

**Objetivos:** Relacionar los conceptos vistos en el terico con la implementacin en Haskell de esos conceptos. Repasar Haskell.

Sea  $\tau$  un tipo (de los de Lgica) 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 rboles de los trminos de tipo  $\tau$  con el siguiente tipo de datos:

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

Para llevar rboles de trminos a strings definimos la funcin 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 rboles
 

```
t1 = Var 2
t2 = G (Var 0) (F C D)
t3 = F t2
```
- (2) Es la funcin `concr` inyectiva? Es suryectiva?
- (3) Explique por qu no es suryectiva. Qu hace la funcin `abstr` que va de la imagen de `concr` a `Termino`?
- (4) Escriba una funcin `cerrado? :: Termino -> Bool` tal que `cerrado? t = True` si y slo si `t` es un trmino cerrado.
- (5) Es una funcin semntica? Es su definicin 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) cmo interpretar las variables (esto no es novedoso: los estados son eso);
  - (b) cmo interpretar cada constante: en nuestro caso esto es elegir `c :: a` y `d :: a`;
  - (c) cmo interpretar cada smbolo de funcin: qu debemos elegir en nuestro caso concreto?
- (8) Para `concr` y `cerrado?` explique quines son el “estado”, los elementos `c, d` y las interpretaciones de los smbolos de funcin.
  - (9) Defina una funcin `eval` que permita evaluar trminos en cualquier otro tipo:
 

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

 Se lo invita a recordar la definicin de la funcin de interpretacin en “Lgica” para comprender el rol de cada argumento.
  - (10) Escriba definiciones para `concr` y `cerrado?` usando `eval`.