UNIVERSIDADE FEDERAL DO MARANHÃO Centro de Ciências Exatas e Tecnologia	Departamento de Informática - DEINF Internet: www.deinf.ufma.br		3a AVALIAÇÃO
Disciplina: Teoria da Computação	Curso: CIÊNCIA DA COMPUTAÇÃO		P 8,5 T
Código 5607.5	Carga Horária: 60 horas	Créditos: 4.0.0	MEDIA
Professor: Luciano Reis Coutinho	Email: lrc@deinf.ufma.br		8,5

Terceira Avaliação: Prova Escrita **Data: __ abril de 2016**

Aluno : Alexsandro da Silva Soraiva **Código:** 2013002955

INSTRUÇÕES

- A prova deve ser realizada individualmente e sem consulta a livros, anotações, etc.
- 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 – sem exceção – devem ser respondidas na folha de respostas (papel almaço) que foi entregue junto com esta folha de enunciado. Use obrigatoriamente caneta para escrever as respostas. Respostas que não se encontram na folha de respostas não serão consideradas na correção.
- O tempo total de prova é de 100 min.

QUESTÕES

1. **(2,0 pontos) Funções recursivas de Kleene.** As funções recursivas de KLEENE são funções parciais definidas a partir de três funções básicas – a *função constante zero* $\mathbf{Z}(x)=0$, a *função sucessor* $\mathrm{S}(x)=x+1$ e as *funções de projeção* $\mathbf{U}^n_i(x_1,\dots,x_n)=x_i$ (uma para cada $\mathbf{n}$, $\mathbf{i} \in \mathbb{N}$) – utilizando os esquemas de *composição*, *recursão primitiva* e *minimização* de funções. Por exemplo, a função *soma*(x,y) pode ser obtida a partir da seguinte sequência:

   1) $S(x)=x+1$ – função básica *sucessor*
   2) $\mathbf{U}^3_3(x,y,z)=z$ – função básica *projeçao* n=3 e i=3
   3) $g_s(x,y,z)=S(\mathbf{U}^3_3(x,y,z))$ – *composição* de 1) com 2)
   4) $\mathbf{U}^1_1(x)=x$ – função básica *projeção* n=1 e i=1
   5) $soma(x,\ 0)=\mathbf{U}^1_1(x)$
      $soma(x,y+1)=g_s(x,y,soma(x,y))$ – *recursão primitiva* usando 4) e 3) no papel de f(x) e g(x), resp.

   Um outro exemplo é a definição da função *mult*(x,y):

   6) $\mathbf{Z}(x)=0$ – função básica *zero*
   7) $\mathbf{U}^3_1(x,y,z)=x$ – função básica *projeçao* n=3 e i=1
   8) $g_m(x,y,z)=soma(\mathbf{U}^3_1(x,y,z),\ \mathbf{U}^3_3(x,y,z))$ – *composição* de 5) com 7) e com 2)
   9) $mult(x,\ 0)=\mathbf{Z}(x)$
      $mult(x,y+1)=g_m(x,y,mult(x,y))$ – *recursão primitiva* usando 8) e 6) no papel de f(x) e g(x), resp.

   Dando continuidade, **MOSTRE** passo a passo como as funções fatorial *fat*(x)=x! e exponenciação *exp*(x,y)=x^y podem rigorosamente ser definidas como funções recursivas de KLEENE.

2. **(1,5 pontos)** Usando as definições recursivas das funções *soma*(x,y) e *mult*(x,y) dadas na questão anterior, mostre passo a passo como é calculado o valor *mult*(2,2).

3. **(1,5 pontos) Linguagem lambda**. Seja V um conjunto infinito de variáveis. A linguagem de termos lambda é o menor conjunto Λ definido indutivamente como segue: i) $\mathbf{x} \in \Lambda$, onde x é uma variável em V; ii) **(λx.M)**, onde x é uma variável em V e $M \in \Lambda$; e iii) **(M N)**, onde M, $N \in \Lambda$. Conforme discutido em sala de aula, termos lambda podem ser escritos omitido-se parentênses desde que algumas conveções sejam observadas. Para os termos lambda abaixo, reescreva-os tornando explícitos os parênteses que foram omitidos usando as conveções de notação discutidas em sala de aula.

   a) x x (x x x) x b) w (λxyz.x z(y z)) u v c) ux (yz) (λv.vy)

4. **(2,0 pontos)** No cálculo lambda, qualquer termo da forma (λx.M) N pode ser reescrito (reduzido, contraído) ao termo resultante da substituição de x por N dentro do termo M, ou seja,

$$(\lambda x.M)\ N \quad \rhd_\beta \quad M\,[x \leftarrow N]$$

Esta reescrita é conhecida como regra de redução β, ou β-redução. Aplicando a regra de β-redução (renomeado variáveis quando necessário) reduza os termos abaixo, passo a passo, a um termo mínimo (forma normal β):

   a) (λxy.x)(λu.u) b) (λxy.yx)(uv)zw

5. (1,0 pontos) Sejam as função um **1** e sucessor **S** definidas como sendo os seguintes lambda termos, respectivamente: **1** $\overset{\text{def}}{=}$ λfx.x e **S** $\overset{\text{def}}{=}$ λnfx.f (n f x). Mostre passo a passo qual o resultado da aplicação **(S 1)**. Ou seja, reduza **(S 1)** ao seu termo mínimo via β-redução(ões).

6. (2,0 pontos) No contexto da Teoria da Computação, conforme o que foi discutido durante as aulas, 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 NR para Não Respondida. Assinalando NR você não irá ganhar e nem perder pontos.

F a) Toda função recursiva de KLEENE é Turing-Computável (i.e., existe uma máquina de TURING que a computa); no entanto, o inverso não é verdadeiro (i.e., máquinas de TURING computam função que não são funções recursivas de KLEENE).

V b) Toda função definível do cálculo lambda é NORMA-Computável (i.e., existe um programa para a Máquina NORMA que a computa; e o inverso também é verdadeiro (i.e., toda função NORMA-Computável é lambda-definível).

V c) Um problema de decisão é dito parcialmente solucionável ou computável se existe um algoritmo que solucione o problema de tal maneira que sempre pare quando a resposta é afirmativa (ACEITA).

F d) O problema da auto-aplicação é totalmente solucionável.

F e) Se o problema da auto-aplicação pode ser reduzido ao problema da parada, e sendo o problema da auto-aplicação solucionável, então pode-se concluir que o problema da parada também é solucionável.

V f) Todo problema solucionável é parcialmente solucionável.

V g) Alguns problemas não-solucionáveis são parcialmente solucionáveis.

V h) Conforme o princípio da redução, se um dado problema A pode ser reduzido a um outro problema B, pode-se afirmar que se A é não solucionável então B também é não solucionável.

Boa Sorte!