

8,0

Prova de Compiladores

Nome: *Caroline Ghagor*

Data: *08/05/19*

1. (1,0 ponto) (ENADE 2014 – 15) Considere as seguintes expressões regulares.

$$ER^1 = a (a \cup b)^*$$

$$ER^2 = b (a \cup b)^*$$

Se $L(ER)$ é uma linguagem associada a uma ER, é correto afirmar que:

(a) $L(ER^2) = \{w \mid w \text{ termina com } b\}$

(b) $L(ER^1) = L(ER^2)$

(c) Se ER^3 é uma expressão regular igual a intersecção entre $L(ER^1)$ e $L(ER^2)$, então $L(ER^3)$ é uma linguagem livre do contexto

(d) Um autômato finito não determinístico que reconhece $L(ER^1) \cup L(ER^2)$ tem, pelo menos, quatro estados.

☒ (e) Existe um autômato finito determinístico cuja linguagem é $L(ER^1) \cup L(ER^2)$

2. (1,0 ponto) (POSCOMP 2017 – 68) A tarefa principal de um analisador léxico consiste em ler os caracteres da entrada do programa-fonte, agrupá-los em lexemas e gerar uma sequência de tokens que será enviada ao analisador sintático. Sobre o analisador léxico, analise as assertivas abaixo, marcando V ou F, justificando as falsas.

I. (☒) Além da identificação de lexemas, outras tarefas podem ser realizadas por esse analisador, tais como: remoção de comentários e espaços em branco e a associação de mensagens de erros às linhas do programa-fonte.

II. (☒) Token é a unidade básica do texto-fonte. Sempre é representado por três informações: a classe do token, que representa o tipo do token reconhecido, o valor do token, que é o texto do lexema reconhecido e a posição que indica o local do texto-fonte (linha e coluna) onde ocorreu o token.

III. (☒) Expressões regulares e geradores de analisadores léxicos são notações utilizadas para especificar os padrões de lexemas.

IV. (☒) Na análise léxica, uma representação intermediária do tipo árvore é criada. Esta apresenta a estrutura gramatical da sequência de tokens.

3. (3,0 ponto) Dada a gramática E e o método intuitivo para a construção da tabela de precedência abaixo, apresente a tabela de análise de precedência de operadores e os passos deste analisador para a seguinte palavra: $(id - id * id \wedge (id)) \$$

$$E \rightarrow E - E \mid E * E \mid E \wedge E \mid (E) \mid id$$

Método intuitivo:

Este método obtém as relações de precedência a partir do conhecimento da associatividade e precedência dos operadores da gramática. Considere dois operadores θ_1 e θ_2 .

- 1) se o operador θ_1 tem maior precedência que o operador θ_2 , então θ_1 (na pilha) $>$ θ_2 (na entrada) e θ_2 (pilha) $<$ θ_1 (entrada).
- 2) se θ_1 e θ_2 tem igual precedência (ou são iguais) e são associativos à esquerda, então $\theta_1 > \theta_2$ e $\theta_2 > \theta_1$; se são associativos à direita, então $\theta_1 < \theta_2$ e $\theta_2 < \theta_1$.
- 3) para todos os operadores θ , tem-se:

$\theta < id$	$\theta < ($	$) > \theta$	$\theta > \$$	$id > \theta$	$(< \theta$
$\theta >)$	$\$ < \theta$	$(< ($	$\$ < ($	$) > \$$	$(=)$
$\$ < id$	$id > \$$	$) >)$	$(< id$	$id >)$	$\theta < \theta$

4. (3,0 pontos) Dada a tabela LR abaixo, apresente os passos de um analisador ascendente para reconhecer as palavras a seguir:

- a) cccccdd
- b) cdccccdc
- c) Por fim, descreva a linguagem reconhecida por esta gramática, listando as suas 10 menores palavras.

	c	d	\$	S	C
0	e3	e4		1	2
1			AC		
2	e6	e7			5
3	e3	e4	r4		8
4	r3	r3			
5			r1		
6	e6	e7			9
7			r3		
8	r2	r2			
9			r2		

Reduzido:

Regra:

$$(1) S \rightarrow CC$$

$$(2) C \rightarrow cC$$

$$(3) C \rightarrow d$$

$$S \rightarrow cC$$

$$C \rightarrow cC \mid d$$

Caroline Chazar

3)

	id	-	*	^	()	\$
id		>	>	>	>	>	
-	<		<	<	<	>	
*	<	>		<	<	>	
^	<	>	>		<	>	
(<	<	<	<		=	
)		>	>	>	>		
\$	<	<	<	<	<		AC

\leq ou $=$
 \rightarrow empilha
 $>$ ou $<$ ou $=$
 \rightarrow reducing

Pilha	Entrada	Ação
\$	$(id - id * id ^ (id)) \$$	empilha
\$($id - id * id ^ (id)) \$$	emp.
\$(id	$- id * id ^ (id)) \$$	reduz $E \rightarrow id$
\$(E	$- id * id ^ (id)) \$$	emp.
\$(E-	$id * id ^ (id)) \$$	emp.
\$(E-id	$* id ^ (id)) \$$	reduz $E \rightarrow id$
\$(E-E	$* id ^ (id)) \$$	emp.
\$(E-E*	$id ^ (id)) \$$	emp.
\$(E-E*id	$^ (id)) \$$	reduz $E \rightarrow id$
\$(E-E*E	$^ (id)) \$$	emp.
\$(E-E*E^	$(id)) \$$	emp.
\$(E-E*E^($id)) \$$	emp.
\$(E-E*E^(id	$) \$$	reduz $E \rightarrow id$
\$(E-E*E^(E	$) \$$	emp.
\$(E-E*E^(E	$) \$$	reduz $E \rightarrow (E)$
\$(E-E*E^(E	$) \$$	reduz $E \rightarrow E^E$
\$(E-E*E^E	$) \$$	reduz $E \rightarrow E * E$
\$(E-E*E	$) \$$	reduz $E \rightarrow E - E$
\$(E-E	$) \$$	emp.
\$(E	$) \$$	reduz $E \rightarrow (E)$
\$(E)	$) \$$	
\$E	$) \$$	ACEITA!

4) a)

Pilha	Entrada	Ação
O	ccccccdd\$	emp. 3
Oc3	ccccccdd\$	"
Oc3c3	ccccdd\$	"
Oc3c3c3	cccd\$	"
Oc3c3c3c3	cccd\$	"
Oc3c3c3c3c3	cd\$	"
Oc3c3c3c3c3c3	dd\$	emp. 4
Oc3c3c3c3c3c3d4	d\$	reduz 3
Oc3c3c3c3c3c3C2	d\$	reduz 2
Oc3c3c3c3c3C8	d\$	"
Oc3c3c3c3C8	d\$	"
Oc3c3c3C8	d\$	"
Oc3c3C8	d\$	"
OC2	d\$	emp. 7
OC2d7	\$	reduz 3
OC2C5	\$	reduz 1
OS1	\$	AC!

b)

Pilha	Entrada	Ação
0	cdcccdcd\$	emp. 3
0c3	dcccdcd\$	emp. 4
0c3d4	cccdcd\$	reduz. 3
0c3c8	cccdcd\$	reduz. 2
0C2	cccdcd\$	emp. 6
0C2c6	cccdcd\$	emp. 6
0C2c6c6	cccdcd\$	"
0C2c6c6c6	cdcd\$	"
0C2c6c6c6c6	dc\$	emp. 7
0C2c6c6c6c6cd7	c\$	ERRO!

c) {dd, cdd, dcd, cdcd, cddd, dccd, cdcd, dccc, cccdd, ccdcd, ...}

É uma linguagem com recursividade à direita que contém pelo menos duas ocorrências de d.

1) E.

2) I. ☒

II. ☐ F, nem sempre é representado por 3 informações, o token. Pela informação de classe, pode-se não utilizar a informação de valor, podendo utilizar somente valores de classe e posição para representar.

III. ☐ F, para especificar e gerar tokens pode-se utilizar ER, GR e AFR.

IV. Na análise léxica não apenas gerador tokens, na análise sintática é que não gerador os átomos p/a estrutura gramatical. ☐ F