

Trabajo Práctico 3

Análisis de Lenguajes de Programación

Alumnas:

Cipullo, Inés

Sullivan, Katherine

Universidad Nacional de Rosario

2021

Ejercicio 1

Tomemos $\Gamma = \{x : E \rightarrow E \rightarrow E, y : E \rightarrow E, z : E\}$

$$\begin{array}{c}
 \frac{x : E \rightarrow E \rightarrow E \in \Gamma}{\Gamma \vdash x : E \rightarrow E \rightarrow E} \text{ (T-VAR)} \quad \frac{z : E \in \Gamma}{\Gamma \vdash z : E} \text{ (T-VAR)} \quad \frac{y : E \rightarrow E \in \Gamma}{\Gamma \vdash y : E \rightarrow E} \text{ (T-VAR)} \quad \frac{z : E \in \Gamma}{\Gamma \vdash z : E} \text{ (T-VAR)} \\
 \frac{\Gamma \vdash x : E \rightarrow E \rightarrow E \quad \Gamma \vdash z : E}{\Gamma \vdash (x z) : E \rightarrow E} \text{ (T-APP)} \quad \frac{\Gamma \vdash y : E \rightarrow E \quad \Gamma \vdash z : E}{\Gamma \vdash (y z) : E} \text{ (T-APP)} \\
 \frac{\Gamma \vdash (x z) : E \rightarrow E \quad \Gamma \vdash (y z) : E}{\Gamma \vdash (x z) (y z) : E} \text{ (T-APP)} \\
 \frac{\Gamma \vdash (x z) (y z) : E}{\{x : E \rightarrow E \rightarrow E, y : E \rightarrow E\} \vdash \lambda z : E \cdot (x z) (y z) : E \rightarrow E} \text{ (T-ABS)} \\
 \frac{\{x : E \rightarrow E \rightarrow E, y : E \rightarrow E\} \vdash \lambda z : E \cdot (x z) (y z) : E \rightarrow E}{\{x : E \rightarrow E \rightarrow E\} \vdash \lambda y : E \rightarrow E \cdot \lambda z : E \cdot (x z) (y z) : (E \rightarrow E) \rightarrow E \rightarrow E} \text{ (T-ABS)} \\
 \frac{\{x : E \rightarrow E \rightarrow E\} \vdash \lambda y : E \rightarrow E \cdot \lambda z : E \cdot (x z) (y z) : (E \rightarrow E) \rightarrow E \rightarrow E}{\vdash \lambda x : E \rightarrow E \rightarrow E \cdot \lambda y : E \rightarrow E \cdot \lambda z : E \cdot (x z) (y z) : (E \rightarrow E \rightarrow E) \rightarrow (E \rightarrow E) \rightarrow E \rightarrow E} \text{ (T-ABS)}
 \end{array}$$

Ejercicio 2

La función *infer* retorna un valor de tipo *Either String Type* para poder retornar el tipo deseado en caso de que se haya podido realizar una inferencia de manera correcta, o una string que explique el error en caso de encontrarse con uno.

La función ($>>=$) toma un dato del tipo *Either String Type* y una función que dado un *Type* devuelve un *Either String Type*. Nuestra función chequea si el dato pasado es un *String* (en nuestra implementación entendámoslo como un error), si es el caso, no hace nada con este dato (simplemente propaga el error), en cambio, si el dato pasado es un *Type* le aplica la función pasada a este y ($>>=$) devuelve ese resultado.