Tarea 4 García Ponce José Camilo

función factorial

$0! = 1$

$(n+1)! = n! \times (n+1)$

1) Definela en el formato de las funciones recursivas $-\mu$

usaremos estas funciones (fueron vistas en la ayudantía 16/5/23)

$z(n) = 0$ (cero) / $s(n) = n+1$ (sucesor)

(multiplicación)
$mult(0,y) = z() = 0$
$mult(x+1,y) = suma(y, mult(x,y))$

(suma)
$suma(0,x) = \pi_1^1(x) = x$
$suma(x+1,y) = s(\pi_3^3(x,y,suma(x,y)))$

entonces factorial queda

$fact(0) \equiv_{def} s(z(0)) = 1$

$fact(n+1) \equiv_{def} mult(fact(n), s(n))$

2) Ahora defínelo como término Λ con base en tu respuesta 1

las funciones que usaremos son (vistas en la clase 17/5/23)

$Cero \equiv_{def} \lambda x. x V$ (dice si es $\emptyset$), $S \equiv_{def} \lambda x. [F, x]$ (sucesor), $P \equiv_{def} \lambda x. x F$ (predecesor)

$Mult \equiv_{def} Y \lambda f. \lambda y x. (Cero\ y)(C\ y)(suma\ (mult\ (P y)\ x)\ x)$ (multiplicación)

$C \equiv_{def} \lambda x. \overline{\emptyset}$ (cero), $suma \equiv_{def} Y \lambda f. \lambda y x. (Cero\ y)(H x)(G (P_y) x (f (P_y) x))$ (suma)

$H \equiv_{def} \lambda x. x$ $\quad G \equiv_{def} \lambda z, x, y. S((\lambda x, y, \omega. \omega) z x y)$

entonces factorial será

$fact \equiv_{def} Y \lambda f. \lambda x. (Cero\ x)(S x)(mult\ (fact\ (P_x))\ x)$

3) A partir de la funcion recursiva-μ, definela con un programa de IMP

veamos quienes seran los Programas P_h y P_g

tal que

$$\langle P_h, \sigma \rangle \to \sigma' \qquad y \qquad \langle P_g, \sigma_1 \rangle \to \sigma_1'$$

$\sigma'(X_0) = \text{suceso}(\sigma(X_3))$ donde X_3 es una localidad con 0 almacenado

$\sigma_1'(X_0) = \text{mult}(\text{fact}(\sigma(X_2)), \text{suceso}(\sigma(Z)))$

entonces

$$P_h = X_0 := X_3 + 1$$

$$P_g = X_0 := (X_2 \times (Z+1))$$

y queda

```
X3 := 0;  W := Z;

Y0 := 0;

Ph;

while Y0 < W do
   Z := Y0;
   X2 := X0;
   Pg;
   Y0 := Y0 + 1;
```

de esta manera se calcula $\text{fact}(Z)$ y se guarda en X_0

4) Define la minimalización no acotada por medio de programas while.

sea $g: \mathbb{N}^{n+1} \to \mathbb{N}$ función computable, y supongamos que g' es un programa que lo calcula, ahora realizaremos un programa f para la minimalización no acotada, donde el resultado estará en Y

$Z := 0$, $Y_1 := 0$

$W_1 := X_1, \ldots, W_n := X_n$

$Y := 0$, $X_2 := 1$

while $\neg(Z = 0)$ do

$Y_1 := Y$

$X_1 := W_1, \ldots, X_n := W_n$

$X_2 := g(Y, X_1, \ldots, X_n)$

if $(X_2 = 0)$ then

$Z := 1$

else

$Y := Y_1 + 1$