

Nome: _____ Nota: _____

DURAÇÃO DA PROVA: 14:30:00H–16:10H — 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 avaliação seja *chamada por nome* (*call-by-name*).
5. (2.5 pt) Considerando a seguinte especificação da sintaxe do cálculo- λ :

$$\begin{aligned} \langle \text{LambdaExp} \rangle &::= \langle \text{Identifier} \rangle \\ &\quad | \text{lambda}(\langle \text{Identifier} \rangle) \langle \text{LambdaExp} \rangle \\ &\quad | \langle \text{LambdaExp} \rangle \langle \text{LambdaExp} \rangle \end{aligned}$$

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 ($e_1 \ e_2$), tal que e_1 é uma abstração, i.e., tem a forma $(\text{lambda } (x) \ e_1)$. Note que um beta-redex pode ocorrer dentro de uma abstração lambda, i.e., a subexpressão e_1 pode conter um beta-redex.

6. Explique como podemos codificar recursão no cálculo- λ utilizando o combinador de ponto fixo.