DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Trabajo práctico 1

Programación funcional

20 de octubre de 2025 | Paradigmas ~~de Lenguajes~~ de Programación

Grupo pythonlovers

Integrante	LU	Correo electrónico
Comerci, Lucas	818/24	`lukicomerci@gmail.com`
Rancati, Hernan	785/00	`hernan.rancati@gmail.com`
Luis, Theo	130/23	`theoluis44@gmail.com`
Zea, Marcos	405/09	`zea.marcos@gmail.com`

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Ciudad Universitaria - (Pabellón I/Planta Baja)
Intendente Güiraldes 2610 - C1428EGA
Ciudad Autónoma de Buenos Aires - Rep. Argentina
Tel/Fax: (++54 +11) 4576-3300
http://www.exactas.uba.ar

Demostración

Enunciemos nuestro predicado a demostrar. Sea `e` :: `Expr`,

$$P(\texttt{e}) \equiv \texttt{cantLit e = S (cantOp e)}$$

Casos bases. Tenemos dos casos base, uno donde `e` es una constante y otra donde es un rango.

- Caso `e` = `Const a`, donde `a` :: `Float`:

$$P(\texttt{Const a}) \equiv \texttt{cantLit (Const a) = S (cantOp (Const a))}$$
$$\texttt{S Z} \overset{(L1)}{=} \texttt{cantLit (Const a) = S (cantOp (Const a))} \overset{(01)}{=} \texttt{S Z}$$
$$\texttt{S Z = S Z} \implies P(\texttt{Const a})$$

- Caso `e` = `Rango a b`, donde `a`,`b` :: `Float`:

$$P(\texttt{Rango a b}) \equiv \texttt{cantLit (Rango a b) = S (cantOp (Rango a b))}$$
$$\texttt{S Z} \overset{(L2)}{=} \texttt{cantLit (Rango a b) = S (cantOp (Rango a b))} \overset{(02)}{=} \texttt{S Z}$$
$$\texttt{S Z = S Z} \implies P(\texttt{Rango a b})$$

Casos inductivos.

- $\forall$`x,y` :: `Expr`.$P(\texttt{x}) \wedge P(\texttt{y}) \implies P(\texttt{Suma x y})$

  HI. $P(\texttt{x}) \wedge P(\texttt{y})$

  TI. $P(\texttt{Suma x y}) \equiv \texttt{cantLit (Suma x y) = S (cantOp (Suma x y))}$

  Desarrollemos el lado izquierdo de la igualdad de la TI:

$$\begin{aligned}
\texttt{cantLit (Suma x y)} &\overset{(L3)}{=} \texttt{suma (cantLit x) (cantLit y)} \overset{(HI)}{=} \texttt{suma (S (cantOp x)) (S (cantOp y))} \\
&\overset{(S2)}{=} \texttt{S (suma (cantOp x) (S (cantOp y)))} \overset{(CONMUT)}{=} \texttt{S (suma (S (cantOp y) (cantOp x)))} \\
&\overset{(S2)}{=} \texttt{S (S (suma (cantOp y) (cantOp x)))} \overset{(CONMUT)}{=} \texttt{S (S (suma (cantOp x) (cantOp y)))} \\
&\overset{(03)}{=} \texttt{S (cantOp (Suma x y)}
\end{aligned}$$
$$\texttt{cantLit (Suma x y) = S (cantOp (Suma x y)} \implies P(\texttt{Suma x y})$$

- $\forall$`x,y` :: `Expr`.$P(\texttt{x}) \wedge P(\texttt{y}) \implies P(\texttt{Resta x y})$. Es análogo al caso suma.
- $\forall$`x,y` :: `Expr`.$P(\texttt{x}) \wedge P(\texttt{y}) \implies P(\texttt{Mult x y})$. Es análogo al caso suma.
- $\forall$`x,y` :: `Expr`.$P(\texttt{x}) \wedge P(\texttt{y}) \implies P(\texttt{Div x y})$. Es análogo al caso suma.

Hemos probado los casos bases e inductivos. Entonces $\forall$`e` :: `Expr`.$P(\texttt{e})$.