

# Tarea 1 - Evaluador de Chi en Haskell

Teoría de la Computación  
Universidad ORT Uruguay

Setiembre 2024

El objetivo de esta tarea es codificar<sup>1</sup> en Haskell el *lenguaje*  $\chi$  estudiado en el curso como modelo funcional de computabilidad. Ello incluye:

- sintaxis abstracta, y
- reglas de evaluación débil y evaluación completa,

tales como han sido descritas en la especificación del lenguaje publicada.

Se pide, concretamente:

1. Declarar un tipo inductivo (**data**) apropiado para representar las expresiones (sintaxis abstracta) de  $\chi$ .
2. Declarar un tipo inductivo (**data**) apropiado para representar a los valores y formas canónicas débiles de  $\chi$ .
3. Definir el tipo de las **sustituciones**, así como el efecto de ellas sobre expresiones  $\chi$  (para esto se requiere también definir las operaciones de búsqueda y bajas).
4. Definir la función (parcial<sup>2</sup>) de **evaluación débil**.
5. Definir la función de **evaluación completa** de expresiones.
6. Codificar en  $\chi$  embebido en Haskell las funciones:
  - **or**: la disyunción booleana.
  - **triple**: que dado un natural  $n$ , retorna el triple  $n$ .
  - **duplicar**: que dada una lista  $l$ , retorna  $l$  con todos sus elementos duplicados. Ejemplo. `duplicar [1,2,3] = [1,1,2,2,3,3]`

---

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

<sup>2</sup>Cuando indicamos *parcial*, nos referimos a que no actúa sobre valores y además falla en los casos así indicados en la especificación.

- **ramaC**: dado un árbol ternario, con información en los nodos, y las hojas, retorna una lista con todos los elementos de la rama central del árbol.