

Tarea 1 - Evaluador de Chi en Haskell

Teoría de la Computación
Universidad ORT Uruguay

Marzo 2022

El objetivo de esta tarea es codificar¹ en Haskell el *lenguaje* χ estudiado en el curso como modelo funcional de computabilidad. Ello incluye:

- Sintaxis abstracta.
- Reglas de evaluación.

tal como han sido descriptas en los repartidos publicados.

Se pide, concretamente:

1. Declarar un tipo inductivo (**data**) apropiado para representar las expresiones (sintaxis abstracta) de χ .
2. Definir el tipo de las **sustituciones**, así como el efecto de ellas sobre expresiones χ .
3. Definir la función (parcial²) de **evaluación**.
4. Codificar en χ embebido en Haskell las funciones:
 - **and**: la conjunción booleana.
 - **duplicar**: que dado un natural n , retorna el doble n .
 - **unir**: que dadas dos listas l_1 y l_2 , retorna l_1 seguido de l_2 (lo que conocemos en haskell como $(l_1 ++ l_2)$).
 - **ramaI**: un árbol binario, con información en los nodos, y hojas sin información, retorna una lista con todos los elementos de la rama izquierda.

Probar la función **unir** con una lista que contenga al cero y uno, y otra que tenga al dos y al tres.

¹Otro término técnico utilizado es *embeber*. En inglés se usan *to encode* y *to embed*.

²Cuando indicamos parcial, nos referimos a que puede fallar y no devolver un resultado.

Luego probar la función `ramaI` con un árbol que tenga al menos 3 niveles en la rama izquierda.

Adicionalmente se pueden aprovechar para explorar algunas condiciones raras, como el uso de variables no declaradas, listas de parámetros y argumentos que no tengan el mismo largo, incluso intentar ejecutar la evaluación de funciones sobre argumentos que no tendrían el “Tipo” esperado por la función. Ejemplos:

- `and True True False`
- `λx-> case y of {True -> False; False -> True}`, notar que la `x` y la `y` no coinciden, que ocurriría al aplicarle `True` o `False`.
- `unir (S(S 0)) True`, notar que la función `unir` recibe 2 listas, que ocurre si le paso un natural y un booleano, o incluso, una lista y un booleano, como en el ejemplo siguiente.
- `unir [1,3,0] False`