

## Tarea 3 - Intérprete de MT en Haskell

Teoría de la Computación  
Universidad ORT Uruguay

Marzo 2022

El objetivo de esta tarea es embeber en Haskell las *MT*, estudiadas en el curso como modelo imperativo de computabilidad. Ello incluye:

- Representación de la memoria y el código.
- Reglas de transición de un estado a otro.
- Reglas de ejecución completa de programas.

tal como se han visto en clase y descriptas en los repartidos publicados.

Se pide, concretamente:

1. Definir un tipo (**type**) infinito, apropiado para representar los símbolos (**Symbol**) del lenguaje y declarar la constante **blank** que representará al símbolo reservado blanco (**#**), además un tipo para las cintas (**Tape**).
2. Definir el tipo (**type**) del código de los programas (**Code**), para ello debe definir previamente, un tipo (**type**) para los estados (**State**) y un tipo (**data**) para las acciones (**Action**). Declarar las constantes **init** y **halt**, como estados reservados del lenguaje.
3. Declarar un tipo (**type**) para las configuraciones (**Config**), las cuales son una pareja con un estado y una cinta. Nos servirá para saber que momento del código estamos ejecutando y como se encuentra la cinta en ese momento.
4. Definir la función parcial, **step :: Code -> Config -> Config**, que ejecuta la transición de un estado a otro en el programa.
5. Definir la función parcial **exec :: Code -> Tape -> Tape** que realiza la ejecución completa de un programa (Comenzando desde el estado **init**, hasta llegar a **halt**).

6. Codificar en *MT* embebido en Haskell los programas:

- **par**: que determina si un natural dado es o no par.

$$\sum = \{ I, T, F, \# \}$$

entrada :

$$\dots \# \underbrace{I I \dots I}_{n} \# \dots$$

salida :

$$\begin{cases} \dots \# I I \dots I \# T \# \dots & \text{Si } n = 2 \\ \dots \# I I \dots I \# F \# \dots & \text{Si } n \neq 2 \end{cases}$$

- **shift-right**: que mueve una secuencia de símbolos (considerar un lenguaje que contenga solo 3 símbolos  $\sigma_1$ ,  $\sigma_2$  y  $\sigma_3$ ) un lugar a la derecha.

$$\sum = \{ \sigma_1, \sigma_2, \sigma_3, \# \}$$

entrada :

$$\dots \# \sigma_1 \sigma_1 \dots \sigma_3 \# \dots$$

salida :

$$\dots \# \# \sigma_1 \sigma_1 \dots \sigma_3 \# \dots$$

- **reverse**: que dada una secuencia de símbolos (considerar un lenguaje que contenga solo 3 símbolos  $\sigma_1$ ,  $\sigma_2$  y  $\sigma_3$ ), la retorna invertida.

$$\sum = \{ \sigma_1, \sigma_2, \sigma_3, X, \# \}$$

entrada :

$$\dots \# \sigma_1 \sigma_1 \dots \sigma_2 \sigma_3 \# \dots$$

salida :

$$\dots \# \sigma_3 \sigma_1 \dots \sigma_2 \sigma_3 \# \sigma_3 \sigma_2 \dots \sigma_1 \sigma_3 \# \dots$$