

Semântica das Linguagens de Programação

Exame (19 de Junho de 2019) / Duração: 2:30

Questão 1 Sejam $C, P \in \mathbf{Stm}$ os seguintes programas:

$$C \equiv \begin{array}{l} \text{if } y \leq 10 \\ \text{then } \{ x := x + y ; y := 2 * x \} \\ \text{else } \{ x := 7 \} \end{array} \quad \Bigg| \quad P \equiv \{ x := 1 ; y := a + y \} ; C$$

1. Recorrendo à semântica de transições (*small-step*) simule a execução do programa P a partir do estado inicial s , em que $sa = 3$ e $sy = 5$. Apresente as árvores de prova para as 2 primeiras transições apenas.
2. Sendo s' o estado em que $s'a = 8$ e $s'y = 4$, apresente a árvore de derivação do juízo de avaliação *big-step*: $\langle P, s' \rangle \rightarrow s'[x \mapsto 7, y \mapsto 12]$.
3. Apresente a interpretação denotacional do programa P .

Questão 2 Pretende-se estender a linguagem WHILE com uma nova forma de ciclo de acordo com a seguinte sintaxe abstracta: $\mathbf{Stm} \ni C ::= \dots \mid \mathbf{do } C \mathbf{ while } b$

A descrição informal da semântica deste comando é a seguinte: *O comando C é executado repetidamente enquanto o valor da expressão b for verdade, sendo o teste feito depois da execução do comando.*

1. Especifique formalmente o comportamento deste novo ciclo, escrevendo regras apropriadas (que não devem fazer referência a outros ciclos) para:
 - (a) a semântica natural.
 - (b) a semântica operacional estrutural.
2. Proponha uma regra da lógica de Hoare para os ciclos **do C while b** (sem recorrer a outras formas de ciclo) e prove a sua correcção tendo em conta a equivalência semântica entre os comandos $\{\mathbf{do } C \mathbf{ while } b\}$ e $\{C ; \mathbf{while } b \mathbf{ do } C\}$.
3. Estenda a geração de código da máquina a **AM** de forma a lidar com este novo comando.

Questão 3 Sejam $b_1, b_2 \in \mathbf{Bexp}$ tal que $b_1 \Rightarrow \neg b_2$, e $A, B, C, P_1, P_2 \in \mathbf{Stm}$ tal que

$$\begin{array}{l} P_1 \equiv \mathbf{while } b_1 \mathbf{ do } \{\{\mathbf{if } b_2 \mathbf{ then } A \mathbf{ else } B\} ; C\} \\ P_2 \equiv \mathbf{while } b_1 \mathbf{ do } \{B ; C\} \end{array}$$

Com base na semântica natural (big-step), indique o significado formal da afirmação:

“ P_1 e P_2 são programas semanticamente equivalentes”

e demonstre uma das implicações (à sua escolha) envolvidas na prova dessa equivalência.

Questão 4 Considere o seguinte termo do lambda calculus puro

$$(\lambda a. \lambda b. \lambda c. (a\ c)\ (a\ b)\ c)\ (\lambda x. \lambda y. y)\ ((\lambda u. u)\ ((\lambda x. x\ x)(\lambda x. x\ x)))\ (\lambda a. \lambda b. a)$$

1. Apresente a sequência correspondente à ordem normal de avaliação deste termo, até à sua forma normal. Sublinhe o β -redex que é selecionado em cada passo de redução.
2. Como classifica este termo quanto normalização? Justifique a sua resposta.

Questão 5 Considere a seguinte expressão

$$\begin{aligned} \text{let } g &\equiv \lambda x. \lambda y. \langle x + y, x - y \rangle \\ &\text{in listcase } ((3 * 3) :: 4 :: \text{nil}) \text{ of } (g\ 0\ 0, \lambda h. \lambda t. g\ h\ (g\ 2\ 1).2) \end{aligned}$$

1. Apresente a sequência da avaliação call by value desta expressão, passo a passo, até à sua forma canónica.
2. Construa uma árvore de prova do seguinte juízo

$$g : \text{Int} \rightarrow \text{Int} \rightarrow \text{Int} \times \text{Int}, l : \text{List Int}, x : \text{Int} \vdash \text{listcase } l \text{ of } (\langle 0, 0 \rangle, \lambda h. \lambda t. g\ h\ x) : \text{Int} \times \text{Int}$$

Questão 6 Pretende-se estender a linguagem de programação funcional, com um novo tipo de dados para representar o equivalente ao seguinte tipo de dados do Haskell:

```
data Arv a b = Vazia | Folha a | Nodo b (Arv a b) (Arv a b)
```

Ou seja, árvores binárias com informação (de tipo b) nos nodos intermédios e com folhas que tanto podem ter informação (de tipo a) como ser vazias.

1. Estenda a sintaxe, as regras de tipo, e as regras de avaliação call by name, para incorporar na linguagem este tipo de dados.
2. Defina uma função **soma**, de tipo $(\text{Arv Int } \theta) \rightarrow \text{Int}$, que calcula a soma das folhas de uma árvore.