

Semântica das Linguagens de Programação

2º Teste (9 de Junho 2016) – Duração: 90 min

Questão 1 Considere o λ -calculus puro.

1. Indique o que entende por *forma normal*.
2. Apresente a sequência correspondente à *ordem aplicativa* de redução do termo abaixo indicado até à sua *forma normal*. Em cada passo de redução sublinhe o β -redex que foi seleccionado.

$$(\lambda n. \lambda x. \lambda y. n (\lambda z. y) x) ((\lambda a. \lambda f. \lambda b. f (a f b)) (\lambda x. x))$$

3. Considere o seguinte termo:

$$(\lambda a. \lambda x. x) ((\lambda y. (\lambda x. x x) y) (\lambda z. z z))$$

Indique, justificando, se o termo é fortemente normalizável, fracamente normalizável, ou não normalizável.

Questão 2 Considere a seguinte expressão P da linguagem de programação funcional estudada:

```
let fun  $\equiv \lambda x. \lambda l. \text{listcase } l \text{ of } (\text{False}, \lambda h. \lambda t. h = x),$   
    ex  $\equiv 1 :: \text{nil}$   
in fun (5-3) ex
```

1. Apresente a avaliação “*call-by-name*” da expressão P até à sua forma canónica.
2. Construa a árvore de prova do juízo de tipificação $\vdash P : \text{Bool}$
3. Defina nesta linguagem a função **atezero** que dá o segmento inicial de uma lista de inteiros até à primeira ocorrência de 0. Por exemplo, **atezero** $(7::1::5::0::3::2::0::8::\text{nil})$ deve dar como resultado $(7::1::5::\text{nil})$.

Questão 3 Pretende-se agora construir uma extensão da linguagem de programação funcional, por forma a incluir um tipo de árvores binárias polimórficas com informação apenas nas folhas (“*leaf trees*”).

1. Defina a sintaxe abstracta das novas expressões, do novo tipo, e as novas regras de inferência de tipo.
2. Indique as novas formas canónicas da linguagem e as novas regras de avaliação “*call-by-name*”.