Ejercicio 12

Necesitamos demostrar que toda expresión tiene un literal más que su cantidad de operadores. Los literales son las constantes y los rangos. Para esto se dispone de las siguientes definiciones:

1. Enunciado

```
data Nat = Z | S Nat


data Expr  = Const Float
           | Rango Float Float
           | Suma Expr Expr
           | Resta Expr Expr
           | Mult Expr Expr
           | Div Expr Expr

suma : Nat → Nat → Nat
suma Z m = m                                            {S1}
suma (S n) m = S(suma n m)                              {S2}

cantLit :: Expr ->Nat
cantLit (Const _) = S Z                                 {L1}
cantLit (Rango _ _) = S Z                               {L2}
cantLit (Suma a b) = suma (cantLit a) (cantLit b)       {L3}
cantLit (Resta a b) = suma (cantLit a) (cantLit b)      {L4}
cantLit (Mult a b) = suma (cantLit a) (cantLit b)       {L5}
cantLit (Div a b) = suma (cantLit a) (cantLit b)        {L6}

cantOp :: Expr → Nat
cantOp (Const _) = Z                                    {O1}
cantOp (Rango _ _) = Z                                  {O2}
cantOp (Suma a b) = S (suma (cantOp a) (cantOp b))      {O3}
cantOp (Resta a b) = S (suma (cantOp a) (cantOp b))     {O4}
cantOp (Mult a b)= S (suma (cantOp a) (cantOp b))       {O5}
cantOp (Div a b) = S (suma (cantOp a) (cantOp b))       {O6}
```

Nos piden demostrar la siguiente afirmación

$$\forall e ::= \texttt{Expr. cantLit e = S (cantOp e)}$$

2. Lema

Vamos primero a plantear el siguiente lema:

$$\forall\texttt{n}, \forall\texttt{m:Nat. suma Sn Sm = S(S(suma n m))}$$

Lo demostramos por Inducción :

Planteamos el predicado unario

`P(n) =` $\forall$`m:Nat. suma Sn Sm = S(S(suma n m))`

Queremos demostrar que $\forall$`n :Nat.P(n)` vale.

Al ser `n : Nat`, hay que analizar el caso de ambos constructores, `Z` o `S Nat`. El primer constructor corresponde al caso base, dado que no es recursivo, y el segundo requiere el uso de la hipotesis inductiva.

Vamos a probar que `P(n)` vale con n = `Z` **(Caso Base)**:

$$\begin{aligned}
\texttt{suma SZ Sm} &= \texttt{S(S(suma Z m))} & \\
\texttt{S(suma Z Sm)} &= \texttt{S(S(suma Z m))} & \{S2\} \\
\texttt{S(Sm)} &= \texttt{S(S(suma Z m))} & \{S1\} \\
\texttt{S(Sm)} &= \texttt{S(Sm)} & \{S1\}
\end{aligned}$$

✓**cumple el caso base**

Paso inductivo `n = Sn'`

Vamos a suponer que `P(n')` vale, o sea que

$$\forall\texttt{n' : Nat. P(n') = }\forall\texttt{m:Nat.suma Sn' Sm = S(S(suma n' m))} \qquad \{HI\}$$

Queremos probar que entonces vale `P(n)`

$$\begin{aligned}
\texttt{suma (S(Sn')) Sm} &= \texttt{S(S(suma Sn' m))} & \\
\texttt{S(suma Sn' Sm)} &= \texttt{S(S(suma Sn' m))} & \{\text{S2}\} \\
\texttt{S(S(S(suma n' m)))} &= \texttt{S(S(suma Sn' m))} & \{\text{HI}\} \\
\texttt{S(S(S(suma n' m)))} &= \texttt{S(S(S(suma n' m)))} & \{\text{S2}\}
\end{aligned}$$

✓**cumple el paso inductivo**

Como pudimos probar que `P(n)` vale $\forall$`n: Nat.` Entonces, nuestro Lema vale.

3. Demostración de la propiedad

Recordemos la propiedad a demostrar

$$\forall\texttt{e ::= Expr. cantLit e = S (cantOp e)}$$

Vamos a demostrarla usando inducción sobre e.

Planteamos el predicado unario: `P(e)= cantLit e = S (cantOp e)`

Queremos probar que se cumple ∀`e: Expr. P(e)`

`Expr.` tiene varios constructores pero solo vamos a demostrar la propiedad para `Const a`, `Rango a b.` y `Suma e f` con `a, b:Float` y `e, f : Expr`. Los demas constructores son analogos a `Suma`.

Probemos los casos base `e= Const a. a: Float`:

$$\begin{aligned}
\texttt{cantLit (Const a)} &= \texttt{S (cantOp (Const a)} \\
\texttt{S Z} &= \texttt{S (cantOp (Const a)} && \{\text{L1}\} \\
\texttt{S Z} &= \texttt{S Z} && \{\text{O1}\}
\end{aligned}$$

✓**cumple este caso base**

Caso `e= Rango a b. a,b:Float`

$$\begin{aligned}
\texttt{cantLit (Rango a b)} &= \texttt{S (cantOp (Rango a b)} \\
\texttt{S Z} &= \texttt{S (cantOp (Rango a b)} && \{\text{L2}\} \\
\texttt{S Z} &= \texttt{S Z} && \{\text{O2}\}
\end{aligned}$$

✓**cumple este caso base**

Paso inductivo, `e = Suma f g. f, g: Expr`

Siendo `e = Suma e' f'`, vamos a ver que `P(e') ^ P(f')` $\implies$ `P(Suma e' f')`

Suponemos que `P(e')` y `P(f')` valen. Es decir :

$$\begin{aligned}
&\forall\texttt{e': Expr. P(e')=cantLit e' = S (cantOp e')} && \{\text{HI 1}\} \\
&\forall\texttt{f': Expr. P(f')=cantLit f' = S (cantOp f')} && \{\text{HI 2}\}
\end{aligned}$$

Queremos ver que `P(e)`

$$\begin{aligned}
\texttt{cantLit (Suma e' f')} &= \texttt{S (cantOp (Suma e' f'))} \\
\texttt{suma (cantLit e') (cantLit f')} &= \texttt{S (cantOp (Suma e' f'))} && \{\text{L3}\} \\
\texttt{suma (S (cantOp e')) (S (cantOp f'))} &= \texttt{S (cantOp (Suma e' f'))} && \{\text{H1 y H2}\} \\
\texttt{S (S (suma (cantOp e') (cantOp f')))} &= \texttt{S (cantOp (Suma e' f'))} && \{\text{Lema}\} \\
\texttt{S (S (suma (cantOp e') (cantOp f')))} &= \texttt{S (S (suma (cantOp e') (cantOp f')))} && \{\text{O3}\}
\end{aligned}$$

✓**cumple el paso inductivo**

Como pudimos probar que `P(e)` vale ∀`e:Expr.` Podemos decir que la propiedad es verdadera.