

INF 1022 – P2 de Anal. Sintáticos e Léxicos – 2021.1
Prof Edward Hermann Haeusler

Resolva as questões abaixo.

1. Sejam as gramáticas abaixo:

1. G1

$$S \rightarrow aSa \mid bSb \mid \epsilon$$

2. G2

$$S \rightarrow aSa \mid bSb \mid c$$

Diga se a afirmação a seguir é verdadeira ou falsa, justificando.

A linguagem gerada por G1 não pode ser analisada por um parser LR(1). No entanto, a linguagem gerada por G2 possui parser LR(1).

→ A afirmação é verdadeira.

→ Dever-se notar que o parser LR(1) só analisa linguagens livres de contexto que sejam determinísticas, de modo que ele produza uma única análise comela sem guesswork ou backtracking, em tempo linear. Portanto, basta analisar o determinismo das linguagens geradas pelas gramáticas.

→ A linguagem gerada por G1 é da forma $\{ww^R : w \in \{a,b\}^*\}$, dos palíndromos pares sobre a e b. Ela não é determinística. Eis a demonstração:

"Seja $v_k = ab^{2k}$ uma família de palavras, que não palíndromos, sobre a e b. Seja $w_0 = \epsilon$ e $w_{k+1} = v_k v_k v_k$, que formam palíndromos que tem palíndromos como prefixo. A linguagem $\{v_k : k \in \mathbb{N}\}$ não pode ser aceita por PDA".

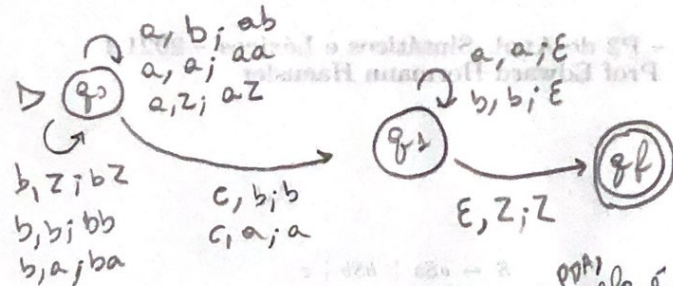
Portanto, se a linguagem não é aceita por PDA, ela não é determinística, e não possui parser LR(1). ■

(assim, necessariamente encontramos ao menos um conflito na tabela LR(1)).

→ A linguagem descrita por G2 é da forma $\{wcw^R : w \in \{a,b\}^*\}$ dos palíndromos de tamanho ímpar sobre a e b, com c sendo o símbolo do centro da palavra. Temos que a linguagem em questão é determinística. Pensando intuitivamente, para cada "a" ou "b" lido antes de "c", empilha-se tal símbolo na pilha, que será a base de leitura, o símbolo do PDA. O processo repete até atingir a base de "c", que funciona como um gatilho para começar a retirar os símbolos da pilha. O último símbolo empilhado será o primeiro a ser desempilhado, de modo que ao final do processo teremos gerado o mesmo da palavra lida antes de "c".

↳ PRÓXIMA PÁGINA

Para provar a hipótese inicial de determinismo, deve-se encontrar um PDA que aceite a LEC em questão. Eis o PDA:



→ Como a linguagem gerada por G2 aceita por PDA, ela é determinística, e portanto possível por LR(1). ■

2. Seja G a gramática a seguir.

$$S \rightarrow aS \mid T$$

$$T \rightarrow aTb \mid \epsilon$$

Responda os seguintes itens:

1. Explique como um analisador ascendente tipo LR(0) e SLR(1) lida com as produções vazias. Use G como exemplo e mostre como fica a tabela LR(0) e a SLR(1). Lembre-se que é necessário usar a forma aumentada da gramática nos parsers vistos no curso.
2. G é LR(0)?
3. G é SLR(1)?
4. G é LL(1)? **NÃO** faça a tabela LL(1). Dica: veja por FIRST e FOLLOW.

1. Utilizando G como exemplo. Aumentando a gramática:

$$(0) S' \rightarrow S \$$$

$$(1) S \rightarrow aS$$

$$(2) S \rightarrow T$$

$$(3) T \rightarrow aTb$$

$$(4) T \rightarrow \epsilon$$

$$\text{Follow}(S') = \text{Follow}(S)$$

$$= \{ \$ \}$$

$$\text{Follow}(T) = \{ \$, b \}$$

Tabela LR(0):

Estado	Ação			GoTo	
	\$	a	b	S	T
0	r4	s2 r4	r4	1	3
1	acc				
2	r4	s2 r4	r4	4	5
3	r2	r2	r2		
4	r1	r1	r1		
5	r2	r2	s6 r2		
6	r3	r3	r3		

→ Há conflitos do tipo shift-reduce na tabela

LR(0) devido à existência de ϵ , enquanto que na tabela SLR(1) esses conflitos são resolvidos.

→ Tal exemplo condiz com a observação de que gramáticas com ϵ e ϵ não são LR(0).

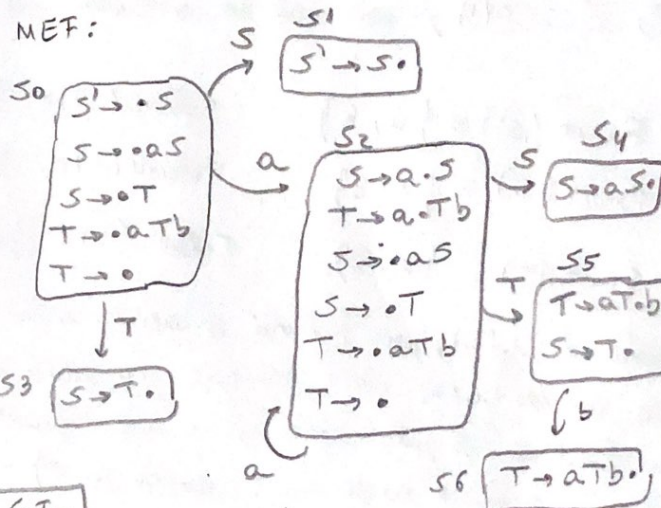


Tabela SLR(1):

Estado	Ação			GoTo	
	\$	a	b	S	T
0	r4	s2 r4	r4		
1	acc				
2	r4	s2 r4	r4	4	5
3	r2				
4	r1				
5	r2		s6 r2		
6	r3		r3		

→ PRÓXIMA PÁGINA

Explicação: Devido à regra ϵ , a análise atinge pontos onde há conflitos de shift-reduce, nos quais o computador se aplica um shift e realiza a leitura do símbolo, ou reduz a regra ϵ . No caso, o parser $LK(0)$ não sabe qual ação tomar, já que ele não utiliza nenhum do Ken de lookahead para escolher a melhor ação. Por outro lado, ao posicionar $SLK(1)$ é permitida olhar 1 do Ken de lookahead para decidir pela ação de shift ou reduce. No caso, o $SLK(1)$ irá reduzir uma regra da forma $A \rightarrow W$ se quando ele ler o próximo do Ken da input, qual do Ken aparecer depois de A em alguma derivação. Caso contrário, ele opta pelo shift.

2. G não é $LK(0)$, pois há conflitos de shift-reduce na tabela $LK(0)$

3. G é $SLK(1)$, pois não há conflitos na tabela $SLK(1)$.

$$\begin{array}{ll} 4. \text{First}(S') = \{a, \epsilon\} & \text{Follow}(S') = \{\$, \epsilon\} \\ \text{First}(S) = \{a, \epsilon\} & \text{Follow}(S) = \{\$, \epsilon\} \\ \text{First}(T) = \{a, \epsilon\} & \text{Follow}(T) = \{\$, b\} \end{array}$$

G não é $LL(1)$, pois haverá mais de uma regra em uma das células da tabela $LL(1)$. Isso acontece por causa do símbolo " a ", que é tanto um first de S pela regra $S \rightarrow aS$ quanto pela regra $S \rightarrow T$ (já que " a " é first tanto de S quanto de T):

3. Mostre que a gramática abaixo:

$$\begin{aligned} S' &\rightarrow S\$ \\ S &\rightarrow \text{if } E \text{ then } A \text{ else } S \mid \text{if } E \text{ then } S \mid \text{stat} \\ A &\rightarrow \text{if } E \text{ then } A \text{ else } A \mid \text{stat} \\ E &\rightarrow \text{exp} \end{aligned}$$

Obs: $\Sigma = \{\text{if, exp, stat, then, else, \$}\}$.

1. Não é ambígua.
2. É SLR(1).

$\text{First}(S') =$
 $\text{First}(S) =$
 $\text{First}(A) =$
 $\{\text{if, stat}\}$
 $\text{First}(E) =$
 $\{\text{exp}\}$
 $\text{Follow}(S') =$
 $\text{Follow}(S) =$
 $\{\$, \}$
 $\text{Follow}(A) =$
 $\{\text{else}\}$
 $\text{Follow}(E) =$
 $\{\text{then}\}$

1. Sebe-se que uma gramática ambígua apresentará inicialmente conflitos shift/reduce ou reduce/reduce para os métodos de análise LR, incluindo o SLR(1). Logo, pela contrapositiva, temos que: se não existirem nenhum conflito na tabela SLR(1), a gramática não será ambígua. Verificação:

Estados:

$(0) S' \rightarrow S \$$
 $(1) S \rightarrow \text{if } E \text{ then } A \text{ else } S$
 $(2) S \rightarrow \text{if } E \text{ then } S$
 $(3) S \rightarrow \text{stat}$
 $(4) A \rightarrow \text{if } E \text{ then } A \text{ else } A$
 $(5) A \rightarrow \text{stat}$
 $(6) E \rightarrow \text{exp}$

$S_0: S' \rightarrow \cdot S$
 $S \rightarrow \cdot \text{if } E \text{ then } A \text{ else } S$
 $S \rightarrow \cdot \text{if } E \text{ then } S$
 $S \rightarrow \cdot \text{stat}$

$S_1: S' \rightarrow S \cdot$

$S_2: S \rightarrow \text{if} \cdot E \text{ then } A \text{ else } S$
 $S \rightarrow \text{if} \cdot E \text{ then } S$
 $E \rightarrow \cdot \text{exp}$

$S_3: S \rightarrow \text{stat} \cdot$

$S_4: S \rightarrow \text{if } E \cdot \text{ then } A \text{ else } S$
 $S \rightarrow \text{if } E \cdot \text{ then } S$

$S_5: E \rightarrow \text{exp} \cdot$

$S_6: S \rightarrow \text{if } E \text{ then} \cdot A \text{ else } S$
 $S \rightarrow \text{if } E \text{ then} \cdot S$
 $A \rightarrow \cdot \text{if } E \text{ then } A \text{ else } A$
 $A \rightarrow \cdot \text{stat}$
 $S \rightarrow \cdot \text{if } E \text{ then } A \text{ else } S$
 $S \rightarrow \cdot \text{if } E \text{ then } S$
 $S \rightarrow \cdot \text{stat}$

$S_7: S \rightarrow \text{if } E \text{ then } A \cdot \text{ else } S$

$S_8: S \rightarrow \text{if } E \text{ then } S \cdot$

$S_9: A \rightarrow \text{if} \cdot E \text{ then } A \text{ else } A$
 $S \rightarrow \text{if} \cdot E \text{ then } A \text{ else } S$
 $S \rightarrow \text{if} \cdot E \text{ then } S$
 $E \rightarrow \cdot \text{exp}$

$S_{10}: A \rightarrow \text{stat} \cdot$
 $S \rightarrow \text{stat} \cdot$

$S_{11}: S \rightarrow \text{if } E \text{ then } A \text{ else} \cdot S$
 $S \rightarrow \text{if } E \text{ then } A \text{ else} \cdot S$
 $S \rightarrow \text{if } E \text{ then } S$
 $S \rightarrow \cdot \text{stat}$

→ PRÓXIMA PAGINA

S_{12} : $A \rightarrow \text{if } E \text{ then } A \text{ else } A$
 $S \rightarrow \text{if } E \text{ then } A \text{ else } S$
 $S \rightarrow \text{if } E \text{ then } S$

S_{13} : $S \rightarrow \text{if } E \text{ then } A \text{ else } S$

S_{14} : $A \rightarrow \text{if } E \text{ then } \cdot A \text{ else } A$
 $S \rightarrow \text{if } E \text{ then } \cdot A \text{ else } S$
 $S \rightarrow \text{if } E \text{ then } \cdot S$
 $A \rightarrow \cdot \text{if } E \text{ then } A \text{ else } A$
 $A \rightarrow \cdot \text{stat}$
 $S \rightarrow \cdot \text{if } E \text{ then } A \text{ else } S$
 $S \rightarrow \cdot \text{if } E \text{ then } S$
 $S \rightarrow \cdot \text{stat}$

S_{15} : $A \rightarrow \text{if } E \text{ then } A \cdot \text{else } A$
 $S \rightarrow \text{if } E \text{ then } A \cdot \text{else } S$

S_{16} : $A \rightarrow \text{if } E \text{ then } A \text{ else } \cdot A$
 $S \rightarrow \text{if } E \text{ then } A \text{ else } \cdot S$
 $A \rightarrow \cdot \text{if } E \text{ then } A \text{ else } A$
 $A \rightarrow \cdot \text{stat}$
 $S \rightarrow \cdot \text{if } E \text{ then } A \text{ else } S$
 $S \rightarrow \cdot \text{if } E \text{ then } S$
 $S \rightarrow \cdot \text{stat}$

S_{17} : $A \rightarrow \text{if } E \text{ then } A \text{ else } A \cdot$

Tabela SLR(1):

Estado	Ação							GOTO		
	if	then	else	stat	exp	\$		S	A	E
0	S_2			S_3				1		
1						acc				
2					S_5					4
3						r_3				
4		S_6								
5		r_6								
6	S_9			S_{10}				8	7	
7			S_{11}							
8						r_2				
9					S_5					12
10			r_5			r_3				
11	S_2			S_3				13		
12		S_{14}								
13						r_1				
14	S_9			S_{10}				8	15	
15			S_{16}							
16	S_9			S_{10}				13	17	
17			r_4							

2. A gramática é SLR(1), pois não há conflitos na tabela SLR(1)
1. Pela contrapositiva abordada inicialmente, conclui-se que a gramática não é ambígua.

4. Considere a gramática G abaixo:

$S \rightarrow B$
 $B \rightarrow Ab \mid aa$
 $A \rightarrow a \mid bAa$

Pede-se:

1. Mostre a Máquina de Estados Finita para análise LR(0) da linguagem gerada por G .
2. G não é SLR(1). A tabela de Ações construída a partir da MEF acima e as informações de FOLLOW devem confirmar isso. Indique essa confirmação em detalhe.
3. G é LaLR(1)? Mostre os conjuntos FOLLOW para os não-terminais da gramática e construa a partir da MEF LR(0) que você mostrou no item 1 os lookaheads LaLR(1) para cada item do tipo redução na MEF. Deve haver 6 itens deste tipo nos estados da sua MEF.
4. G é LR(1)?

1. gramática aumentada de G :

(0) $S' \rightarrow S \$$
 (1) $S \rightarrow B$
 (2) $B \rightarrow Ab$
 (3) $B \rightarrow aa$
 (4) $A \rightarrow a$
 (5) $A \rightarrow bAa$

MEF LR(0):

S_0 : $S' \rightarrow \cdot S$
 $S \rightarrow \cdot B$
 $B \rightarrow \cdot Ab$
 $B \rightarrow \cdot aa$
 $A \rightarrow \cdot a$
 $A \rightarrow \cdot bAa$

S_1 : $S' \rightarrow S \cdot$
 S_2 : $S \rightarrow B \cdot$
 S_3 : $B \rightarrow A \cdot b$
 S_4 : $B \rightarrow a \cdot a$
 $A \rightarrow a \cdot$
 S_5 : $A \rightarrow b \cdot Aa$
 $A \rightarrow \cdot a$
 $A \rightarrow \cdot bAa$
 S_6 : $B \rightarrow Ab \cdot$
 S_7 : $B \rightarrow aa \cdot$
 S_8 : $A \rightarrow bA \cdot a$
 S_9 : $A \rightarrow a \cdot$
 S_{10} : $A \rightarrow bAa \cdot$

2. Tabela SLR(1):

Estado	Ação			GOTO		
	b	a	\$	S	B	A
0	S5	S4		1	2	3
1			acc			
2			r1			
3	S6					
4	r4	r7				
5	S5	S9				8
6			r2			
7			r3			
8		S10				
9	r4	r7				
10	r5	r5				

$First(S') =$
 $First(S) =$
 $First(A) =$
 $First(B) =$
 $\{a, b\}$
 $Follow(S') = Follow(S) = Follow(B) = \{ \$ \}$
 $Follow(A) = \{ b, a \}$

NA PRÓXIMA PÁGINA

→ É possível verificar que G não é SLR(1) pela tabela gerada. Na análise do estado 4, há um conflito shift-reduce, no qual o parser não sabe se reduz a regra $A \rightarrow a$ ou se prossegue com a leitura do símbolo "a" na configuração $B \rightarrow a \cdot a$, passando para o estado 7.

→ Tal redução ^{r4}aprove no estado 4 porque "a" é um follow de A, e $A \rightarrow a \cdot$ é um item presente nesse estado.

3. Conjunto Follow:

$$\text{Follow}(S') = \text{Follow}(S) = \text{Follow}(B) = \{ \$ \}$$

$$\text{Follow}(A) = \{ b, a \}$$

Itens do tipo redução e seus lookaheads:

1. $S' \rightarrow S \cdot, \$$
2. $S \rightarrow B \cdot, \$$
3. $B \rightarrow Ab \cdot, \$$
4. $B \rightarrow aa \cdot, \$$
5. $A \rightarrow a \cdot, a$
6. $A \rightarrow ba \cdot, b/a$

Tabela LALR(1):

Estado	Ação			Goto		
	b	a	\$	S	B	A
0	S5	S4		1	2	3
1			acc			
2			r1			
3	S6					
4	r4	S7				
5	S5	S9				8
6			r2			
7			r3			
8		S10				
9		r4				
10	r5	r5				

→ A gramática G é LALR(1), pois não há conflitos na tabela LALR(1).

4. Pela hierarquia dos parsers que analisam linguagens livres de contexto, sabe-se que de uma gramática, que descreve determinada linguagem, é LALR(1), ela também será LR(1) (LR(1) é mais poderoso que LALR(1)). Logo, pela resposta do item 3, também pode-se concluir que G é LR(1).

5. Seja G a seguinte gramática:

$$S \rightarrow P$$

$$P \rightarrow (P)P \mid \epsilon$$

Ela gera todas as cadeias de parênteses balanceados. Pede-se:

1. Escreva a forma aumentada de G (com o símbolo \$).
2. Construa os estados LR(0) e a MEF associada.
3. Mostre os conjuntos FOLLOW.
4. Mostre a tabela SLR(1).
5. Responda se G é SLR(1), LR(0), ambas, ou nenhuma delas.

1. Forma aumentada de G :

- (0) $S' \rightarrow S\$$
- (1) $S \rightarrow P$
- (2) $P \rightarrow (P)P$
- (3) $P \rightarrow \epsilon$

2. Estados LR(0) e MEF:

3. Conjuntos follow:

$\text{Follow}(S') = \text{Follow}(S)$
 $= \{ \$ \}$

$\text{Follow}(P) = \{ \$,) \}$

4. Tabela SLR(1):

Estado	Ação			GoTo	
	()	\$	S	P
0	S3	r3	r3	1	2
1			acc		
2			r1		
3	S3	r3	r3		4
4		S5			
5	S3	r3	r3		6
6		r2	r2		

5. G é SLR(1), pois não há conflitos na tabela SLR(1).
 Entretanto, G não é LR(0), pois não há conflitos de shift-reduce nos estados 0, 3 e 5, já que o LR(0) não pode ver nenhum token de lookahead, portanto não saberá se deve aplicar um shift e ler o símbolo "(" (shift 3), ou reduzir a regra ϵ (reduce 3).

Boa Prova!