

Capítulo III

Histórico de revisões:

2010-2014 – Carlos M. Fonseca

2008-2010 – Luís Macedo

Anos anteriores – F. Amílcar Cardoso

Nullable, FIRST, FOLLOW, Tabela de Parsing

2

□ Exemplo:

$$\begin{aligned} Z &\rightarrow d \\ Z &\rightarrow X Y Z \end{aligned}$$
$$\begin{aligned} Y &\rightarrow \varepsilon \\ Y &\rightarrow c \end{aligned}$$
$$\begin{aligned} X &\rightarrow Y \\ X &\rightarrow a \end{aligned}$$

Nullable, FIRST, FOLLOW, Tabela de Parsing

3

□ Exemplo:

$Z \rightarrow d$
 $Z \rightarrow X Y Z$

$Y \rightarrow \varepsilon$
 $Y \rightarrow c$

$X \rightarrow Y$
 $X \rightarrow a$

	Nullable	FIRST	FOLLOW
X	Sim	a c	a c d
Y	Sim	c	a c d
Z	Não	a c d	

Tabela do Analisador Preditivo

4

Exemplo:

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

	nullable	FIRST	FOLLOW
X	yes	a c	a c d
Y	yes	c	a c d
Z	no	a c d	

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

FIRST(a)={a}
Nullable(Y), FOLLOW(X)={a,c,d}
Nullable(ϵ), FOLLOW(Y)={a,c,d}
FIRST(XYZ)={a,c,d}
FIRST(c)={c}
FIRST(d)={d}

As células com mais do que uma produção revelam uma gramática com características que não se adequam ao algoritmo (gramática não é LL(1))

Análise Descendente Recursiva

5

- Características de gramáticas que impedem a utilização deste tipo de parsing, mas que podem ser alvo de transformação:
 - Ambiguidade (já vimos como resolver)
 - Recursividade à esquerda
 - Produções de uma mesma categoria gramatical começam (à direita) com mesmo símbolo
- Como se pode resolver estes problemas?
 - O segundo, à custa de uma reescrita usando recursividade à direita
 - O terceiro, à custa de factorização à esquerda

Eliminação de Recursividade à Esquerda

6

$$\begin{aligned} P &\rightarrow P a \\ P &\rightarrow b \end{aligned}$$

- Reescrevem-se as produções com recursividade à direita:

$$\begin{aligned} P &\rightarrow b P' \\ P' &\rightarrow a P' \mid \varepsilon \end{aligned}$$

Outro exemplo:

$$\begin{array}{l} E \rightarrow E + T \\ E \rightarrow T \end{array} \longrightarrow \begin{array}{l} E \rightarrow T E' \\ E' \rightarrow + T E' \mid \varepsilon \end{array}$$

Fatorização

7

Duas produções para um mesmo não-terminal que começam com mesmo símbolo :

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

- Fatorizarão à esquerda: substituir a diferença por um símbolo não-terminal

$$S \rightarrow \text{if } E \text{ then } S \text{ } X$$
$$X \rightarrow \text{else } S$$
$$X \rightarrow \epsilon$$

Análise LL(1)

8

- As gramáticas livres de contexto que obedecem às características atrás descritas designam-se por Gramáticas LL(1):
 - L: left scan (leitura da esquerda para a direita - num passo)
 - L: leftmost derivation (derivação pela esquerda)
 - (1): 1-symbol lookahead (antecipação de 1 símbolo)

Análise LL(1) com Tabela de Parsing

10

- Considere-se a gramática

$$S \rightarrow E \$$$

$$E \rightarrow T E'$$

$$E' \rightarrow + T E'$$

$$E' \rightarrow - T E'$$

$$E' \rightarrow \varepsilon$$

$$T \rightarrow F T'$$

$$T' \rightarrow * F T'$$

$$T' \rightarrow / F T'$$

$$T' \rightarrow \varepsilon$$

$$F \rightarrow \text{id}$$

$$F \rightarrow \text{num}$$

$$F \rightarrow (E)$$

Análise LL(1) com Tabela de Parsing

11

□ Calcular nullable, FIRST e FOLLOW

$$S \rightarrow E \$$$

$$E \rightarrow T E'$$

$$E' \rightarrow + T E'$$

$$E' \rightarrow - T E'$$

$$E' \rightarrow \varepsilon$$

$$T \rightarrow F T'$$

$$T' \rightarrow * F T'$$

$$T' \rightarrow / F T'$$

$$T' \rightarrow \varepsilon$$

$$F \rightarrow \text{id}$$

$$F \rightarrow \text{num}$$

$$F \rightarrow (E)$$

	nullable	FIRST	FOLLOW
<i>S</i>		(id num	
<i>E</i>	no	(id num) \$
<i>E'</i>	yes	+ -) \$
<i>T</i>	no	(id num) + - \$
<i>T'</i>	yes	* /) + - \$
<i>F</i>	no	(id num) * / + - \$

Análise LL(1) com Tabela de Parsing

12

□ Construir Tabela de Parsing

	nullable	FIRST	FOLLOW
S		(id num	
E	no	(id num) \$
E'	yes	+ -) \$
T	no	(id num) + - \$
T'	yes	* /) + - \$
F	no	(id num) * / + - \$

 $S \rightarrow E \$$
 $E \rightarrow T E'$
 $E' \rightarrow + T E'$
 $E' \rightarrow - T E'$
 $E' \rightarrow \epsilon$
 $T \rightarrow F T'$
 $T' \rightarrow * F T'$
 $T' \rightarrow / F T'$
 $T' \rightarrow \epsilon$
 $F \rightarrow \text{id}$
 $F \rightarrow \text{num}$
 $F \rightarrow (E)$

(parcial)

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

Análise Sintática Ascendente

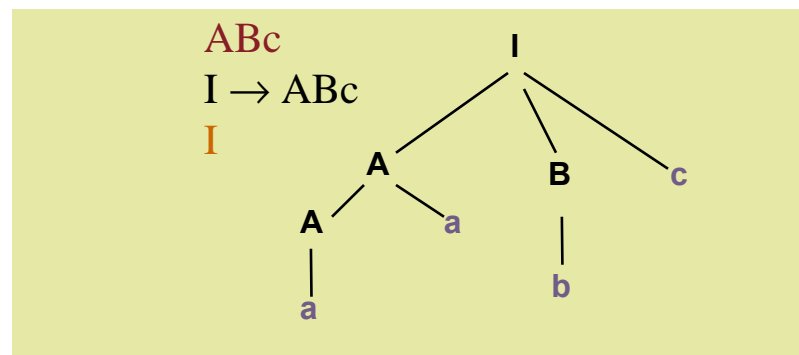
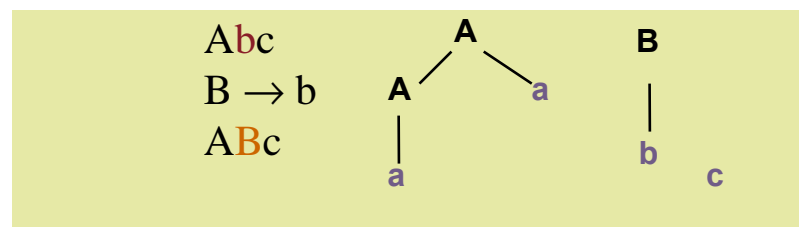
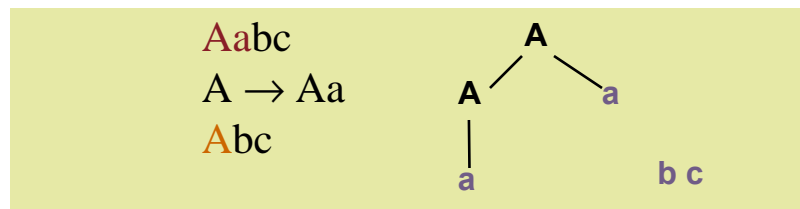
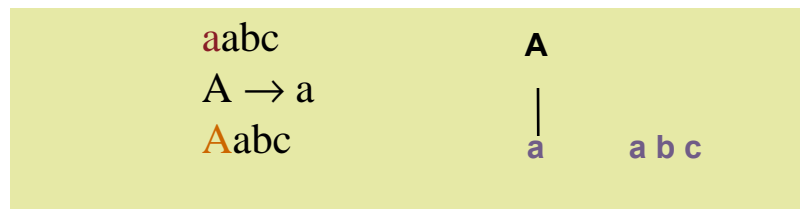
13

□ Gramática:

- $I \rightarrow ABc \mid B$
- $A \rightarrow Aa \mid a$
- $B \rightarrow b$

□ frase:

- aabc



Parsing LR:

L: scans left to right

R: rightmost-derivation

Parsing bottom-up

Análise Sintática Ascendente

14

- Recurso a uma Pilha
- Sucessão de operações deslocar/reduzir
- Objectivo: reduzir string a símbolo inicial da gramática

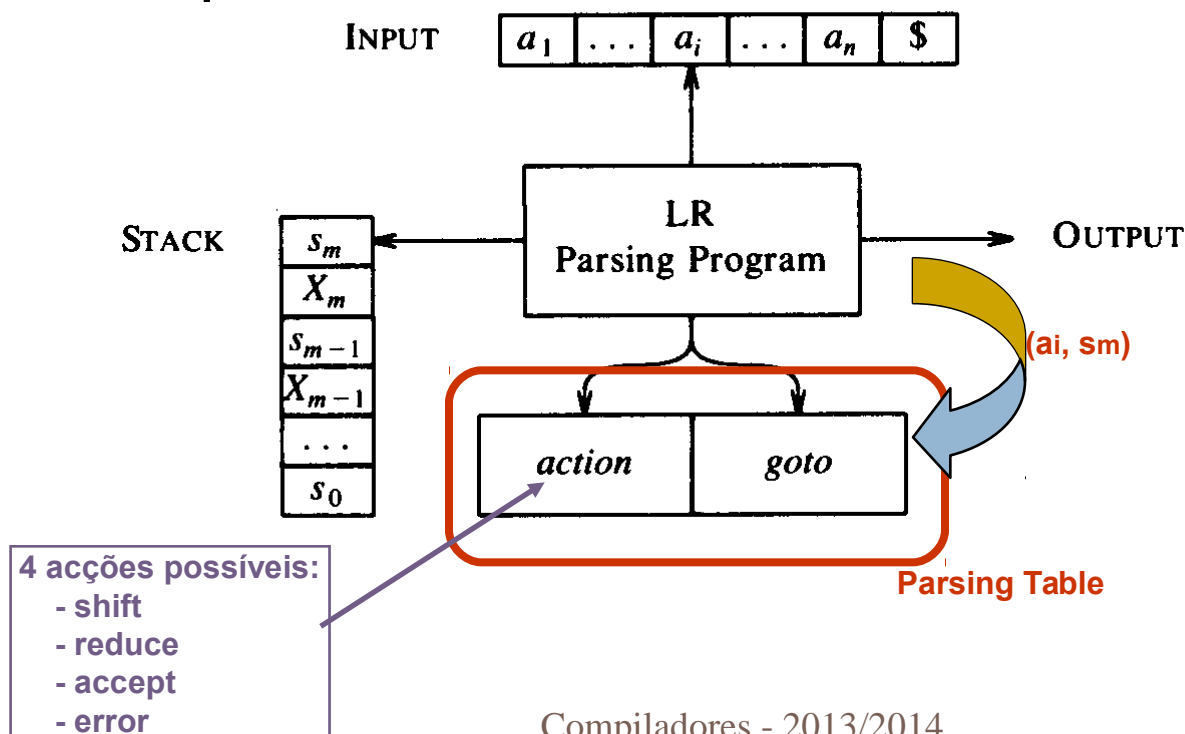
$$\begin{aligned} S &\rightarrow I \$ \\ I &\rightarrow ABc \mid B \\ A &\rightarrow Aa \mid a \\ B &\rightarrow b \end{aligned}$$

Pilha	Input	Acção
	a abc\$	desloca
a	abc\$	reduz ($A \rightarrow a$)
A	a bc\$	desloca
Aa	bc\$	reduz ($A \rightarrow Aa$)
A	b c\$	desloca
Ab	c\$	reduz ($B \rightarrow b$)
AB	c \$	desloca
ABc	\$	reduz ($I \rightarrow ABc$)
I	\$	aceita

Análise Sintática Ascendente

15

- Quando reduzir (reduce), quando deslocar (shift)?
 - DFA aplicado sobre a Pilha



Análise Sintática Ascendente

16

□ Exemplo:

⁰	$S' \rightarrow S \$$	⁴	$E \rightarrow \text{id}$	⁸	$L \rightarrow E$
¹	$S \rightarrow S ; S$	⁵	$E \rightarrow \text{num}$	⁹	$L \rightarrow L, E$
²	$S \rightarrow \text{id} := E$	⁶	$E \rightarrow E + E$		
³	$S \rightarrow \text{print}(L)$	⁷	$E \rightarrow (S, E)$		

- Parsing da frase:
 - $a := 7; b := c + (d := 5 + 6, d)$

Análise Sintática Ascendente

17

□ Tabela de parsind:

	id	num	print	;	,	+	:=	()	\$	S	E	L
1	s4		s7								g2		
2				s3						a			
3	s4		s7								g5		
4						s6							
5				r1	r1					r1			
6	s20	s10					s8					g11	
7							s9						
8	s4		s7								g12		
9	s20	s10					s8					g15	g14
10				r5	r5	r5		r5	r5				
11				r2	r2	s16				r2			
12				s3	s18								
13				r3	r3					r3			
14					s19				s13				
15					r8				r8				
16	s20	s10					s8					g17	
17				r6	r6	s16		r6	r6				
18	s20	s10					s8					g21	
19	s20	s10					s8					g23	
20				r4	r4	r4		r4	r4				
21								s22					
22				r7	r7	r7		r7	r7				
23					r9	s16		r9					

sn: shift; transitar para estado n
 gn: transitar para estado n
 rk: reduce pela regra k
 a: aceitar

Análise Sintática Ascendente

18

Stack	Input	Action
1	a := 7 ; b := c + (d := 5 + 6 , d) \$	shift
1 id ₄	:= 7 ; b := c + (d := 5 + 6 , d) \$	shift
1 id ₄ := 6	7 ; b := c + (d := 5 + 6 , d) \$	shift
1 id ₄ := 6 num ₁₀	; b := c + (d := 5 + 6 , d) \$	reduce E → num
1 id ₄ := 6 E ₁₁	; b := c + (d := 5 + 6 , d) \$	reduce S → id := E
1 S ₂	; b := c + (d := 5 + 6 , d) \$	shift
1 S ₂ ; 3	b := c + (d := 5 + 6 , d) \$	shift
1 S ₂ ; 3 id ₄	:= c + (d := 5 + 6 , d) \$	shift
1 S ₂ ; 3 id ₄ := 6	c + (d := 5 + 6 , d) \$	shift
1 S ₂ ; 3 id ₄ := 6 id ₂₀	+ (d := 5 + 6 , d) \$	reduce E → id
1 S ₂ ; 3 id ₄ := 6 E ₁₁	+ (d := 5 + 6 , d) \$	shift
1 S ₂ ; 3 id ₄ := 6 E ₁₁ + 16	(d := 5 + 6 , d) \$	shift
1 S ₂ ; 3 id ₄ := 6 E ₁₁ + 16 (8	d := 5 + 6 , d) \$	shift
1 S ₂ ; 3 id ₄ := 6 E ₁₁ + 16 (8 id ₄	:= 5 + 6 , d) \$	shift
1 S ₂ ; 3 id ₄ := 6 E ₁₁ + 16 (8 id ₄ := 6	5 + 6 , d) \$	shift
1 S ₂ ; 3 id ₄ := 6 E ₁₁ + 16 (8 id ₄ := 6 num ₁₀	+ 6 , d) \$	reduce E → num
1 S ₂ ; 3 id ₄ := 6 E ₁₁ + 16 (8 id ₄ := 6 E ₁₁	+ 6 , d) \$	shift
1 S ₂ ; 3 id ₄ := 6 E ₁₁ + 16 (8 id ₄ := 6 E ₁₁ + 16	6 , d) \$	shift
1 S ₂ ; 3 id ₄ := 6 E ₁₁ + 16 (8 id ₄ := 6 E ₁₁ + 16 num ₁₀	, d) \$	reduce E → num
1 S ₂ ; 3 id ₄ := 6 E ₁₁ + 16 (8 id ₄ := 6 E ₁₁ + 16 E ₁₇	, d) \$	reduce E → E + E
1 S ₂ ; 3 id ₄ := 6 E ₁₁ + 16 (8 id ₄ := 6 E ₁₁	, d) \$	reduce S → id := E
1 S ₂ ; 3 id ₄ := 6 E ₁₁ + 16 (8 S ₁₂	, d) \$	shift
1 S ₂ ; 3 id ₄ := 6 E ₁₁ + 16 (8 S ₁₂ , 18	d) \$	shift
1 S ₂ ; 3 id ₄ := 6 E ₁₁ + 16 (8 S ₁₂ , 18 id ₂₀) \$	reduce E → id
1 S ₂ ; 3 id ₄ := 6 E ₁₁ + 16 (8 S ₁₂ , 18 E ₂₁) \$	shift
1 S ₂ ; 3 id ₄ := 6 E ₁₁ + 16 (8 S ₁₂ , 18 E ₂₁) 22	\$	reduce E → (S , E)
1 S ₂ ; 3 id ₄ := 6 E ₁₁ + 16 E ₁₇	\$	reduce E → E + E
1 S ₂ ; 3 id ₄ := 6 E ₁₁	\$	reduce S → id := E
1 S ₂ ; 3 S ₅	\$	reduce S → S ; S
1 S ₂	\$	accept

O parser sabe o estado do DFA correspondente a cada símbolo colocado na pilha, conjugado com o estado anterior

Analizador Sintático LR(0)

19

□ Parser LR(0):

- trabalha olhando apenas para o topo da pilha

□ Gramática:

- | | | | |
|---|-------------------------|---|-----------------------|
| 0 | • $S' \rightarrow S \$$ | 3 | $L \rightarrow S$ |
| 1 | • $S \rightarrow (L)$ | 4 | $L \rightarrow L , S$ |
| 2 | • $S \rightarrow x$ | | |

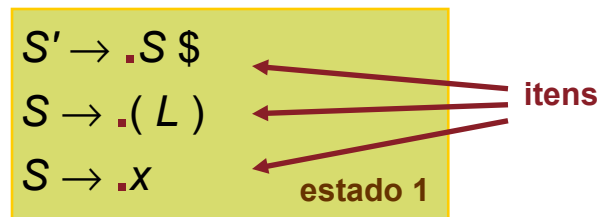
□ Início:

- pilha vazia
- frase completa à entrada, seguida de '\$'

Construção do Diagrama de Estados

20

□ Estado 1



$(A \rightarrow \alpha.\beta)$ indica que

- a sequência α está no topo da pilha
- na cabeça de leitura está uma cadeia derivável de β

$$S' \rightarrow S \$$$

$$L \rightarrow S$$

$$S \rightarrow (L)$$

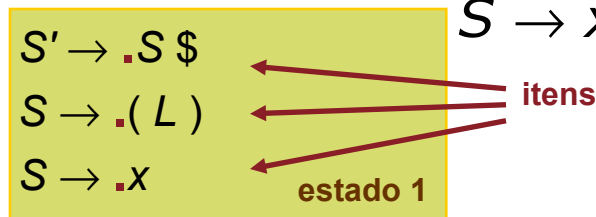
$$L \rightarrow L , S$$

$$S \rightarrow x$$

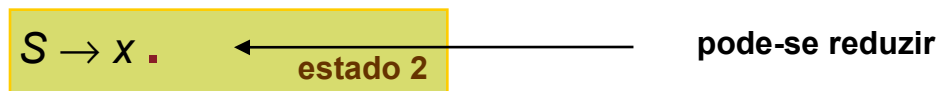
Construção do Diagrama de Estados

21

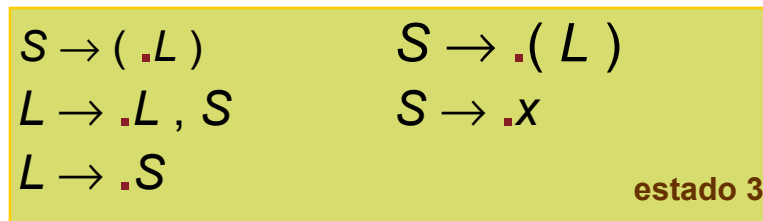
□ Estado 1

 $S' \rightarrow S \$$
 $L \rightarrow S$
 $S \rightarrow (L)$
 $L \rightarrow L , S$
 $S \rightarrow x$


Se no Estado 1 temos x para ler, desloca-se x (uma operação *shift*)...



Se no Estado 1 temos '(' para ler, desloca-se para a pilha...



Construção do Diagrama de Estados

22

□ Estado 1

$S' \rightarrow \cdot S \$$
 $S \rightarrow \cdot (L)$
 $S \rightarrow \cdot x$

estado 1

$S' \rightarrow S \$$ $L \rightarrow S$
 $S \rightarrow (L)$ $L \rightarrow L , S$
 $S \rightarrow x$

□ Se no Estado 1 S está no topo da pilha...

$S' \rightarrow S \cdot \$$

estado 4

□ ... salta-se para o Estado 4

ou seja,
 fez-se
 reduce
 para S

Construção do Diagrama de Estados

23

□ Closure(I):

- acrescentar novos itens ao conjunto I quando o ponto está à esquerda de um não-terminal

Closure(I) =

repeat

for any item $A \rightarrow \alpha.X\beta$ in I

for any production $X \rightarrow \gamma$

$I \leftarrow I \cup \{X \rightarrow .\gamma\}$

until I does not change.

return I

- Exemplo:

$S' \rightarrow .S \$$

$S \rightarrow .(L)$

$S \rightarrow .x$

estado 1

Construção do Diagrama de Estados

24

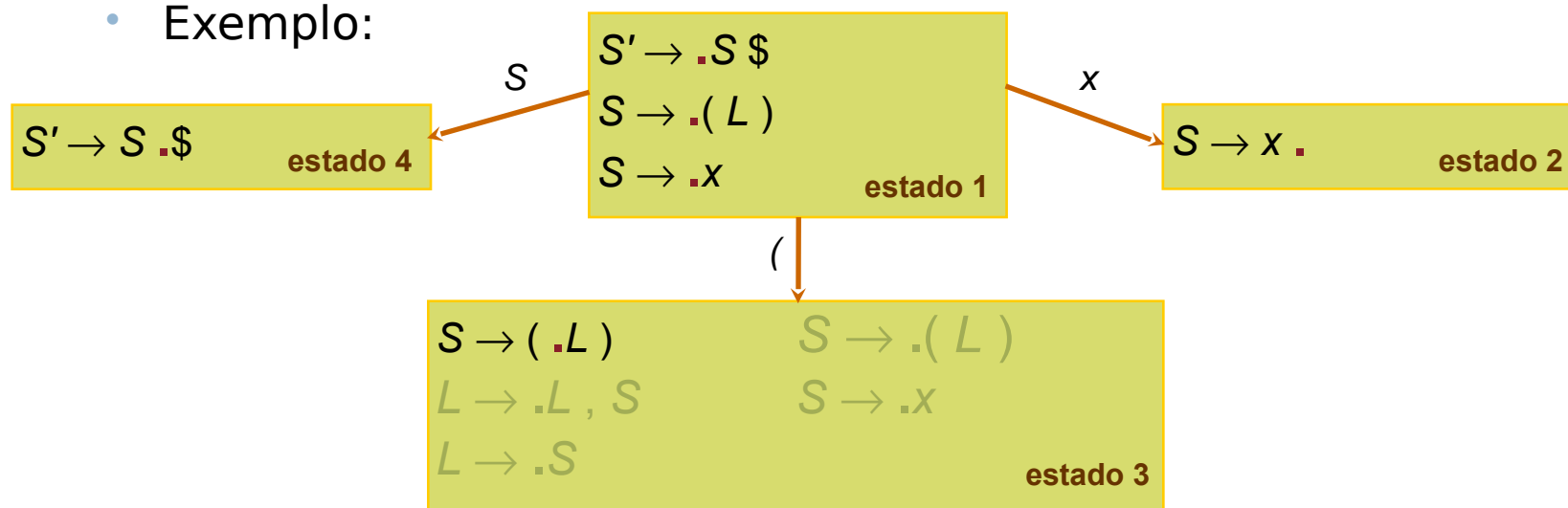
□ Goto(I, X):

- Devolve conjunto de itens que resulta de se mover o ponto para a direita de X em todos os itens

Goto(I, X) =

set J to the empty set
for any item $A \rightarrow \alpha.X\beta$ in I
 add $A \rightarrow \alpha X.\beta$ to J
return Closure(J)

- Exemplo:



Construção do Diagrama de Estados

25

- T : conjunto de estados já visitados
- E : conjunto de arestas já percorridas

Initialize T to $\{\text{Closure}(\{S' \rightarrow .S\})\}$

Initialize E to empty.

repeat

for each state I in T

for each item $A \rightarrow \alpha.X\beta$ in I

let J be $\text{Goto}(I, X)$

$T \leftarrow T \cup \{J\}$

$E \leftarrow E \cup \{I \xrightarrow{X} J\}$

until E and T did not change in this iteration

Reduções:

$R \leftarrow \{\}$

for each state I in T

for each item $A \rightarrow \alpha.$ in I

$R \leftarrow R \cup \{(I, A \rightarrow \alpha)\}$

Diagrama de Estados

26

⁰ $S' \rightarrow S \$$	³ $L \rightarrow S$
¹ $S \rightarrow (L)$	⁴ $L \rightarrow L , S$
² $S \rightarrow X$	

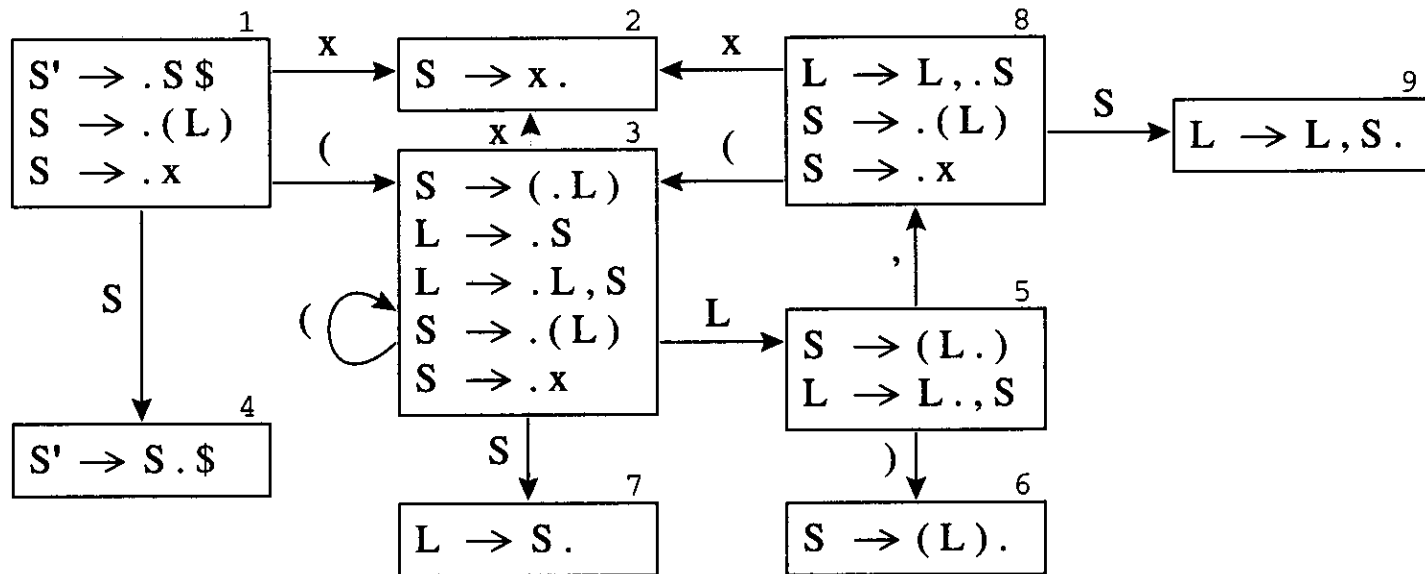


Tabela de Parsing

27

 $S' \rightarrow S \$$
 $L \rightarrow S$
 $S \rightarrow (L)$
 $L \rightarrow L , S$
 $S \rightarrow x$

	()	x	,	\$	S	L
1	s3		s2			g4	
2	r2	r2	r2	r2	r2		
3	s3		s2			g7	g5
4					a		
5		s6		s8			
6	r1	r1	r1	r1	r1		
7	r3	r3	r3	r3	r3		
8	s3		s2			g9	
9	r4	r4	r4	r4	r4		

Construção da Tabela do Analisador LR(0)

28

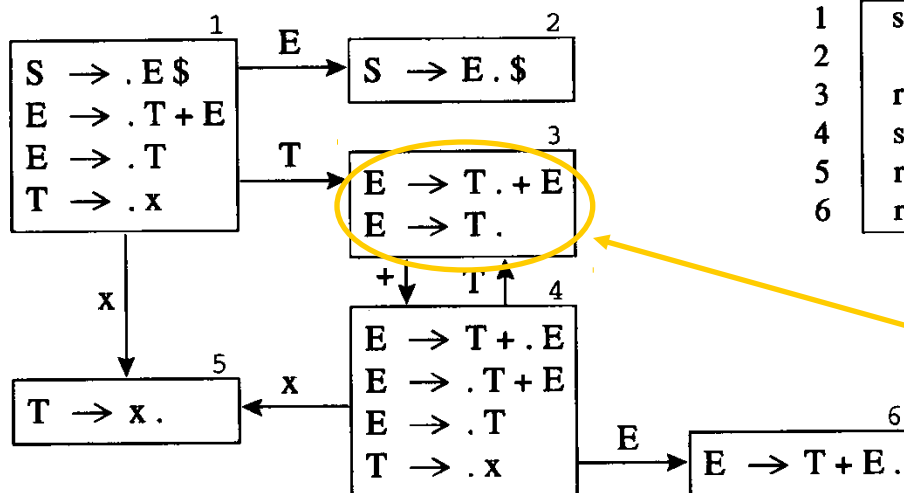
- Para cada aresta $I \rightarrow_X J$,
 - se X é terminal, coloca-se *shift* J na posição (I, X) da tabela
 - se X é não-terminal, coloca-se *goto* J na posição (I, X)
- Para cada estado I contendo um item $S' \rightarrow S.\$$, coloca-se um *accept* em $(I, \$)$
- Para cada estado contendo um item $A \rightarrow \gamma$. coloca-se uma *redução* n em (I, Y) para cada token Y ; n é número da produção $A \rightarrow \gamma$

Analizador Sintático LR(0)

29

- Um exemplo em que há um problema:

0	$S \rightarrow E \$$	2	$E \rightarrow T$
1	$E \rightarrow T + E$	3	$T \rightarrow x$



	x	+	\$	E	T
1	s5			g2	g3
2			a		
3	r2	s4,r2	r2		
4	s5			g6	g3
5	r3	r3	r3		
6	r1	r1	r1		

“famoso” erro de shift/reduce!!
gramática não é LR(0)
necessário algoritmo adequado

Resumo

30

- Algoritmo de análise descendente
- Algoritmo de análise ascendente:
 - Construção do Diagrama de Estados
 - Construção da Tabela de Parsing
 - Como se processa a análise
- Próxima aula:
 - Outras variantes de análise ascendente

Exercício

31

- Construa o analisador LR(0) para a seguinte gramática:
- $S \rightarrow E \$$
- $E \rightarrow id$
- $E \rightarrow id (E)$
- $E \rightarrow E + id$

