UNIVERSIDADE FEDERAL DE SANTA MARIA — CURSO DE CIÊNCIA DA COMPUTAÇÃO - PROVA DE TEORIA DA COMPUTAÇÃO(ELC1008–TURMA)

Profa. Juliana Kaizer Vizzotto

25/06/2018

Nome: Matheus Einloft Nota: 5,0

DURAÇÃO DA PROVA: 09:00:00H–10:40H — INDIVIDUAL —

1. (2.0 pt) Assinale a afirmativa INCORRETA.

   (a) O cálculo-λ tem o mesmo poder de expressão que a máquina de Turing.

   (b) Os modelos teóricos de computação, como a máquina de Turing e o cálculo-λ, identificam formalmente a noção de procedimento computável.

   (c) No cálculo-λ, as funções $\lambda x.xy$ e $\lambda z.zy$ são equivalentes.

   (d) No cálculo-λ, a expressão $(\lambda x.xx)(\lambda y.yx)z$ reduz em n passos de beta-redução para xxz, considerando a estratégia de avaliação *full beta reduction* e as convenções vistas em aula.

   (e) No cálculo-λ, a expressão $(\lambda x.(\lambda y(xy))y)z$ reduz em n passos de beta-redução para yz, considerando a estratégia de avaliação *full beta reduction*.

2. (2.0 pt) Analise as seguintes afirmativas

   - I - No cálculo-λ, podemos codificar *booleanos* e *números naturais*, mas não conseguimos codificar *listas*, por exemplo.
   - II - No cálculo-λ é natural a definição de funções de alta ordem.
   - III - No cálculo-λ as funções são anônimas limitando o poder de expressão do mesmo.

   A análise permite concluir que estão CORRETAS

   (a) apenas as afirmativas I e II.

   (b) apenas a afirmativa II.

   (c) apenas as afirmativas II e III.

   (d) apenas a afirmativa III.

   (e) todas as afirmativas.

3. (1.5 pt) Explique as possíveis estratégias de avaliação que podemos utilizar no cálculo-λ. Mostre exemplos na sua resposta. Discuta a importância das estratégias de avaliação nas Linguagens de Programação de propósito geral, como *Java*, *C*, *Scheme*, *Haskell*, etc.

4. (2.0 pt) Dadas as regras para avaliação *chamada por valor* (*call-by-value*) vistas em aula. Apresente as modificações necessárias para que a avalição seja *chamada por nome* (*call-by-name*).

5. (2.5 pt) Considerando a seguinte especificação da sintaxe do cálculo-λ:

$$
\begin{array}{rcl}
< LambdaExp > & ::= & < Identifier > \\
 & | & (\textbf{lambda}(< Identifier >) < LambdaExp >) \\
 & | & (< LambdaExp >< LambdaExp >)
\end{array}
$$

   Escolha alguma linguagem de programação e defina uma função ou procedimento `(contains-beta-redex? e)`, o qual retorna `true` se a expressão do cálculo-λ, `e`, contém um beta-redex. Caso contrário, o procedimento retorna `false`. Um beta-redex é uma aplicação `(e1 e2)`, tal que `e1` é uma abstração, i.e., tem a forma `(lambda (x) e1)`. Note que um beta-redex pode ocorrer dentro de uma abstração lambda, i.e., a subexpressão $e1$ pode conter um beta-redex.

6. Explique as classes de problemas P e NP mostrando exemplos de problemas em cada classe.

Matthews Eimloft

③ Chamada por valor: é uma das mais utilizadas, sendo utilizada em linguagens como C. Ela consiste em, ao se chamar uma função, passar como argumento o valor das variáveis em si, ou seja, uma cópia delas (e não o endereço delas, como na chamada por referência). Isso faz com que o valor das variáveis originais não sejam alteradas, somente as das cópias delas.
Chamada por referência: utilizada na linguagem Haskell, ela consiste em, ao se chamar uma função, passar como argumento o endereço da variável. Isso faz com que não seja necessário criar uma cópia da variável em outro local de memória.
Avaliação por ordem normal: dá-se primeiro os redex mais à esquerda da expressão.

Full beta reduction: não há preferência para reduzir um redex de uma expressão, ou seja, pode-se escolher qual reduzir por primeiro.

(5) #TRUE = $\lambda x.\lambda y.x$

#FALSE = $\lambda x.\lambda y.y$

expressao =

def contains-beta-redex(expressao):

if(e1 == constante) # mais à esquerda precisa ser uma função lambda.

return false

else if(

exemplo: 2 ($\lambda x.x$)

↳ assim não há como reduzir, já contendo 1 de:

$\lambda x.x$ (2)

2

(4) Chamada por valor em C por exemplo, não utiliza ponteiros na função, pois é passado o valor da variável e não seu endereço. Para modificar para uma chamada por nome, é necessário utilizar ponteiros para alterar os valores das variáveis "originais".