Disc.: LFA - Linguagens Formais e Autômatos	Lista de Exercícios - Parte 2
Prof.: Leandro Israel Pinto	

1. O que é uma gramática ambígua? Dê um exemplo.

2. Num compilador, aquilo que for reduzido primeiro terá as instruções geradas primeiro no assembly/binário. Logo, uma gramática ambígua pode permitir a redução de + antes de $*$. Faça uma gramática não ambígua para expressões aritméticas e mostre a árvore de derivação para as expressões a seguir. A gramática deve sempre reduzir primeiro $*$ e /:

   a) $n + n * n$

   b) $9 + b * 7 - 8 + a/8 + 1$

   c) $8 * 8 + 6/5$

   d) $a + b - 8 * 2 + 3$

3. (1,0) Marque V para verdadeiro e F para falso:

   a) ___ AP Determinístico tem menos poder computacional que um AP Não-Determinístico;

   b) ___ Um AP não pode processar uma linguagem regular;

   c) ___ Uma GLC não pode ser transformada em um AP de forma trivial;

   d) ___ Máquina de Turing é considerado o mais geral dispositivo de computação. Ela pode inclusive processar linguagens sensíveis ao contexto;

   e) ___ Um AP de duas ou mais pilhas tem o mesmo poder computacional que um AP de apenas uma pilha.

   f) ___ O que diferencia as linguagens na hierarquia de Chomsky é basicamente a restrição aplicada sobre a gramática.

4. (1,5) Construa autômatos de pilha determinísticos para:

   a) $L = \{a^n b^n \mid n >= 0\}$

   b) $L = \{a^n (ba) * b^{3n} \mid m >= 0\}$

   c) $L = \{(a + b + c) * \mid |w|_a = |w|_b\}$

   d) $L = \{(a + b)^n (c + d)^n \mid |w|_a = |w|_b\}$

5. Considere a gramática $G = (V, T, P, S)$, onde:

$$P = \begin{Bmatrix} S \to aKb \mid KLj \mid \epsilon, & K \to Kk \mid lL \\ L \to nLM \mid \epsilon, & M \to mMm \mid S \end{Bmatrix}$$

   Determine o conjunto de produções D segundo o algoritmo a seguir:

$$\begin{array}{l} D = \emptyset \\ \text{para todo } A \to \alpha B, \text{ onde } A, B \in V \\ \quad \text{faça } D = D \cup \{C \to \alpha A \beta\} \\ \quad \text{onde } |\alpha| > 0 \text{ e } |\beta| > 0 \end{array}$$

6. Um programador está fazendo um analisador sintático para verificar arquivos de uma linguagem de programação como mostrada a seguir:

```
a = 10;
b = 0;
enquanto (a > 0) faca
    b = b*a;
    a = a + 1;
    se (b > 200) entao
        escreva "b maior que 200\n";
    fim_se
fim_enquanto
```

   Parte da gramática já está pronta como segue:

$$\begin{array}{rl} Arquivo & \to Arquivo\ Comandos \mid \varepsilon \\ Comandos & \to Se \mid Enquanto \mid Attr \mid Escreva \\ Se & \to ?, \quad Enquanto \to ? \\ Attr & \to ID = LogicExpr\,; \\ LogicExpr & \to LogicExpr \text{ and } ExpRel \\ & \quad \mid LogicExpr \text{ or } ExpRel \mid ExpRel \\ ExpRel & \to ExpRel < Expr \mid ExpRel > Expr \\ & \quad \mid Expr \\ Expr & \to ? \\ Escreva & \to \text{escreva } STRING\,; \end{array}$$

   Complete os comandos **se** e **enquanto** nessa gramática, considerando que a expressão deve conter parênteses.