

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 ascendente

- *Bottom-up*, ascendente ou redutiva
 - Analisadores de precedência de operadores
 - **Analisadores LR**
 - SLR: *Simple LR*
 - LR Canônico
 - *Look Ahead LR*: LALR

2

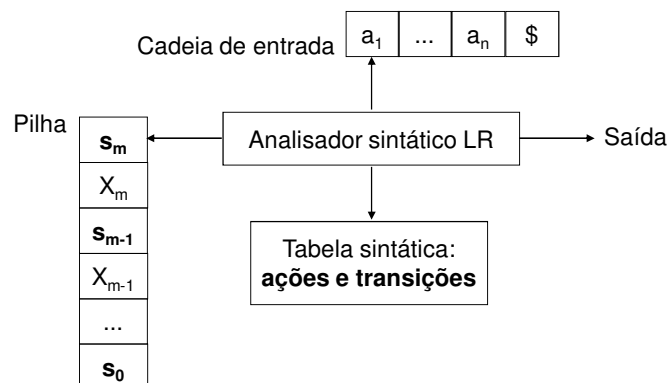
Analísadores LR

- **LR(k)**: *Left to right, Rightmost derivation, with k lookahead symbols*
 - LR(1)
 - Ampla classe de gramáticas
- Método mais genérico e poderoso de análise
 - Podem reconhecer praticamente todas as construções possíveis em linguagens de programação
 - Implementação eficiente
 - Desvantagem: manipulação complexa da tabela sintática
- Yacc segue esta estratégia

3

Analísadores LR

- Esquema de um analisador LR
 - X_i são símbolos gramaticais
 - s_i são estados que resumam a informação contida abaixo na pilha



4

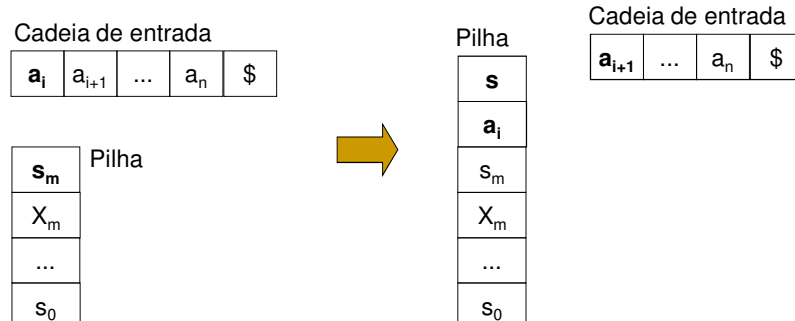
Analísadores LR

- A combinação do estado do topo da pilha e o primeiro símbolo da cadeia de entrada é utilizada para indexar a tabela sintática e determinar o que se faz (empilha ou reduz)
- Comportamento do analisador
 - Para o estado do topo da pilha s_m e o símbolo da entrada a_i , consulta-se ação[s_m, a_i] na tabela
 - Quatro possíveis valores
 - Empilhamento de um estado
 - Redução por uma regra gramatical
 - Aceitação da cadeia de entrada
 - Erro
 - Pela transição (ou “desvio”) indicada na tabela, toma-se um estado e um símbolo gramatical e se produz um novo estado como saída

5

Analísadores LR

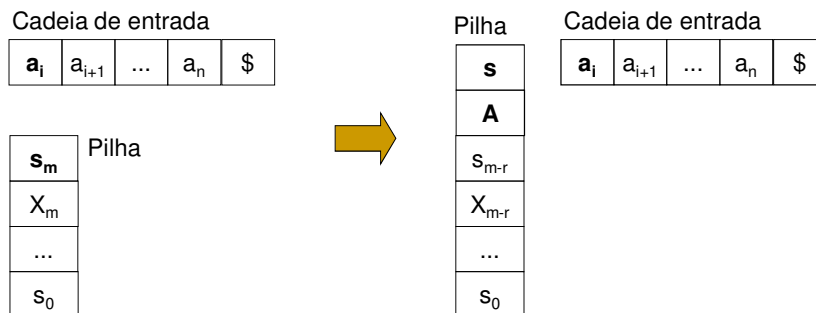
- Para o estado do topo da pilha s_m e o símbolo da entrada a_i , consulta-se ação[s_m, a_i]
 - Se ação[s_m, a_i] = “empilhar s ”, então empilham-se a_i e s



6

Analísadores LR

- Se $\text{ação}[s_m, a_i] = \text{"reduzir } A \rightarrow \beta \text{"}$, então
 - Sendo r o tamanho de β , desempilham-se $2r$ elementos (r estados e r símbolos gramaticais): o topo torna-se s_{m-r}
 - Empilha-se A (lado esquerdo da regra utilizada)
 - Empilha-se o estado s , indicado por transição $[s_{m-r}, A]$



7

Analísadores LR

- Se $\text{ação}[s_m, a_i] = \text{"aceitar"}$, então a análise sintática tem sucesso e é encerrada
- Se $\text{ação}[s_m, a_i] = \text{"erro"}$, então o analisador descobriu um erro e deve tratá-lo

8

Analísadores LR

■ Em outras palavras

- Sempre deve haver um estado no topo da pilha
 - O estado diz tudo sobre a análise sintática
 - Símbolos gramaticais são dispensáveis na pilha, na realidade
- Quando se empilha um símbolo gramatical, deve-se empilhar um estado em seguida
- Ao se reduzir, (i) removem-se da pilha os símbolos gramaticais que correspondem ao lado direito do *handle* e os estados correspondentes e (ii) empilha-se o lado esquerdo do *handle* (i.e., um símbolo gramatical), (iii) gerando a necessidade de se empilhar mais um estado

9

Analísadores LR

■ Algoritmo de análise LR

empilha-se o estado inicial s_0 ;
concatena-se o símbolo delimitador \$ no final da cadeia de entrada
fazer ip apontar para o primeiro símbolo da cadeia
repetir
 seja s_n o estado no topo da pilha e a o símbolo apontado por ip
 se (ação[s_n, a] = "empilhar s_{n+1} ") então
 empilhar a ;
 empilhar s_{n+1} ;
 avancar ip ;
 senão se (ação[s_n, a] = "reduzir $A \rightarrow \beta$ ") então
 desempilhar $2 * |\beta|$ elementos;
 empilhar A ;
 empilhar o estado indicado por transição[$s_{n-2 * |\beta|}, A$];
 senão se (ação[s_n, a] = "aceitar") então SUCESSO
 senão ERRO;
fim-repetir;

10

Analísadores LR

- Reconhecer a cadeia id^*id+id

$$(1) \langle E \rangle ::= \langle E \rangle + \langle T \rangle$$

(2) $\langle E \rangle ::= \langle T \rangle$

(3) $\langle T \rangle ::= \langle T \rangle^* \langle F \rangle$

(4) $\langle T \rangle ::= \langle F \rangle$

$$(5) \langle F \rangle ::= (\langle E \rangle)$$

(6) $\langle F \rangle ::= \text{id}$

Na tabela, tem-se que:

- *sí* indica “empilhar *i*”

- *ri* indica “reduzir por regra *i*”

Tabela sintática LR

Estados	Ações						Transições		
	id	+	*	()	\$	E	T	F
0	s5			s4			1	2	3
1		s6				OK			
2		r2	s7		r2	r2			
3		r4	r4		r4	r4			
4	s5			s4			8	2	3
5		r6	r6		r6	r6			
6	s5			s4				9	3
7	s5			s4					10
8		s6			s11				
9		r1	s7		r1	r1			
10		r3	r3		r3	r3			
11		r5	r5		r5	r5			

Analísadores LR

- Reconhecer a cadeia id^*id+id

Pilha	Cadeia	Regra
0	id*id+id\$	

Analísadores LR

- Reconhecer a cadeia $id*id+id$

Pilha	Cadeia	Regra
0	$id*id+id\$$	s5
0id5	$*id+id\$$	r6
0F3	$*id+id\$$	r4
0T2	$*id+id\$$	s7
0T2*7	$id+id\$$	s5
0T2*7id5	$+id\$$	r6
0T2*7F10	$+id\$$	r3
0T2	$+id\$$	r2
0E1	$+id\$$	s6
0E1+6	$id\$$	s5
0E1+6id5	$\$$	r6
0E1+6F3	$\$$	r4
0E1+6T9	$\$$	r1
0E1	$\$$	OK

13

Analísadores LR

- Exercício: reconhecer a cadeia (id)

Pilha	Cadeia	Regra
0	(id)\$	

14

Analísadores LR

- Exercício: reconhecer a cadeia (id)

Pilha	Cadeia	Regra
0	(id)\$	s4
0(4	id)\$	s5
0(4id5)\$	r6
0(4F3)\$	r4
0(4T2)\$	r2
0(4E8)\$	s11
0(4E8) <u>11</u>	\$	r5
0F3	\$	r4
0T2	\$	r2
0E1	\$	OK

15

Analísadores LR

- Notem que
 - Transições são para símbolos não terminais
 - As transições para os símbolos terminais estão implícitas nas ações
 - O estado no topo da pilha oferece toda a informação sobre o *handle* encontrado
 - Não é preciso percorrer a pilha para encontrar o *handle*
 - Eficiência

16

Analísadores LR

- **Três técnicas** para construir tabelas sintáticas para gramáticas LR
 - *Simple* LR (SLR)
 - Mais fácil de implementar, mas o menos poderoso
 - LR canônico
 - Mais complexo, mas mais poderoso
 - *Look Ahead* LR (LALR)
 - Complexidade e poder intermediários
- Tabelas possivelmente distintas para cada técnica, determinando o poder do analisador

17

Analísadores LR

- **Gramáticas LR**
 - Uma gramática é LR se é possível construir uma tabela sintática LR para ela
 - Não pode haver ambigüidade
- **LL(k) vs. LR(k)**
 - LL(k): decide-se por uma produção olhando-se apenas os k primeiros símbolos da cadeia de entrada
 - LR(k): reconhece-se o lado direito de uma produção tendo visto tudo que foi derivado a partir desse lado direito (na pilha) mais o esquadramento antecipado de k símbolos (da cadeia de entrada)
 - Mais poderoso do que LL(k): pode descrever mais linguagens

18

Análise SLR

- A análise sintática por meio de uma tabela SLR é chamada **análise sintática SLR**
- Uma gramática é SLR se for possível construir uma tabela SLR para ela

19

Análise SLR

- A construção da tabela SLR se baseia no **conjunto canônico de itens $LR(0)$**
 - $LR(0)$: não se olha nenhum símbolo a frente
- Um item para uma gramática G é uma regra de produção com alguma indicação do que já foi derivado/consumido na regra durante a análise sintática
 - Exemplo: $A \rightarrow XYZ$
 - $A \rightarrow .XYZ$
 - $A \rightarrow X.YZ$
 - $A \rightarrow XY.Z$
 - $A \rightarrow XYZ.$
 - Regras do tipo $A \rightarrow \lambda$ geram somente um item $A \rightarrow .$

20

Análise SLR

■ Construção do conjunto canônico de itens

□ Duas operações

1. Acrescentar à gramática a produção $S' \rightarrow S$ (em que S é o símbolo inicial da gramática)
 - Permite a identificação do fim da análise, mais especificamente, $S' \rightarrow S$.
2. Computar as funções *fechamento* e *transição* para a nova gramática

21

Análise SLR

■ Função *fechamento*

- Seja I um conjunto de itens LR(0)
 1. Todo item em I pertence ao fechamento(I)
 2. Se $A \rightarrow \alpha.X\beta$ está em fechamento(I) e $X \rightarrow \gamma$ é uma produção, então adiciona-se $X \rightarrow .\gamma$ ao conjunto

□ Em outras palavras

- Inicializa-se o conjunto I com as regras iniciais da gramática, colocando-se o indicador (.) no início de cada regra
- Para cada regra no conjunto, adicionam-se as regras dos não terminais que aparecem precedidos pelo indicador (.)

22

Análise SLR

■ Exemplo

$S' \rightarrow S$

$S \rightarrow a \mid [L]$

$L \rightarrow L;S \mid S$

$I = \{S \rightarrow [\cdot L]\}$

fechamento(I) =

23

Análise SLR

■ Exemplo

$S' \rightarrow S$

$S \rightarrow a \mid [L]$

$L \rightarrow L;S \mid S$

$I = \{S \rightarrow [\cdot L]\}$

fechamento(I) = $\{S \rightarrow [\cdot L], L \rightarrow \cdot L;S, L \rightarrow \cdot S, S \rightarrow \cdot a, S \rightarrow \cdot [L]\}$

24

Análise SLR

■ Função *transição*

- *transição*(I,X): consiste avançar o indicador (.) através do símbolo gramatical X das produções correspondentes em I e calcular a função *fechamento* para o novo conjunto

■ Exemplo

$I = \{S \rightarrow [L., L \rightarrow L.; S\}$

$\text{transição}(I,;) =$

25

Análise SLR

■ Função *transição*

- *transição*(I,X): consiste avançar o indicador (.) através do símbolo gramatical X das produções correspondentes em I e calcular a função *fechamento* para o novo conjunto

■ Exemplo

$I = \{S \rightarrow [L., L \rightarrow L.; S\}$

$\text{transição}(I,;) = \{L \rightarrow L.; S, S \rightarrow .a, S \rightarrow .[L]\}$

26

Análise SLR

- Algoritmo para obter o conjunto canônico de itens LR(0)

$C := \{I_0 = \text{fechamento}(\{S' \rightarrow S\})\}$

repita

para cada conjunto I em C e X símbolo de G , tal que
transição(I, X) $\neq \lambda$

adicione transição(I, X) a C

até que todos os conjuntos tenham sido adicionados a C

27

Análise SLR

- Exemplo

0) $S' \rightarrow S$

1) $S \rightarrow a$

2) $S \rightarrow [L]$

3) $L \rightarrow L;S$

4) $L \rightarrow S$

28

Análise SLR

Exemplo

Conjunto de itens

0) $S' \rightarrow S$	$I_0 = \{S' \rightarrow .S, S \rightarrow .a, S \rightarrow .[L]\}$
1) $S \rightarrow a$	$\text{transição}(I_0, S) = \{S' \rightarrow S.\} = I_1$
2) $S \rightarrow [L]$	$\text{transição}(I_0, a) = \{S \rightarrow a.\} = I_2$
3) $L \rightarrow L;S$	$\text{transição}(I_0, [) = \{S \rightarrow [.L], L \rightarrow .L;S, L \rightarrow .S, S \rightarrow .a, S \rightarrow .[L]\} = I_3$
4) $L \rightarrow S$	$\text{transição}(I_3, L) = \{S \rightarrow [L.], L \rightarrow L.;S\} = I_4$
	$\text{transição}(I_3, S) = \{L \rightarrow S.\} = I_5$
	$\text{transição}(I_3, a) = \{S \rightarrow a.\} = I_2$
	$\text{transição}(I_3, [) = \{S \rightarrow [.L], L \rightarrow .L;S, L \rightarrow .S, S \rightarrow .a, S \rightarrow .[L]\} = I_3$
	$\text{transição}(I_4,) = \{S \rightarrow [L.]\} = I_6$
	$\text{transição}(I_4, ;) = \{L \rightarrow L.;S, S \rightarrow .a, S \rightarrow .[L]\} = I_7$
	$\text{transição}(I_7, S) = \{L \rightarrow L;S.\} = I_8$
	$\text{transição}(I_7, a) = \{S \rightarrow a.\} = I_2$
	$\text{transição}(I_7, [) = \{S \rightarrow [.L], L \rightarrow .L;S, L \rightarrow .S, S \rightarrow .a, S \rightarrow .[L]\} = I_3$

29

Análise SLR

Construção da tabela sintática

- Seja $C = \{I_0, I_1, \dots, I_n\}$, os estados são $0 \dots n$, com 0 sendo o estado inicial
- A linha i da tabela é construída pelo conjunto I_i
 - Ações na tabela
 - Se $\text{transição}(I_i, a) = I_j$, então ação $[i, a] = sj$
 - Com exceção da regra $S' \rightarrow S$ adicionada, para todas as outras regras, se $A \rightarrow \alpha$ está em I_i , então, para todo a em $\text{seguidor}(A)$, faça ação $[i, a] = rn$, em que n é o número da produção $A \rightarrow \alpha$
 - Se $S' \rightarrow S$ está em I_i , então faça ação $[i, \$] = OK$
 - Transições na tabela
 - Se $\text{transição}(I_i, A) = I_j$, então $\text{transição}(i, A) = j$

- Entradas não definidas indicam erros
- Ações conflitantes indicam que a gramática não é SLR

30

Análise SLR

■ Construção da tabela sintática

- 0) $S' \rightarrow S$
- 1) $S \rightarrow a$
- 2) $S \rightarrow [L]$
- 3) $L \rightarrow L;S$
- 4) $L \rightarrow S$

$S(S') = \{\$ \}$
 $S(S) = S(S') \cup S(L) = \{\$, [, ; \}$
 $S(L) = \{[, ; \}$

OBS: agora, para indicar fim de cadeia, utiliza-se o \$
Antes, estávamos sobrecarregando o símbolo de cadeia vazia λ

31

Análise SLR

■ Construção da tabela sintática

- 0) $S' \rightarrow S$
- 1) $S \rightarrow a$
- 2) $S \rightarrow [L]$
- 3) $L \rightarrow L;S$
- 4) $L \rightarrow S$

$S(S') = \{\$ \}$
 $S(S) = S(S') \cup S(L) = \{\$, [, ; \}$
 $S(L) = \{[, ; \}$

Tabela sintática SLR

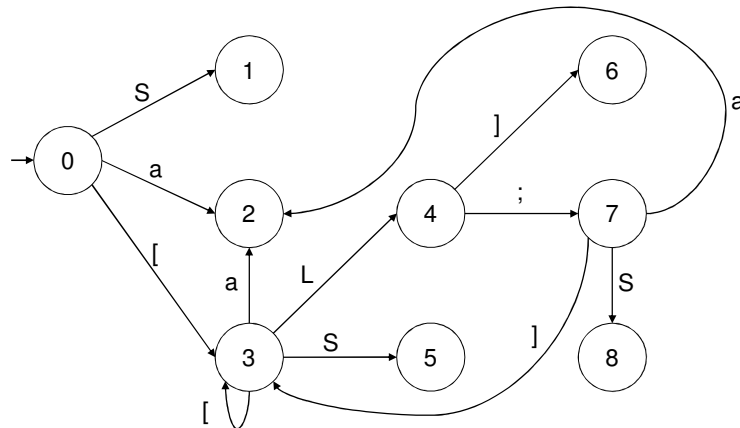
Estados	Ações					Transições	
	a	[]	;	\$	S	L
0	s2	s3				1	
1					OK		
2			r1	r1	r1		
3	s2	s3				5	4
4			s6	s7			
5			r4	r4			
6			r2	r2	r2		
7	s2	s3				8	
8			r3	r3			

OBS: agora, para indicar fim de cadeia, utiliza-se o \$
Antes, estávamos sobrecarregando o símbolo de cadeia vazia λ

32

Análise SLR

■ Autômato correspondente



33

Análise SLR

- Como o autômato não é usado para reconhecer cadeias, mas para acompanhar o estado da análise sintática, não há estados finais
- O analisador aceitará uma cadeia quando ocorrer uma redução pela regra adicionada $S' \rightarrow S$

34

Análise SLR

- Exercício: construir a tabela para a gramática

$S \rightarrow \text{if } E \text{ then } C \mid C$

$E \rightarrow a$

$C \rightarrow b$

35

Análise SLR

- Passo 1: adicionar a regra $S' \rightarrow S$

0) $S' \rightarrow S$

1) $S \rightarrow \text{if } E \text{ then } C$

2) $S \rightarrow C$

3) $E \rightarrow a$

4) $C \rightarrow b$

36

Análise SLR

■ Passo 2: construir o conjunto de itens

- 0) $S' \rightarrow S$
- 1) $S \rightarrow \text{if } E \text{ then } C$
- 2) $S \rightarrow C$
- 3) $E \rightarrow a$
- 4) $C \rightarrow b$

37

Análise SLR

■ Passo 2: construir o conjunto de itens

- 0) $S' \rightarrow S$
- 1) $S \rightarrow \text{if } E \text{ then } C$
- 2) $S \rightarrow C$
- 3) $E \rightarrow a$
- 4) $C \rightarrow b$

Conjunto de itens

$I_0 = \{S' \rightarrow .S, S \rightarrow .\text{if } E \text{ then } C, S \rightarrow .C, C \rightarrow .b\}$
 $t(I_0, S) = \{S' \rightarrow S.\} = I_1$
 $t(I_0, \text{if}) = \{S \rightarrow \text{if } .E \text{ then } C, E \rightarrow .a\} = I_2$
 $t(I_0, C) = \{S \rightarrow C.\} = I_3$
 $t(I_0, b) = \{C \rightarrow b.\} = I_4$
 $t(I_2, E) = \{S \rightarrow \text{if } E .\text{then } C\} = I_5$
 $t(I_2, a) = \{E \rightarrow a.\} = I_6$
 $t(I_5, \text{then}) = \{S \rightarrow \text{if } E \text{ then } .C, C \rightarrow .b\} = I_7$
 $t(I_7, C) = \{S \rightarrow \text{if } E \text{ then } C.\} = I_8$
 $t(I_7, b) = \{C \rightarrow b.\} = I_4$

38

Análise SLR

■ Passo 3: construir a tabela sintática

- 0) $S' \rightarrow S$
- 1) $S \rightarrow \text{if } E \text{ then } C$
- 2) $S \rightarrow C$
- 3) $E \rightarrow a$
- 4) $C \rightarrow b$

$S(S') = \{\$ \}$
 $S(S) = S(S') = \{\$ \}$
 $S(E) = \{\text{then}\}$
 $S(C) = S(S) = \{\$ \}$

39

Análise SLR

■ Passo 3: construir a tabela sintática

- 0) $S' \rightarrow S$
- 1) $S \rightarrow \text{if } E \text{ then } C$
- 2) $S \rightarrow C$
- 3) $E \rightarrow a$
- 4) $C \rightarrow b$

$S(S') = \{\$ \}$
 $S(S) = S(S') = \{\$ \}$
 $S(E) = \{\text{then}\}$
 $S(C) = S(S) = \{\$ \}$

Tabela sintática SLR

Estados	Ações					Transições		
	if	then	a	b	\$	S	E	C
0	s2			s4		1		3
1					OK			
2			s6				5	
3					r2			
4					r4			
5		s7						
6		r3						
7				s4				8
8					r1			

40

Análise SLR

- Exercício: reconhecer a cadeia *if a then b*

Pilha	Cadeia	Regra
0	if a then b \$	

41

Análise SLR

- Exercício: reconhecer a cadeia *if a then b*

Pilha	Cadeia	Regra
0	if a then b \$	s2
0 if 2	a then b \$	s6
0 if 2 a 6	then b \$	r3
0 if 2 E 5	then b \$	s7
0 if 2 E 5 then 7	b \$	s4
0 if 2 E 5 then 7 b 4	\$	r4
0 if 2 E 5 then 7 C 8	\$	r1
0 S 1	\$	OK

42