

SEGUNDA AVALIAÇÃO DE LÓGICA COMPUTACIONAL (PCS-2046)

sem consulta. Duração: 90 minutos

1. (2 pontos-A.P.) Construa um autômato de pilha (AP) determinístico que aceite a linguagem gerada pela gramática G cujas regras são $\{S \rightarrow S * F, S \rightarrow F, F \rightarrow x_1, F \rightarrow x_2, F \rightarrow (S)\}$, onde S é o símbolo inicial, F é um não-terminal e “*”, “(”, “)”, “x”, “1” e “2” são símbolos terminais. **Você deve usar** a transformação de gramática para autômato segundo o teorema da equivalência entre a linguagem gerada por uma gramática Livre de Contexto e um AP, depois modifique a gramática e o autômato convenientemente para tornar o AP determinístico.
2. (1 ponto-A.P. e L.L.C.) Através de algum conceito estudado como, por exemplo, o teorema do bombeamento ou o autômato com pilha, identifique (e justifique) o tipo da linguagem e o porquê dela ser ou não livre de contexto:
 - a) $\{x^r y^s z^t : r, s, t > 0, r = s + t\}$; (Obs. Use o modelo de autômato ou alguma propriedade de fechamento);
 - b) $\{x^m y^n x^m y^n : m, n > 0\}$; (Obs. Use o bombeamento).
3. (2 pontos) Seja $f: \mathbb{N} \rightarrow \mathbb{N}$ uma função recursiva primitiva. Defina-se $F: \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ como:

$$F(\underline{n}, \underline{m}) = \underbrace{f(f(f(\dots f(\underline{n}) \dots)))}_{\underline{n} - \underline{m} \text{ composições (operação minus, em } \mathbb{N})}$$

através de $\underline{n} - \underline{m}$ composições de função, ou seja, caso \underline{m} seja maior ou igual a \underline{n} não haverá composição, $F(\underline{n}, \underline{m}) = \underline{n}$. Mostre que F é recursiva primitiva (encontre uma formulação recursiva para F).

4. (2 pontos) Defina uma gramática para a linguagem abaixo, diga a quais tipos (todos) ela pertence (dentro da hierarquia de Chomsky), depois construa uma máquina de Turing para decidir esta linguagem.

$$L = \{ww \mid w \in \{a, b\}^*\}, \text{ linguagem "cópia"}$$

5. (2 pontos) Construa uma máquina de Turing **composta** (chamada de **MUL**) para efetuar a operação "**produto**" de dois números naturais escritos em binário. Esta máquina deve receber em sua fita de entrada dois parâmetros (os números \mathbf{N}_1 e \mathbf{N}_2), e produzir como resultado ($\mathbf{N}_1 * \mathbf{N}_2$). Você pode usar como sub-máquinas para o problema as seguintes:

PLUS($\mathbf{N}_1, \mathbf{N}_2$): $(s, \# \mathbf{N}_1 \# \mathbf{N}_2 \#) \vdash_M^* (\mathbf{h}, \# \mathbf{r} \#)$, onde “ \mathbf{r} ” representa a adição ($\mathbf{N}_1 + \mathbf{N}_2$).

COPY(\mathbf{N}_1): $(s, \# \mathbf{N}_1 \#) \vdash_M^* (\mathbf{h}, \# \mathbf{N}_1 \# \mathbf{N}_1 \#)$, onde \mathbf{N}_1 representa um número binário.

Obs.: Não é necessário retornar o resultado a partir da última posição da fita à esquerda, nem apagá-la, e você pode assumir qualquer configuração de MT.

6. (1 ponto) É possível resolver mais problemas ao usar um processador diferente, por exemplo, paralelo? E usar linguagens diferentes como, por exemplo, linguagem lógica? Justifique suas respostas.