

Semânticas de Linguagens de Programação - CC3033

Exercícios 8

Linguagem funcional SFUN

1. Escreve um programa em **SFUN** que define funções para calcular o mínimo e o máximo valor entre um par de inteiros.
2. Escreve um programa em **SFUN** que define uma função para calcular n -ésimo número de Fibonacci.
3. Mostra que o programa **SFUN**

$$square(x) = x \times x$$

$$double(x) = 2 \times x$$

e o termo fechado $square(double(3))$ admitem tipo no ambiente

$$\varepsilon(square) = \text{int} \rightarrow \text{int}$$

$$\varepsilon(double) = \text{int} \rightarrow \text{int}$$

Indica qual o contexto (ambiente de variáveis) que é necessário para o programa.

4. Mostra que o programa **SFUN**

$$g(x) = \text{if } x = 0 \text{ then } 0 \text{ else if } x = 1 \text{ then } 1 \text{ else } g(x - 1) + g(x - 2)$$

e o termo fechado $g(3)$ admitem tipo no ambiente $\varepsilon(g) = \text{int} \rightarrow \text{int}$. Indica qual o contexto (ambiente de variáveis) que é necessário para o programa.

5. Prova a unicidade de tipo, i.e. Para qualquer termo t de **SFUN**, contexto Γ e ambiente de símbolos funcionais ε , se $\Gamma \vdash_{\varepsilon} t : \tau$ e $\Gamma \vdash_{\varepsilon} t : \sigma$ então $\tau = \sigma$.
6. Para o factorial definido como

$$fact(x) = \text{if } x = 0 \text{ then } 1 \text{ else } x \times fact(x - 1)$$

avalia $fact(0)$ para a semântica $\Downarrow_P^v$.

7. Determina a avaliação do termo $square(double(3))$ com respeito ao programa

$$square(x) = x \times x$$

$$double(x) = 2 \times x$$

(a) para a semântica operacional com passagem de argumentos por valor

(b) para a semântica operacional com passagem de argumentos por nome

8. Dá exemplo de um programa em **SFUN** e de um termo fechado cuja avaliação por nome e por valor seja diferente.
9. (Lema da Substituição) Mostra que se $x_1 : \sigma_1, \dots, x_n : \sigma_n \vdash_{\varepsilon} t : \tau$ e $t_1, \dots, t_n$ são termos fechados tal que $\vdash_{\varepsilon} t_i : \sigma_i$ para $1 \leq i \leq n$, então $\vdash_{\varepsilon} t[x_i \mapsto t_i] : \tau$. Nota: começa por definir formalmente a substituição $t[x \mapsto t']$ por indução no termo t .
10. (Preservação do Tipo para $\Downarrow_P^v$) Mostra que se t é um termo fechado com tipo σ e $t \Downarrow_P^v v$ então v tem tipo σ .

11. (Determinismo e Tipificação forte para $\Downarrow_P^n$) Mostrar que a avaliação de um termo t fechado e que admite tipo para um programa P não pode produzir erros. E se $t \Downarrow_P^n v_1$ e $t \Downarrow_P^n v_2$ então $v_1 = v_2$.
12. (Preservação do Tipo para $\Downarrow_P^n$) Mostra que se t é um termo fechado com tipo σ e $t \Downarrow_P^n v$ então v tem tipo σ .
13. Determina a avaliação do termo $g(3)$ com respeito ao programa

$$g(x) = \text{if } x = 0 \text{ then } 0 \text{ else if } x = 1 \text{ then } 1 \text{ else } g(x - 1) + g(x - 2)$$

- (a) para a semântica operacional com passagem de argumentos por valor
- (b) para a semântica operacional com passagem de argumentos por nome

Regras do sistema de tipos para SFUN

$$\begin{array}{c}
\frac{}{\Gamma \vdash_{\varepsilon} b : \text{bool}} \text{ (b)} \\
\frac{}{\Gamma \vdash_{\varepsilon} n : \text{int}} \text{ (n)} \\
\frac{\Gamma(x) = \sigma}{\Gamma \vdash_{\varepsilon} x : \sigma} \text{ (var)} \\
\frac{\Gamma \vdash_{\varepsilon} t_1 : \text{int} \quad \Gamma \vdash_{\varepsilon} t_2 : \text{int}}{\Gamma \vdash_{\varepsilon} t_1 \text{ op } t_2 : \text{int}} \text{ (op)} \\
\frac{\Gamma \vdash_{\varepsilon} t_1 : \text{int} \quad \Gamma \vdash_{\varepsilon} t_2 : \text{int}}{\Gamma \vdash_{\varepsilon} t_1 \text{ bop } t_2 : \text{bool}} \text{ (bop)} \\
\frac{\Gamma \vdash_{\varepsilon} t_1 : \text{bool} \quad \Gamma \vdash_{\varepsilon} t_2 : \text{bool}}{\Gamma \vdash_{\varepsilon} t_1 \wedge t_2 : \text{bool}} \text{ (and)} \\
\frac{\Gamma \vdash_{\varepsilon} t : \text{bool}}{\Gamma \vdash_{\varepsilon} \neg t : \text{bool}} \text{ (not)} \\
\frac{\Gamma \vdash_{\varepsilon} t_0 : \text{bool} \quad \Gamma \vdash_{\varepsilon} t_1 : \sigma \quad \Gamma \vdash_{\varepsilon} t_2 : \sigma}{\Gamma \vdash_{\varepsilon} \text{if } t_0 \text{ then } t_1 \text{ else } t_2 : \sigma} \text{ (if)} \\
\frac{\Gamma \vdash_{\varepsilon} t_1 : \sigma_1 \dots \quad \Gamma \vdash_{\varepsilon} t_{a_i} : \sigma_{a_i} \quad \varepsilon(f_i) = (\sigma_1, \dots, \sigma_{a_i}) \rightarrow \sigma}{\Gamma \vdash_{\varepsilon} f_i(t_1, \dots, t_{a_i}) : \sigma} \text{ (fn)}
\end{array}$$

Semânticas com passagem de argumentos por valor e por nome para SFUN dado um programa P , $f_i(x_1, \dots, x_{a_i}) = d_i$, $1 \leq i \leq k$.

$$\begin{array}{c}
\frac{}{n \Downarrow_P^v n} \text{ (n)} \\
\frac{}{b \Downarrow_P^v b} \text{ (b)} \\
\frac{t_1 \Downarrow_P^v n_1 \quad t_2 \Downarrow_P^v n_2}{t_1 \text{ op } t_2 \Downarrow_P^v n_1 \text{ op } n_2} \text{ (op)} \\
\frac{t_1 \Downarrow_P^v n_1 \quad t_2 \Downarrow_P^v n_2}{t_1 \text{ bop } t_2 \Downarrow_P^v n_1 \text{ bop } n_2} \text{ (bop)} \\
\frac{t_1 \Downarrow_P^v b_1 \quad t_2 \Downarrow_P^v b_2}{t_1 \wedge t_2 \Downarrow_P^v b_1 \wedge b_2} \text{ (and)} \\
\frac{t \Downarrow_P^v b}{\neg t \Downarrow_P^v \neg b} \text{ (not)} \\
\frac{t_0 \Downarrow_P^v \text{true} \quad t_1 \Downarrow_P^v v_1}{\text{if } t_0 \text{ then } t_1 \text{ else } t_2 \Downarrow_P^v v_1} \text{ (ifT)} \\
\frac{t_0 \Downarrow_P^v \text{false} \quad t_2 \Downarrow_P^v v_2}{\text{if } t_0 \text{ then } t_1 \text{ else } t_2 \Downarrow_P^v v_2} \text{ (ifF)} \\
\frac{t_1 \Downarrow_P^v v_1 \dots t_{a_i} \Downarrow_P^v v_{a_i} \quad d_i[x_1 \mapsto v_1, \dots, x_{a_i} \mapsto v_{a_i}] \Downarrow_P^v v}{f_i(t_1, \dots, t_{a_i}) \Downarrow_P^v v} \text{ (fn)} \\
\frac{d_i[x_1 \mapsto t_1, \dots, x_{a_i} \mapsto t_{a_i}] \Downarrow_P^n v}{f_i(t_1, \dots, t_{a_i}) \Downarrow_P^n v} \text{ (fn}_N\text{)}
\end{array}$$