

**EA876/2010 - Primeira Prova**  
**Sem consulta – Duração: 120 minutos**

Nome: \_\_\_\_\_ RA: \_\_\_\_\_

**Questão 1** (3 pontos) – Seja a expressão regular para números em hexadecimal:

**$0x(D|L)(D|L)^*$**

Nesta expressão **D** representa dígitos de 0 a 9 e **L** representa letras de A à F.

- determine o autômato finito não determinístico para esta expressão.
- determine o autômato finito determinístico para esta expressão.
- minimize o autômato obtido no item anterior.

**Questão 2** (2 pontos) – Seja a gramática livre de contexto (**E** é o símbolo sentencial):

$E \rightarrow E + T$

$E \rightarrow T$

$T \rightarrow T * F$

$T \rightarrow F$

$F \rightarrow (E)$

$F \rightarrow id$

Esta gramática possui seguinte tabela de deslocamento e redução:

	Id	+	*	(	)	\$
\$	D			D		
E		D		D	D	<u>R</u>
T		R	D		R	R
F		R	R		R	R
id		R	R		R	R
+	D			D		
*	D			D		
(	D			D		
)		R	R		R	R

Mostre que a expressão  **$(id + id)*id$**  é válida nesta gramática e construa durante o processo de análise a árvore sintática para esta expressão.

**Questão 3** (1,5 pontos) – Escreva a gramática da questão 1 em Lex/Flex e a da questão 2 em YACC/Bison.

**Questão 4** (1,5 pontos) – O que é uma gramática ambígua? Como ambigüidades podem ser tratadas no processo de compilação?

**Questão 5** (1 ponto) – Cite duas regras sintáticas típicas das linguagens de programação que não podem ser representadas em gramáticas livres de contexto. Como tais regras são tratadas pelos compiladores?

**Questão 6** (1 ponto) – Descreva três erros detectados no processo de análise semântica.