

Objetivos: Relacionar los conceptos vistos en el teórico con la implementación en Haskell de esos conceptos. Repasar Haskell.

Sea τ un tipo (de los de Lógica) con $C = \{c, d\}$, $F = \{f, g\}$ cuyas aridades son $a(f) = 1$ y $a(g) = 2$. Si aceptamos que el conjunto $\langle \text{var} \rangle$ es \mathbb{Z} , entonces podemos representar en Haskell los árboles de los términos de tipo τ con el siguiente tipo de datos:

```
data Termino = Var Int | C | D | F Termino Termino | G Termino Termino Termino
```

Para llevar árboles de términos a strings definimos la función que sigue:

```
concr :: Termino -> String
concr (Var n) = 'X' : show n
concr C = "c"
concr D = "d"
concr (F t) = concat ["f(", concr t, ")"]
concr (G t1 t2) = concat ["g(", concr t1, ",", concr t2, ")"]
```

- (1) Ejecute `concr` para los siguientes árboles


```
t1 = Var 2
t2 = G (Var 0) (F C D)
t3 = F t2
```
- (2) ¿Es la función `concr` inyectiva? ¿Es suryectiva?
- (3) Explique por qué no es suryectiva. ¿Qué hace la función `abstr` que va de la imagen de `concr` a `Termino`?
- (4) Escriba una función `cerrado? :: Termino -> Bool` tal que `cerrado? t = True` si y sólo si `t` es un término cerrado.
- (5) ¿Es una función semántica? ¿Es su definición dirigida por sintaxis?

Dado un conjunto como `Termino` podemos definir un esquema general para definir funciones de `Termino -> a`, para cualquier tipo `a`, estableciendo

- (a) cómo interpretar las variables (esto no es novedoso: los estados son eso);
 - (b) cómo interpretar cada constante: en nuestro caso esto es elegir `c :: a` y `d :: a`;
 - (c) cómo interpretar cada símbolo de función: qué debemos elegir en nuestro caso concreto?
- (8) Para `concr` y `cerrado?` explique quiénes son el “estado”, los elementos `c, d` y las interpretaciones de los símbolos de función.
 - (9) Defina una función `eval` que permita evaluar términos en cualquier otro tipo:


```
eval :: (a, a, a -> a, a -> a -> a) -> (Int -> a) -> Termino -> a
```

 Se lo invita a recordar la definición de la función de interpretación en “Lógica” para comprender el rol de cada argumento.
 - (10) Escriba definiciones para `concr` y `cerrado?` usando `eval`.