

UNIVERSIDADE FEDERAL DO MARANHÃO Centro de Ciências Exatas e Tecnologia		Departamento de Informática - DEINF Internet: <a href="http://www.deinf.ufma.br">www.deinf.ufma.br</a>		2a AVALIAÇÃO	
Disciplina: Teoria da Computação		Curso: CIÊNCIA DA COMPUTAÇÃO		P	
Código 5607.5	Carga Horária: 60 horas	Créditos: 4.0.0		T	
Professor: Luciano Reis Coutinho		Email: <a href="mailto:luciano.rc@ufma.br">luciano.rc@ufma.br</a>		MEDIA	

Segunda Avaliação: Prova Escrita

Data: 20 de junho de 2022.

Aluno : \_\_\_\_\_

Código: \_\_\_\_\_

**INSTRUÇÕES**

- A prova deve ser realizada INDIVIDUALMENTE. Respostas iguais ocorrendo em provas de alunos diferentes são passíveis de anulação.
- Cada questão consiste em um enunciado e um conjunto de requisitos que uma resposta aceitável deve satisfazer. Respostas dadas que não atendam aos requisitos podem em última instância ser completamente desconsideradas durante a correção da prova. Tenham sempre em mente os requisitos ao dar as suas respostas.
- A interpretação das questões faz parte da avaliação. Caso ache um enunciado ambíguo ou impreciso escreva na folha de resposta sua interpretação e a correspondente resposta. Todas as questões devem ser interpretadas tendo em vista que foi discutido nas aulas de Teoria da Computação.
- Todas as questões devem ser respondidas em arquivo .DOC ou PDF. Ao final, tanto o arquivo de questões quanto o arquivo de respostas devem ser enviados via SIGAA.
- **O tempo total de prova é de 100 min; indo das 14:00 as 15:40.** Depois desse horário, há 20min de tolerância para o envio das respostas. Após 16:00 não será mais possível o envio.

**QUESTÕES**

1. **(2,0 pontos)** Considerando a codificação de programas monolíticos como números naturais que foi discutida durante as aulas (a codificação de Gödel), Descreva o PASSO a PASSO de como o programa iterativo abaixo --- após traduzido para a forma monolítica --- pode ser representado por meio de um único número natural.

**(se T0 então até T1 faça (G;F) senão G)**

2. **(2,0 pontos)** Escreva uma macro  $R := A \% B$  para a máquina NORMA que armazena em R o resto da divisão inteira de um registrador A por outro registrador B. Lembre-se que em NORMA, apenas as operações de **incremento** e **decremento**, e o teste de **zero**, são definidos, conforme implementadas no código C++ abaixo. Assim, quaisquer outras operações e testes necessários DEVEM também ser escritos explicitamente na resposta da questão como macros auxiliares. Para facilitar, assuma como já escritas as macros que realizam **atribuições**.

using namespace std;

// Máquina NORMA (implementação em C++)

// -----

// Memória: com 256 registradores

#define INF 256

unsigned long R[INF]; //

// funções de entrada/saída:

#define ent memset(R, 0, sizeof (R)),\

cout &lt;&lt; "X = ", cin &gt;&gt; R['X']

#define sai cout &lt;&lt; "Y = ", cout &lt;&lt; R['Y'] &lt;&lt; "\n"

// conjunto de operações:

#define inc(K) R[K] += 1 // K := K + 1

#define dec(K) R[K] -= R[K] &gt; 0 ? 1 : 0 // K := K - 1

// conjunto de testes:

#define zero(K) R[K] == 0 // K == 0

// operações auxiliares de atribuição (já implementadas)

```
#define set(K,n)  R[K] = n           // K := n
#define mov(A,B)  R[A] = R[B]       // A := B
```

3. **(2,0 pontos)** Escreva uma MAQUINA DE TURING que calcule a função  $f(n)=2n$ . Especifique detalhadamente os elementos  $(\Sigma, Q, \Pi, q_0, F, V, \beta, *)$ . Desenhe a função de transição  $\Pi$  como uma máquina de estados.
4. **(1,0 ponto)** No contexto da Teoria da Computação, assinale V para verdadeiro ou F para falso nas afirmações abaixo. Tenha cuidado: cada resposta errada irá anular uma resposta certa! Assim, caso não tenha certeza sobre uma afirmação assinale SR para SEM RESPOSTA.
  - (a) O hardware dos computadores modernos mais comuns são exemplares físicos de máquinas de registradores. Neste contexto, em teoria, pode-se afirmar que para qualquer função matemática calculada através de um programa de computador, existe uma Máquina de Turing que também computa a mesma função.
  - (b) A máquina NORMA com apenas dois registradores não é capaz de simular qualquer máquina de TURING com fita de armazenamento infinita.
  - (c) Se  $L$  é uma linguagem recursiva, por definição há uma máquina de Turing que a reconhece e está máquina sempre para para qualquer entrada.
  - (d) Toda linguagem recursiva é também enumerável recursivamente.
  - (e) As máquinas de Turing não-deterministas são mais poderosas (reconhecem uma classe maior de linguagens/ computam um número maior de funções) que as máquinas de Turing padrão.
  - (f) Máquinas de Turing com mais de uma fita ou com fitas infinitas de ambos os lados não são mais poderosas (reconhecem uma classe maior de linguagens/ computam um número maior de funções) que as máquinas de Turing padrão.
5. **(1,0 ponto)** Considerando a máquina de Turing, marque a alternativa errada:
  - a** Uma função parcial  $f: (\Sigma^*)^n \rightarrow \Sigma^*$  é dita função Turing-computável, ou simplesmente função computável, se existe uma máquina de Turing  $M = (\Sigma, Q, \Pi, q_0, F, V, \beta, \odot)$  que computa  $f$ ;
  - b** Uma função total  $f: (\Sigma^*)^n \rightarrow \Sigma^*$  é dita função Turing-computável total ou simplesmente função computável total, se existe uma máquina de Turing  $M = (\Sigma, Q, \Pi, q_0, F, V, \beta, \odot)$  que computa  $f$  e sempre para para qualquer entrada;
  - c** A função programa total sempre induz uma função Turing-computável total;
  - d** A definição de uma função Turing-computável total garante que a função está definida para todos os valores de entrada;
  - e** Toda máquina de Turing vista como um reconhecedor de linguagem também pode ser vista como um processador de função.
6. **(1,0 ponto)** **(a)** Qual a diferença fundamental entre as classes das linguagens recursivas e das linguagens enumeráveis recursivamente? Apresente uma resposta detalhada baseada nas definições das duas classes (mínimo cinco linhas de texto), **(b)** Qual a importância de se distinguir estas duas classes? Apresente uma resposta detalhada (mínimo cinco linhas de texto).
7. **(1,0 ponto)** A hipótese de Church/Turing afirma que (marque a alternativa correta):
  - (a) Qualquer programa pode ser representado na forma de fluxogramas;
  - (b) Qualquer máquina abstrata é uma máquina universal;
  - (c) A codificação de conjuntos estruturados é o modo mais eficiente de representar uma máquina universal;
  - (d) Todo programa monolítico pode ser representado por meio de um programa iterativo;
  - (e) Qualquer função computada pode ser processada por uma máquina de Turing.