Aula 12 - Lógica de Programação

5.3) Expressividade

Naturais

Def.: 1) $M, N \in \Lambda$; $n \in \mathbb{N}_0$;

a notação $M^n(N)$ é definida recursivamente por:

a) $M^0(N) = N$

b) $M^{k+1}(N) = M\ M^k(N)$

2) Dado $n \in \mathbb{N}_0$, o n-ésimo numeral (de Church) é notado por c_k e dado por:

$$c_n = \lambda f x . f^k(x)$$

Exemplo: $c_0 = \lambda f x . x$;

$c_1 = \lambda f x . f x$; $c_2 = \lambda f x . f(f x)$

Def.: $C^+ =_{def} \lambda x y z w . x z (y z w)$ (combinador soma)

$C^\times =_{def} \lambda x y z . x (y z)$ (combinador produto)

Prop.: $\forall m, n \in \mathbb{N}_0$

1) $C^+ c_m\, c_n =_\beta c_{(m+n)}$

2) $C^\times c_m\, c_n =_\beta c_{(m \times n)}$

Dem.: (Não copiei)

1) Por indução em n $(\in \mathbb{N}_0)$.

Caso $n = 0$

$$\text{L.E.} = C^+ c_m\, c_0$$

$$=_\beta \lambda z w . c_m z (c_0\, z\, w)$$

$c_0 = \lambda f x . x$

$$=_\beta \lambda z w . c_m\, z\, w$$

$c_m = \lambda f x . f^m(x)$

$$=_\beta \lambda z w . z^m(w) =_\beta c_m =$$

$= c_{(m+0)} = L.D.$

Lema: $M^n(N)\left[{}^{M_0}/_M\right] =$

$= \left(M\left[{}^{M_0}/_x\right]\right)^n \left(N\left[{}^{M_0}/_x\right]\right)$

Caso $n = k+1$ (Exercício)

2) Lema: $\forall m, n \in \mathbb{N}_0 \quad (c_n\, z)^m (x)$

Dem.: Por indução em m $(\in \mathbb{N}_0)$.

$$L.E. =_\beta \lambda z. \underset{\lambda f x. f^m(x)}{c_m} (c_n\, z)$$

$$=_\beta \lambda z. \lambda x\, (c_n\, z)^m (x)$$

Lema $=_\beta \lambda z. \lambda x. z^{n\times m} (x)$

$= c_{n\times m} = L.D.$

Booleanos, condicionais e pares:

Def.: 1) $true =_{def} \lambda xy. x \;(= K)$

2) $false =_{def} \lambda xy. y \;(= W)$

3) if B then M else $N =_{def} (BM)N$

$(B, M, N \in \Lambda)$

Prop: 1) $B =_\beta true \Rightarrow$ if B then M else $N =_\beta M$

2) $B =_\beta false \Rightarrow$ if B then M else $N =_\beta N$

Dem.: 1) $L.E = BMN =_\beta \underset{\lambda xy. x}{true}\, MN =_\beta M = L.D.$

2) Exerc.

Prop.: 1) $[M, N]\; true =_\beta M$ (1ª projeção)

2) $[M, N]\; false =_\beta N$ (2ª projeção)

Dem.: 1) $L.E. = (\lambda z. zMN)(\lambda xy. x) =_\beta$

$=_\beta (\lambda xy. x) MN =_\beta M = L.D.$

2) (Exercício)

Def: Uma função numérica $f: \mathbb{N}_0^n \to \mathbb{N}_0$ $(n \in \mathbb{N}_0)$ diz-se **λ-definível** quando para algum combinador F:

$$\forall_{k_1,\dots,k_n \in \mathbb{N}_0} \quad \underset{\lambda\text{-termo}}{F}\, c_{k_1} \dots c_{k_n} =_\beta c_{\underbrace{F(k_1,\dots,k_n)}_{\in \mathbb{N}_0}}$$

numerais de Church — numeral de Church — $\in \mathbb{N}_0$

dizendo-se que **F λ-define f.**

$$\underset{F}{c^+}\, c_m c_n =_\beta \underset{f}{c_{m+n}}$$

Exemplos: 1) A soma e o produto em $\mathbb{N}_0$ são λ-definidos por c^+ e $c^\times$, respetivamente.

$succ =_\beta c_{n+1}$

2) $suc =_{def} \lambda n f x.\, f(n f x)$ λ-define a função sucessor

$$\left(\begin{array}{l} S: \mathbb{N}_0 \longrightarrow \mathbb{N}_0 \\ \quad n \longmapsto n+1 \end{array} \right)$$

Def: A classe R das funções numéricas recursivas é a menor classe tal que:

1) R contém as funções numéricas iniciais:

a) $\underset{\substack{\uparrow \\ i\text{-ésima projeção} \\ \text{(de entre } p \text{ argumentos)}}}{U_i^p}(n_1,\dots,n_p) = n_i \quad (1 \leq i \leq p)$, $p \in \mathbb{N}$

b) $S(n) = n+1$ — Sucessor

c) $Z(n) = 0$ — Função constante igual a 0

2) R é fechada para a composição de funções, isto é:

$f, g_1, \dots, g_n \in R \Rightarrow h \in R$

para h tal que

$$h(\underbrace{n_1,\dots,n_k}_{\vec{n}}) = f(g_1(\vec{n}),\dots,g_n(\vec{n}))$$

3) R é fechado para minimização, isto é:

$$f\in R \wedge \forall_{\vec{n}}\exists_m\ f(\vec{n},m)=0 \Rightarrow g\in R$$

para g tal que

$$g(\vec{n}) = \mu m.[f(\vec{n},m)=0] =$$

operador de minimização

$$= \text{mínimo}\{m\in\mathbb{N}_0 : f(\vec{n},m)=0\}$$

Teor.: (Kleene, 36) As funções numéricas λ-definíveis são exatamente as funções recursivas.

Tese de Church-Turing .....

Obs.: O operador de minimização é codificado em λ-calculus com recurso aos chamados combinadores de ponto-fixo.

Def.: M é um combinador de ponto fixo quando:

$$\forall F\in\Lambda.\ MF =_\beta F(MF)$$

Prop.: $Y = \lambda f.(\lambda x. \underset{F}{f}(xx))(\lambda x. \underset{F}{f}(xx))$ é um combinador de ponto fixo.

$x \in x$
xx

Dem.: Dado $F\in\Lambda$. Seja

$$A_F = \lambda x. F(xx)$$

Então:

$$YF =_\beta A_F A_F =_\beta F(A_F A_F) =_\beta F(YF)$$

Símbolo de conclusão

$\therefore$ Y é combinador de ponto fixo.

(Termina aqui a matéria para o teste,