aplica n vezes f em x.

7) succ = $\lambda n.\lambda f.\lambda x.\ f(n f x)$

pred = $\lambda n.\lambda f.\lambda x.$ extract (n inc $\underbrace{\text{const}}$)

const : não aplica f ao x

↳ inc const = value ((n-1) f x)

↳ extract (value v) = v

inc (value v) = value (f v)

plus = $\lambda m.\lambda n.\lambda f.\lambda x.\ m f (n f x)$

sub = $\lambda m.\lambda n\ (n\ \text{pred})\ m$

-n

def: o valor da codificação de Church representa quantas vezes uma função ordem maior for composta.

→ succ (0) = $(\lambda n.\lambda f.\lambda x.\ f(n f x))0 = \lambda f.\lambda x.\ f(0 f x) = \lambda f.\lambda x\ f x = 1$

→ succ (1) = $(\lambda n.\lambda f.\lambda x.\ f(n f x))1 = \lambda f.\lambda x\ f(f x) = \lambda f.\lambda x f^{\circ 2} x = 2$

→ succ (2) = $(\lambda n.\lambda f.\lambda x.\ f(f(f(x)))) = \lambda f \lambda x\ f^{\circ 3} x = 3$

→ plus(0,1) = $(\lambda m.\lambda n.\lambda f\ \lambda x\ m f (n f x))0)1 = \lambda f.\lambda x\ 0 f (1 f x) = 1$

→ plus(1,2) = $(\lambda m.\lambda n.\lambda f.\lambda x\ m f (n f x))1)2 = \lambda f.\lambda x\ 1 f (2 f x)$

$= \lambda f.\lambda x\ f^{\circ 3} x$

$1 = \lambda f.\lambda x\ f x$

$2 = \lambda f.\lambda x\ f^{\circ 2} x$

$3 = \lambda f.\lambda x.\ f^{\circ 3} x$

$4 = \lambda f.\lambda x.\ f^{\circ 4} x$

$5 = \lambda f.\lambda x.\ f^{\circ 5} x$

$6 = \lambda f.\lambda x.\ f^{\circ 6} x$

$7 = \lambda f.\lambda x.\ f^{\circ 7} x$

$8 = \lambda f.\lambda x.\ f^{\circ 8} x$

$9 = \lambda f.\lambda x.\ f^{\circ 9} x$

$10 = \lambda f.\lambda x.\ f^{\circ 10} x$

m-n

→ sub(1,0) = $(\lambda m.\lambda n\ (n\ \text{pred})\ \cancel{m})\ 1)\ 0$

= (0 pred) 1 = 1

↳ pred (x) = $\begin{cases} 0, & x == 0 \\ (x-1), & x > 0 \end{cases}$

→ sub(2,1) = (1 pred) 2 = pred(2) = 1

argumento de pred

8) O combinador Y é uma forma de criar uma recursão, através do cálculo-λ (com o combinador ômega: $(\lambda x.xx)(\lambda x.xx)$), quando a linguagem não tem suporte para isso.

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

aplicando uma função g qualquer:

$$Yg = \lambda x. g(xx)(\lambda x. g(xx))$$

$$= g\ (\lambda x. g(xx))(\lambda x. g(xx)).$$

$$= g\ Yg.$$