UNIVERSIDADE FEDERAL DO MARANHÃO Centro de Ciências Exatas e Tecnologia		Departamento de Informática - DEINF Internet: www.deinf.ufma.br		1a AVALIAÇÃO	
				P	7,0
Disciplina: Teoria da Computação		Curso: CIÊNCIA DA COMPUTAÇÃO		T	—
Código 5607.5	Carga Horária: 60 horas		Créditos: 4.0.0	MEDIA	
Professor: Luciano Reis Coutinho		Email: lrc@deinf.ufma.br		7,0	

Primeira Avaliação: Prova Escrita **Data: 20 de Janeiro de 2016.**

Aluno : Antonio Carlos Raposo **Código:** ____________

INSTRUÇÕES

- A prova deve ser realizada INDIVIDUALMENTE e SEM CONSULTA à livros, anotações, etc. O professor pode ser consultado. No entanto, o papel do professor é tirar dúvidas quanto ao entendimento das questões. O professor não irá atender a pedidos para saber se estão certas ou erradas suas questões. NÃO INSISTAM.
- 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.
- A prova tem **6 questões**. Todas as questões devem ser respondidas usando CANETA PRETA ou AZUL. O tempo total de prova é de 100 min.

QUESTÕES

1. (1,0 ponto) Marque a resposta INcorreta:

∃ - existe

(a) Um programa pode ser descrito como um conjunto estruturado de instruções que capacitam uma máquina a realizar sucessivamente certas operações básicas e testes sobre os dados iniciais fornecidos, com o objetivo de transformar estes dados numa forma desejável.

(b) Uma computação é, resumidamente, um histórico do funcionamento de uma máquina para um dado programa, considerando um valor inicial.

∀ - pra todo

(c) Máquinas podem ser definidas como programas em execução, pois cada instrução de qualquer programa sempre tem uma interpretação numa máquina.

(d) A relação valor de entrada → valor de saída induzida pelas computações de um programa em uma dada máquina dá origem à noção de Função computada.

(e) De modo geral, funções computadas são funções parciais.

2. (1,0 ponto) Sejam P e Q programas e M e N máquinas. Marque a resposta CORRETA:

~~(a) se ∃M ∃N (<P,M> = <Q, N>), então P e Q são programas equivalentes fortemente.~~ ∀M ∀N

~~(b) se ∀M ∃N (<P,M> = <Q, N>), então P e Q são programas equivalentes fortemente.~~ ∀N

(c) se ∃P ∃Q (<P,M> = <Q, N>), então M é equivalente a N. F

(d) se ∃P ∃Q (<P,M> = <Q, N>), então N simula fortemente M.

D (e) se ∀P ∃Q (<P,M> = <Q, N>), então N simula fortemente M.

3. (2,0 pontos) Tendo em vista as definições de **programas iterativos, monolíticos** e **recursivos** e a definição de **equivalência forte** entre programas que foram apresentadas durante as aulas, traduza o programa abaixo primeiro para um programa monolítico e em seguida para um programa recursivo, ambos equivalentes fortemente ao programa iterativo original.

```
até T faça (
   F; ( se T então √ senão F;G )
)
```

4. (2,0 pontos) Utilizando o método discutido em sala de aula, verifique se os programas P1 e P2 a seguir são ou não são equivalentes fortemente. Lembrete do método: (0) reescreva os programas como fluxogramas (1) transforme os fluxogramas para instruções rotuladas compostas; (2) identifique e simplificando ciclos infinitos; (3) construa a cadeia de conjuntos B0, B1, ..., Bk de rótulos equivalentes fortemente; (4) caso Bk = { } os programas são equivalentes fortemente, caso contrário, não o são.

P1: 1

1: faça F vá_para 2 2
2: se T então vá_para 3 senão vá_para 5
3: faça G vá_para 4 3
4: se T então vá_para 1 senão vá_para 0
5: faça F vá_para 6 4
6: se T então vá_para 7 senao vá_para 2
7: faça G vá_para 8 5
8: se T então vá_para 6 senão vá_para 0

duplica

R_1 def se T ✓ senão Sub; R; Add; Add

P2: 6

1: faça F vá_para 2 7
2: se T então vá_para 3 senão vá_para 1
3: faça G vá_para 4 8
4: se T então vá_para 1 senão vá_para 0

Pn é R_1 onde:

R_1 def sub; R_2

R_2 se T ✓ senão sub; R_1

5. (2,0 pontos) Escreva um programa P que compute a seguinte função: 1

<P,um_reg> : N → N

<P,um_reg>(x) = 2*(x - 1) 2 n

na máquina um_reg definida abaixo. Dica: lembre-se da função duplica discutida em sala de aula.

um_reg = < N, N, N, id , id , {ad, sub}, {zero} >

id : N → N , tal que id(n)=n
ad: N → N , tal que ad(n)=n+1
sub: N → N, tal que sub(n)=n-1, se n ≠ 0; sub(n)=0, se n=0

zero: N → {verdadeiro, falso}, tal que zero(0)=verdadeiro e zero(n)=falso, se n ≠ 0.

6. (2,0 pontos) No programa Q abaixo, T sendo o teste zero, F a operação ad e G a operação sub, diga qual a função computada por Q na máquina um_reg, escrevendo uma expressão que defina precisamente a função <Q, um_reg> . Em seguida, escreva no MÍNIMO 05 linhas de texto explicando de modo correto porque o programa Q na máquina um_reg computa a função que você definiu.

Programa Q:

```
1: se T então vá_para 9 senão vá_para 2
2: faça G vá_para 3
3: se T então vá_para 4 senão vá_para 5
4: faça F vá_para 9
5: faça G vá_para 6
6: se T então vá_para 7 senão vá_para 8
7: faça F vá_para 4
8: faça G vá_para 1
```

0 →
1 → 1
2 →
3 →

Boa Sorte!

g → sub
F ⇒ AD

R1 def ad -