Recursão em Cálculo Lambda

Assuma que temos definições em cálculo λ para `if`, `mult`, `isZero`, `pred`, `succ`, números naturais e valores booleanos.

Queremos permitir definições recursivas de funções, como por exemplo:

`Fac =  $\lambda n.$ if (isZero  $n$ ) then 1 else (mult  $n$  (Fac (pred  $n$ )))`

Se definirmos

`define  $F =$`

`$\lambda f.$  $\lambda n.$ if (isZero  $n$ ) then 1 else (mult  $n$  ( $f$  (pred  $n$ )))`

então precisamos encontrar uma expressão λ `Fac` tal que

$$(F \text{ Fac}) = \text{Fac}$$

Ponto Fixo

Definição: Seja $f : A \rightarrow A$ uma função. Um elemento $x \in A$ é chamado **ponto fixo** de f se $f(x) = x$.

Exemplos:

Função	Pontos fixos
$g(n) = n^2 - 6n$	0 e 7
$g(n) = n$	todo $n \in \mathbb{N}$
$g(n) = n + 5$	nenhum

Qualquer função recursiva $f : A \rightarrow B$ pode ser escrita na forma:

$$f = F f$$

Se garantirmos que esse tipo de equação sempre tem solução em cálculo λ , será possível descrever todas essas funções como expressões λ .

Operador de Ponto Fixo

Definição: Um **operador de ponto fixo** é uma expressão λH tal que para qualquer expressão λf ,

$$H f = f(H f)$$

Ou seja, um operador de ponto fixo é uma função que **gera um ponto fixo** para uma função!

Considere a expressão λ

$$Y = \lambda f.(\lambda x.f(x x))(\lambda x.f(x x))$$

Esta expressão é um operador de ponto fixo, pois (assuma que não ocorre livre em g):

$$\begin{aligned} Y g &\Rightarrow_{\beta} (\lambda x.g(x x))(\lambda x.g(x x)) \\ &\Rightarrow_{\beta} g((\lambda x.g(x x))(\lambda x.g(x x))) \\ &= g(Y g) \end{aligned}$$

Ou seja, $Y g$ é um ponto fixo para g .

Exemplo: Fatorial

$$F \text{ Fac} = \text{Fac}$$

onde

$$F = \lambda f. \lambda n. \text{if (isZero } n) \text{ then 1 else (mult } n \text{ (} f \text{ (pred } n)) \text{))}$$

Podemos definir **Fac** como o ponto fixo de F ,
ou seja,

define $\text{Fac} = Y \ F$ ou

define $\text{Fac} = Y (\lambda f. \lambda n. \text{if (isZero } n) \text{ then 1 else (mult } n \text{ (} f \text{ (pred } n)) \text{))}$

Exemplo: Reduzir (Fac 1)...

λ -Computabilidade

Definição: Uma função (parcial) $f : \mathbb{N} \rightarrow \mathbb{N}$ é **λ -computável** se existe uma expressão λ E tal que, para qualquer $n, k \in \mathbb{N}$

$$(E \ \bar{n}) \Rightarrow^* \bar{k} \text{ sss } f(n) = k$$

onde $\bar{n}$ e $\bar{k}$ são as representações em cálculo λ dos números n e k , respectivamente.

Observações:

- $(E \ \bar{n}) = \overline{f(n)}$ sss $n \in \text{dom}(f)$;
- $(E \ \bar{n})$ não tem forma normal sss $n \notin \text{dom}(f)$.

Teorema: Uma função (parcial) é Turing-computável se e somente se ela é λ -computável.