



| **UNR** Universidad
Nacional de Rosario

FACULTAD DE CIENCIAS EXACTAS, INGENIERÍA
Y AGRIMENSURA

ANÁLISIS DE LENGUAJES DE PROGRAMACIÓN

Trabajo Práctico 3

Autor:

Tomás Castro Rojas
Blas Barbagelata

3 de noviembre de 2021

1. Ejercicio 1

Vamos a dar una derivación de tipo para el término S definido como

$$S = \lambda x : E \rightarrow E \rightarrow E. \lambda y : E \rightarrow E. \lambda z : E. (x z) (y z)$$

Sea $\Gamma' = \Gamma, \lambda x : E \rightarrow E \rightarrow E. \lambda y : E \rightarrow E \vdash \lambda z : E$.

Primero hacemos la derivación de tipo para la aplicación $(x z)$

$$\frac{\frac{x : E \rightarrow E \rightarrow E \in \Gamma'}{\Gamma' \vdash x : E \rightarrow E \rightarrow E} \text{T-VAR} \quad \frac{z : E \in \Gamma'}{\Gamma' \vdash z : E} \text{T-VAR}}{\Gamma' \vdash (x z) : E \rightarrow E} \text{T-APP}$$

Luego la derivación de tipo para la aplicación $(y z)$

$$\frac{\frac{y : E \rightarrow E \in \Gamma'}{\Gamma' \vdash y : E \rightarrow E} \text{T-VAR} \quad \frac{z : E \in \Gamma'}{\Gamma' \vdash z : E} \text{T-VAR}}{\Gamma' \vdash (y z) : E} \text{T-APP}$$

Juntamos los resultados y obtenemos la derivación de tipo para el término S .

$$\frac{\frac{\frac{\Gamma' \vdash (x z) : E \rightarrow E \quad \Gamma' \vdash (y z) : E}{\Gamma' \vdash (x z) (y z) : E} \text{T-APP} \quad \frac{\Gamma, \lambda x : E \rightarrow E \rightarrow E. \lambda y : E \rightarrow E \vdash \lambda z : E. (x z) (y z) : E \rightarrow E}{\Gamma, \lambda x : E \rightarrow E \rightarrow E \vdash \lambda y : E \rightarrow E. \lambda z : E. (x z) (y z) : E \rightarrow E \rightarrow E \rightarrow E} \text{T-ABS}}{\Gamma \vdash \lambda x : E \rightarrow E \rightarrow E. \lambda y : E \rightarrow E. \lambda z : E. (x z) (y z) : (E \rightarrow E \rightarrow E) \rightarrow (E \rightarrow E) \rightarrow E \rightarrow E} \text{T-ABS}$$

2. Ejercicio 2

Infer:

La función `infer` retorna el valor tipo `Either String Type` y no un valor tipo `Type` ya que puede ocurrir un error cuando se quiera determinar el tipo de un término. En caso de no poder inferir el tipo de un término, se detecta el error y se propaga en la ejecución del programa

(`>>=`): Este operador realiza una evaluación de los casos del argumento `v` (tipo `Either`). Si tiene la forma de `Left s`, el argumento recibido es un error, entonces (`>>=`) no lo modifica y lo retorna, propagando el error.

Si `v` tiene la forma de `Right t`, entonces (`>>=`) aplica la función `f` que recibe como argumento a `t`, y continua con la ejecución del programa.