Fecha de entrega: 26 de septiembre.

(*Si redactas por completo tu tarea en LaTeX tienes un punto extra*)

1. Los naturales de Church se definen como sigue:

$$\overline{0} = \lambda s.\lambda z.z$$
$$\overline{1} = \lambda s.\lambda z.s\ z$$
$$\overline{2} = \lambda s.\lambda z.s\ (s\ z)$$
$$\overline{3} = \lambda s.\lambda z.s\ (s\ (s\ z))$$
$$\vdots$$

Se define el par ordenado $\underline{pair} := \lambda x.\lambda y.\lambda p.p\ x\ y$, así, el par ordenado $(a, b) = \underline{pair}\ a\ b = \lambda p.p\ a\ b$. Las funciones para obtener la primer y segunda componente de un par ordenado se definen respectivamente como $\underline{fst} := \lambda p.p\ \underline{true}$ y $\underline{snd} := \lambda p.p\ \underline{false}$.
Sean g_1 y h_1 las siguientes funciones:

$$g_1 := \lambda n.\lambda s.\lambda z.n\ (\lambda h_1.\lambda h_2.h_2\ (h_1\ s))\ (\lambda u.z)\ (\lambda u.u)$$

$$h_1 := \lambda n.\underline{fst}\ (n\ \underline{ss}\ \underline{zz}),\ \text{donde}\ \underline{ss} = \lambda p.\underline{pair}\ (\underline{snd}\ p)\ (\underline{suc}\ (\underline{snd}\ p)),\ \text{y}\ \underline{zz} := \underline{pair}\ \overline{0}\ \overline{0}$$

   a) Calcula $(g_1\ \overline{0})$ y $(g_1\ \overline{3})$
   b) Calcula $(h_1\ \overline{1})$ y $(h_1\ \overline{2})$
   c) ¿Qué hacen las funciones g_1 y h_1?

2. Los naturales de Scott se definen como sigue:

$$\hat{0} = \lambda x.\lambda y.x$$
$$\hat{1} = \lambda x.\lambda y.y\ \hat{0}$$
$$\hat{2} = \lambda x.\lambda y.y\ \hat{1}$$
$$\hat{3} = \lambda x.\lambda y.y\ \hat{2}$$
$$\vdots$$

Sean f_2, g_2 y h_2 las siguientes funciones:

$$f_2 := \lambda n.\lambda x.\lambda y.yn$$
$$g_2 := \lambda n.n\ \hat{0}\ (\lambda x.x)$$
$$h_2 := \lambda n.n\ \underline{true}\ (\lambda x.\underline{false})$$

   a) Calcula $(f_2\ \hat{0})$ y $(f_2\ \hat{3})$
   b) Calcula $(g_2\ \hat{1})$ y $(g_2\ \hat{4})$
   c) Calcula Calcula $(h_2\ \hat{0})$ y $(h_2\ \hat{5})$
   d) ¿Qué hacen las funciones f_2, g_2 y h_2?

    e) (Extra [+1 punto]) Haz una función que haga la suma de naturales de Scott.

3. Sean R, Q y H los siguientes términos del cálculo-λ:
   - $R = \lambda x.\lambda y.(xy)x$
   - $Q = \lambda y.\lambda x.y((xy)x)$
   - $H = RQ$

   Muestra que H es un combinador de punto fijo.
4. Utilizando un combinador de punto fijo, implementa estas funciones de forma recursiva:

   a) Una función que dados n y m calcule n^m.

   b) Una función que decida si un natural de Church es impar.
5. Da una definición inductiva mediante juicios de palabras palíndromas sobre el alfabeto $\{a, b\}$.
   - Enuncia el principio de inducción para los juicios que definiste.
   - Demuestra por inducción matemática que si w es una cadena palíndroma entonces *reverse*$(w) = w$.
6. Utilizando inducción matemática demuestra lo siguiente:
   - Si $\langle k, s\rangle$ *pila*, entonces $\underbrace{(\ldots(}_{k}\, s\; M$.
   - Si $\langle k, s\rangle$ *pila*, entonces $\langle k, (s)\rangle$ *pila*.
7. Extiende el lenguaje EAB con un operador *even* que tome un natural y decida si dicho número es par de la siguiente manera.
   - Extiende la sintaxis concreta.
   - Extiende la sintaxis abstracta.
   - Extiende la semántica estática.
   - Extiende la semántica dinámica.
8. Sean e_1 y e_2 las siguientes expresiones:

   $$e_1 = \texttt{let } y \texttt{ = suc(2+1) in (let } z \texttt{ = pred(5) in } z\texttt{+}y\texttt{ end) } \star \texttt{ 2+}y\texttt{ end}$$
   $$e_2 = \texttt{(let } y \texttt{ = } x\texttt{+}v \texttt{ in (let } z \texttt{ = } x \texttt{ in } x\star y\star z \texttt{ end) end)[}x\texttt{:=}y\star z\texttt{]}$$

   - Convierte las expresiones e_1 y e_2 a sus respectivas representaciones como *asas*, t_1 y t_2, respectivamente.
   - Con los juicios para la semántica estática haz derivaciones para t_1 y t_2.
   - Evalúa t_1 y t_2 usando los juicios para la semántica dinámica.
9. Haz una derivación para la siguiente expresión y el siguiente contexto:

   $$\{x : \mathsf{T} \to \mathsf{R} \to \mathsf{S}\} \vdash \lambda y : \mathsf{R}.\lambda z : \mathsf{T}.xyz : \mathsf{R} \to \mathsf{T} \to \mathsf{S}$$
10. Para cada una de las siguientes expresiones haz una inferencia de tipos utilizando el algoritmo W.
    - $\lambda x.(\lambda x.x)$
    - $\lambda x.\lambda y.\lambda z.y(xz)$
    - $\lambda g.\lambda y.g\ (y + z)$
    - $\texttt{let } h \texttt{ = } \lambda z.z\texttt{+1 in (}\lambda x.h\ x\texttt{) 4 end}$
11. Considera el tipo:

    ```
    data Figura = Circ Float | Rect Float Float | Tri Float Float
    ```

para representar figuras geométricas con sus respectivos lados (el triángulo es un triángulo rectángulo).

- Define el tipo *Figura* y sus constructores usando tipos suma.
- Define el tipo *Figura* y sus constructores usando tipos variante.
- Implementa una función *per* que calcule el perímetro de una figura definida con tipos suma.
- Implementa una función *per* que calcule el perímetro de una figura definida con tipos variante.