditiva não recursiva

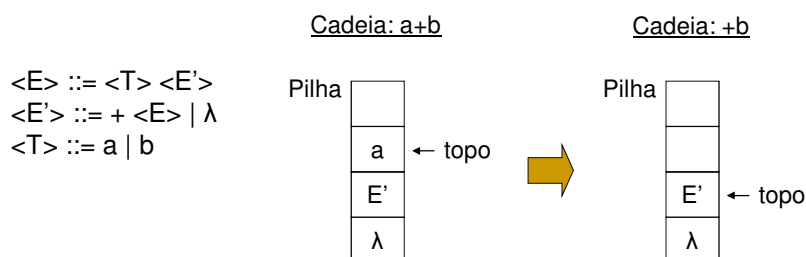
- Ao expandir um não terminal no topo da pilha, ele é desempilhado e seu lado direito da regra gramatical é empilhado (em sentido inverso) para expansão
  - O sentido inverso garante a ordem natural de expansão da esquerda para a direita

$$\begin{aligned} \langle E \rangle &::= \langle T \rangle \langle E' \rangle \\ \langle E' \rangle &::= + \langle E \rangle \mid \lambda \\ \langle T \rangle &::= a \mid b \end{aligned}$$


6

## ASD preditiva não recursiva

- Quando um símbolo terminal estiver no topo da pilha e esse mesmo símbolo estiver no início da cadeia sendo reconhecida, o terminal é desempilhado e o símbolo inicial da cadeia consumido



7

## ASD preditiva não recursiva

- Considere  $X$  o topo da pilha e  $a$  o símbolo inicial da cadeia de entrada
- Possibilidades durante a análise
  - Se  $X=a=\lambda$ , então o analisador termina a análise com sucesso
  - Se  $X=a \neq \lambda$ , então o analisador desempilha  $X$  e consome o símbolo inicial da cadeia
  - Se  $X$  é não terminal, então o analisador procura na tabela sintática a regra de  $X$  que produz o símbolo inicial da cadeia e empilha seu lado direito (em sentido inverso)
  - Se  $X$  é terminal e é diferente de  $a$  ou se  $X$  é não terminal e não há regra cuja derivação produza  $a$ , então um erro ocorreu

8

## ASD preditiva não recursiva

### ■ Algoritmo de análise sintática

empilhe um símbolo delimitador ( $\lambda$ ) e o símbolo inicial da gramática;

concatene ao final da cadeia um símbolo delimitador ( $\lambda$ );

faça  $ip$  apontar para o primeiro símbolo da cadeia;

repetir

$X$ =símbolo no topo da pilha;

$a$ =símbolo apontado por  $ip$ ;

se ( $X$  for um terminal ou o símbolo delimitador) então

se  $X=a$  então desempilhar  $X$ ;

avançar  $ip$ ;

senão ERRO;

senão /\* $X$  é um não terminal\*/

se (existe na tabela sintática uma regra de  $X$  que produza  $a$ ) então

desempilhar  $X$ ;

empilhar em sentido inverso o lado direito da regra selecionada;

senão ERRO;

até que  $X$ =símbolo delimitador ou  $a$ =símbolo delimitador

/\*pilha está vazia ou fim de cadeia\*/

## ASD preditiva não recursiva

### ■ Exemplo

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$

$\langle E' \rangle ::= + \langle E \rangle \mid \lambda$

$\langle T \rangle ::= a \mid b$

Tabela sintática

	a	b	+	$\lambda$
E	$E \rightarrow TE'$	$E \rightarrow TE'$		
E'			$E' \rightarrow +E$	$E' \rightarrow \lambda$
T	$T \rightarrow a$	$T \rightarrow b$		

Reconhecer a+b

Pilha	Cadeia	Regra

## ASD preditiva não recursiva

### Exemplo

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$   
 $\langle E' \rangle ::= + \langle E \rangle \mid \lambda$   
 $\langle T \rangle ::= a \mid b$

Tabela sintática

	a	b	+	$\lambda$
E	$E \rightarrow TE'$	$E \rightarrow TE'$		
E'			$E' \rightarrow +E$	$E' \rightarrow \lambda$
T	$T \rightarrow a$	$T \rightarrow b$		

Reconhecer a+b

Pilha	Cadeia	Regra
$\lambda E$	a+b $\lambda$	$E \rightarrow TE'$
$\lambda E'T$	a+b $\lambda$	$T \rightarrow a$
$\lambda E'a$	a+b $\lambda$	---
$\lambda E'$	+b $\lambda$	$E' \rightarrow +E$
$\lambda E+$	+b $\lambda$	---
$\lambda E$	b $\lambda$	$E \rightarrow TE'$
$\lambda E'T$	b $\lambda$	$T \rightarrow b$
$\lambda E'b$	b $\lambda$	---
$\lambda E'$	$\lambda$	$E' \rightarrow \lambda$
$\lambda$	$\lambda$	SUCESSO

11

## ASD preditiva não recursiva

### Exemplo

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$   
 $\langle E' \rangle ::= + \langle E \rangle \mid \lambda$   
 $\langle T \rangle ::= a \mid b$

Tabela sintática

	a	b	+	$\lambda$
E	$E \rightarrow TE'$	$E \rightarrow TE'$		
E'			$E' \rightarrow +E$	$E' \rightarrow \lambda$
T	$T \rightarrow a$	$T \rightarrow b$		

Reconhecer a\*b

12

## ASD preditiva não recursiva

### Exemplo

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$   
 $\langle E' \rangle ::= + \langle E \rangle \mid \lambda$   
 $\langle T \rangle ::= a \mid b$

Tabela sintática

	a	b	+	$\lambda$
E	$E \rightarrow TE'$	$E \rightarrow TE'$		
E'			$E' \rightarrow +E$	$E' \rightarrow \lambda$
T	$T \rightarrow a$	$T \rightarrow b$		

Pilha	Cadeia	Regra
$\lambda E$	$a^*b\lambda$	$E \rightarrow TE'$
$\lambda E'T$	$a^*b\lambda$	$T \rightarrow a$
$\lambda E'a$	$a^*b\lambda$	---
$\lambda E'$	$*b\lambda$	ERRO

A cadeia não pertence à linguagem!

Reconhecer  $a^*b$

13

## ASD preditiva não recursiva

### Exercício: reconheça a cadeia 01012

$\langle S \rangle ::= 0\langle A \rangle \mid 1\langle B \rangle$   
 $\langle A \rangle ::= 1\langle B \rangle \mid 2$   
 $\langle B \rangle ::= 0\langle A \rangle \mid 2$

Tabela sintática

	0	1	2
S	$S \rightarrow 0A$	$S \rightarrow 1B$	
A		$A \rightarrow 1B$	$A \rightarrow 2$
B	$B \rightarrow 0A$		$B \rightarrow 2$

14

## ASD preditiva não recursiva

- Exercício: reconheça a cadeia 01012

$\langle S \rangle ::= 0\langle A \rangle \mid 1\langle B \rangle$

$\langle A \rangle ::= 1\langle B \rangle \mid 2$

$\langle B \rangle ::= 0\langle A \rangle \mid 2$

Tabela sintática

	0	1	2
S	S→0A	S→1B	
A		A→1B	A→2
B	B→0A		B→2

Pilha	Cadeia	Regra
λS	01012λ	S→0A
λA0	01012λ	---
λA	1012λ	A→1B
λB1	1012λ	---
λB	012λ	B→0A
λA0	012λ	---
λA	12λ	A→1B
λB1	12λ	---
λB	2λ	B→2
λ2	2λ	---
λ	λ	SUCESSO

15

## ASD preditiva não recursiva

- Como construir a tabela sintática?

$\langle S \rangle ::= 0\langle A \rangle \mid 1\langle B \rangle$

$\langle A \rangle ::= 1\langle B \rangle \mid 2$

$\langle B \rangle ::= 0\langle A \rangle \mid 2$

Tabela sintática

	0	1	2
S	S→0A	S→1B	
A		A→1B	A→2
B	B→0A		B→2

- De acordo com o exemplo, para um não terminal  $X$  e um terminal  $a$ , a tabela indica a regra de  $X$  cujo conjunto Primeiro contém  $a$ 
  - Atenção: λ só aparece na tabela se aparecer na gramática

16



## ASD preditiva não recursiva

- Exercício: construa a tabela sintática

$$\langle S \rangle ::= 0\langle A \rangle \mid \langle B \rangle$$

$$\langle A \rangle ::= a\langle A \rangle \mid \lambda$$

$$\langle B \rangle ::= b$$

17

## ASD preditiva não recursiva

- Exercício: reconheça a cadeia **0aa**

$$\langle S \rangle ::= 0\langle A \rangle \mid \langle B \rangle$$

$$\langle A \rangle ::= a\langle A \rangle \mid \lambda$$

$$\langle B \rangle ::= b$$

Tabela sintática

	0	a	b	$\lambda$
S	S→0A		S→B	
A		A→aA		A→ $\lambda$
B			B→b	

18

## ASD preditiva não recursiva

- Exercício: reconheça a cadeia **0aa**

$\langle S \rangle ::= 0\langle A \rangle \mid \langle B \rangle$

$\langle A \rangle ::= a\langle A \rangle \mid \lambda$

$\langle B \rangle ::= b$

Tabela sintática

	0	a	b	$\lambda$
S	$S \rightarrow 0A$		$S \rightarrow B$	
A		$A \rightarrow aA$		$A \rightarrow \lambda$
B			$B \rightarrow b$	

Pilha	Cadeia	Regra
$\lambda S$	0aa $\lambda$	$S \rightarrow 0A$
$\lambda A 0$	0aa $\lambda$	---
$\lambda A$	aa $\lambda$	$A \rightarrow aA$
$\lambda A a$	aa $\lambda$	---
$\lambda A$	a $\lambda$	$A \rightarrow aA$
$\lambda A a$	a $\lambda$	---
$\lambda A$	$\lambda$	$A \rightarrow \lambda$
$\lambda$	$\lambda$	SUCESSO

19

## ASD preditiva não recursiva

- Exercício: construa a tabela e reconheça a cadeia **0a1**

$\langle S \rangle ::= 0\langle A \rangle 1 \mid \langle B \rangle$

$\langle A \rangle ::= a\langle A \rangle \mid \lambda$

$\langle B \rangle ::= b$

20

## ASD preditiva não recursiva

- Exercício: construa a tabela e reconheça a cadeia 0a1

$\langle S \rangle ::= 0\langle A \rangle 1 \mid \langle B \rangle$

$\langle A \rangle ::= a\langle A \rangle \mid \lambda$

$\langle B \rangle ::= b$

Tabela sintática

	0	1	a	b	$\lambda$
S	S→0A1			S→B	
A			A→aA		A→ $\lambda$
B				B→b	

Pilha	Cadeia	Regra
$\lambda S$	0a1 $\lambda$	S→0A1
$\lambda 1 A 0$	0a1 $\lambda$	---
$\lambda 1 A$	a1 $\lambda$	A→aA
$\lambda 1 A a$	a1 $\lambda$	---
$\lambda 1 A$	1 $\lambda$	ERRO

Por que aconteceu o erro?  
A cadeia não faz parte da  
linguagem? Ou a tabela está  
errada?

21

## ASD preditiva não recursiva

- Exercício: construa a tabela e reconheça a cadeia 0a1

$\langle S \rangle ::= 0\langle A \rangle 1 \mid \langle B \rangle$

$\langle A \rangle ::= a\langle A \rangle \mid \lambda$

$\langle B \rangle ::= b$

Tabela sintática

	0	1	a	b	$\lambda$
S	S→0A1			S→B	
A		A→ $\lambda$	A→aA		A→ $\lambda$
B				B→b	

Pilha	Cadeia	Regra
$\lambda S$	0a1 $\lambda$	S→0A1
$\lambda 1 A 0$	0a1 $\lambda$	---
$\lambda 1 A$	a1 $\lambda$	A→aA
$\lambda 1 A a$	a1 $\lambda$	---
$\lambda 1 A$	1 $\lambda$	ERRO

Por que aconteceu o erro?  
A cadeia não faz parte da  
linguagem? Ou a tabela está  
errada?

Para se chegar ao terminal 1, A deve produzir  $\lambda$ . Portanto, a regra A→ $\lambda$  deve ser adicionada na tabela para a combinação de A com 1

22

## ASD preditiva não recursiva

- Regras para construção da tabela sintática
  1. Para cada produção  $A \rightarrow \alpha$  da gramática, execute os passos 2 e 3 abaixo
  2. Para cada terminal  $a$  em  $\text{Primeiro}(\alpha)$ , adicione  $A \rightarrow \alpha$  em  $T[A, a]$
  3. Se  $\lambda$  estiver em  $\text{Primeiro}(\alpha)$ , adicione  $A \rightarrow \alpha$  em  $T[A, b]$  para cada terminal  $b$  em  $\text{Seguidor}(A)$
  4. Faça cada entrada indefinida da tabela indicar erro

23

## ASD preditiva não recursiva

- Exercício: construir a tabela sintática para a gramática abaixo

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$   
 $\langle E' \rangle ::= + \langle T \rangle \langle E' \rangle \mid \lambda$   
 $\langle T \rangle ::= \langle F \rangle \langle T' \rangle$   
 $\langle T' \rangle ::= * \langle F \rangle \langle T' \rangle \mid \lambda$   
 $\langle F \rangle ::= (E) \mid \text{id}$

24

## ASD preditiva não recursiva

- Exercício: construir a tabela sintática para a gramática abaixo

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$   
 $\langle E' \rangle ::= + \langle T \rangle \langle E' \rangle \mid \lambda$   
 $\langle T \rangle ::= \langle F \rangle \langle T' \rangle$   
 $\langle T' \rangle ::= * \langle F \rangle \langle T' \rangle \mid \lambda$   
 $\langle F \rangle ::= (E) \mid id$

Tabela sintática

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

25

## ASD preditiva não recursiva

- Exercício: reconhecer a cadeia **id+id\*id**

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$   
 $\langle E' \rangle ::= + \langle T \rangle \langle E' \rangle \mid \lambda$   
 $\langle T \rangle ::= \langle F \rangle \langle T' \rangle$   
 $\langle T' \rangle ::= * \langle F \rangle \langle T' \rangle \mid \lambda$   
 $\langle F \rangle ::= (E) \mid id$

Tabela sintática

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

26

## ASD preditiva não recursiva

- Considere a gramática

$$\langle S \rangle ::= \text{if } \langle E \rangle \text{ then } \langle S \rangle \mid \text{if } \langle E \rangle \text{ then } \langle S \rangle \text{ else } \langle S \rangle$$

Transformando em LL(1):

$$\langle S \rangle ::= \text{if } \langle E \rangle \text{ then } \langle S \rangle \langle S' \rangle$$

$$\langle S' \rangle ::= \text{else } \langle S \rangle \mid \lambda$$

- Reconhecer a cadeia **if E1 then if E2 then C1 else C2**

27

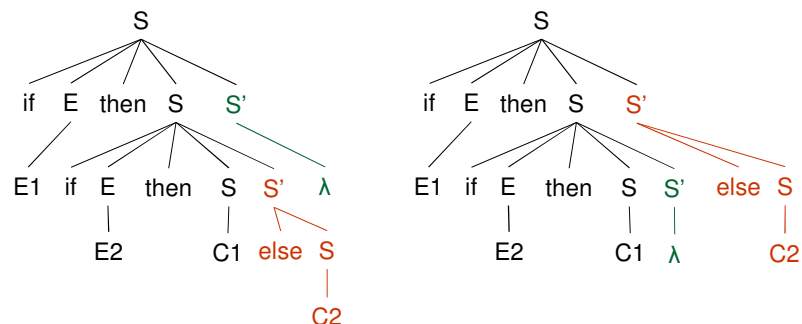
## ASD preditiva não recursiva

- Duas possibilidades: gramática ambígua

$$\langle S \rangle ::= \text{if } \langle E \rangle \text{ then } \langle S \rangle \langle S' \rangle$$

$$\langle S' \rangle ::= \text{else } \langle S \rangle \mid \lambda$$

if E1 then if E2 then C1 else C2



28

## ASD preditiva não recursiva

- Construir tabela sintática

$$\begin{aligned} \langle S \rangle &::= \text{if } \langle E \rangle \text{ then } \langle S \rangle \langle S' \rangle \\ \langle S' \rangle &::= \text{else } \langle S \rangle \mid \lambda \end{aligned}$$

29

## ASD preditiva não recursiva

- Construir tabela sintática

$$\begin{aligned} \langle S \rangle &::= \text{if } \langle E \rangle \text{ then } \langle S \rangle \langle S' \rangle \\ \langle S' \rangle &::= \text{else } \langle S \rangle \mid \lambda \end{aligned}$$

$$\begin{aligned} \text{Primeiro}(S) &= \{\text{if}\} \\ \text{Primeiro}(S') &= \{\text{else}, \lambda\} \\ \text{Seguidor}(S) &= \{\lambda\} \cup \text{Primeiro}(S') \cup \text{Seguidor}(S') = \{\text{else}, \lambda\} \\ \text{Seguidor}(S') &= \text{Seguidor}(S) = \{\text{else}, \lambda\} \end{aligned}$$

	if	then	else	$\lambda$
S	$S \rightarrow \text{if } E \text{ then } S S'$			
S'			$S' \rightarrow \text{else } S$ $S' \rightarrow \lambda$	$S' \rightarrow \lambda$

Como resolver isso?

30

## ASD preditiva não recursiva

- Solução

- Escolhe-se uma das regras

	if	then	else	$\lambda$
S	S→if E then S S'			
S'			S'→else S	S'→ $\lambda$

- Resultado: o *else* se relaciona ao *then* mais próximo