Semântica das Linguagens de Programação

2º Teste (2 de Junho de 2022)

Cotação	Questão 1: 3, 2	Questão 2: 2, 3, 3	Questão 3: 3, 3, 1

Questão 1 Considere os seguintes termos do lambda calculus puro:

$$\begin{array}{rcl@{\qquad}rcl} K & \equiv & (\lambda a.\lambda b.\, a) & F & \equiv & (\lambda a.\lambda b.\, b) \\ S & \equiv & (\lambda x.\lambda y.\lambda z.\, x\, z\, (y\, z)) & A & \equiv & (\lambda x.\, x\, x) \end{array}$$

1. Apresente a sequência da *ordem aplicativa* de redução até à forma normal da expressão

$$F\,(S\,(F\,A)\,(K\,K\,A))$$

   Sublinhe o β-redex que é selecionado em cada passo de redução.

2. Apresente um exemplo de um λ-termo que é normalizável, mas não fortemente normalizável. Justifique a sua resposta.

Questão 2

1. Na linguagem funcional que estudou, defina uma função que recebe um inteiro e uma lista de inteiros, e calcula quantos elementos da lista são maiores do que o inteiro recebido.

2. Apresente, passo a passo, a sequência da avaliação *call-by-name* da expressão

$$\begin{array}{l} \mathsf{let\ sumfst} \equiv \lambda x.\, \lambda y.\ x + y.1\,, \\ \quad \mathsf{head} \equiv \lambda l.\, \mathsf{listcase}\ l\ \mathsf{of}\ (\,0,\ \lambda h.\, \lambda t.\, h\,) \\ \mathsf{in}\ \ \mathsf{sumfst}\ (\mathsf{head}\,(4{+}3){::}8{::}\mathsf{nil})\ \langle 2, 3{*}3\rangle \end{array}$$

3. Construa uma árvore de prova do juizo

$$\mathsf{head} : \mathsf{List\,Int} \to \mathsf{Int},\ a : \mathsf{List\,Int}\ \vdash\ \mathsf{let\ sumfst} \equiv \lambda x.\, \lambda y.\ x + y.1\ \mathsf{in\ sumfst}\,(\mathsf{head}\,a)\,\langle 1, 2\rangle\ : \mathsf{Int}$$

Questão 3 Pretende-se estender a linguagem de programação funcional, com um novo tipo de dados para representar uma sequência em que os elementos podem ser acrescentados do lado esquerdo (Esq) ou do lado direito (Dir) da sequência. Por exemplo, um equivalente em Haskell desta estrutura de dados seria:

```
data Seq a = Null | Esq a (Seq a) | Dir (Seq a) a
```

1. Defina a sintaxe abstracta das novas expressões e do novo tipo, e as regras de inferência de tipo para as novas expressões.

2. Indique as novas formas canónicas da linguagem e as novas regras de avaliação *call-by-value*.

3. Defina uma função que calcula o comprimento de uma sequência.