

Tarea 2 - Intérprete de Imp en Haskell

Teoría de la Computación
Universidad ORT Uruguay

Agosto 2024

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

- Sintaxis abstracta.
- Reglas de evaluación de expresiones.
- Reglas de ejecución de programas.

tal como han sido descritas en la especificación publicada.

Se pide, concretamente:

1. Definir tipos apropiados para representar los **programas**, **expresiones** y **valores** de `Imp`.
2. Definir el tipo de la **Memoria** y las funciones para operar sobre ella (búsqueda, actualización, alta y bajas).
3. Definir la función de evaluación de expresiones de `Imp`.
4. Definir la función (parcial) de ejecución de un programa de `Imp`.
5. Codificar en `Imp` embebido en Haskell los **programas**:
 - **par**: que determina si un natural dado es o no par.
 - **suma**: que calcula la suma de dos naturales.
 - **largo**: que calcula la cantidad de elementos de una lista dada.
 - **igualdadN**: que dados dos naturales `m` y `n`, determina si `m` es igual a `n`.
 - **concat**: que dadas dos listas `l1` y `l2`, retorna una nueva lista con los elementos de `l1` seguido de los elementos `l2`. (Lo que en Haskell sería la función `(++)`.)

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