PLP - Clase de repaso

Para el Primer Parcial

Departamento de Computación
FCEyN UBA

1c2025

Dado el tipo de los árboles estrictamente binarios:

```
data AEB a = Hoja a | Bin (AEB a) a (AEB a)
```

- Definir el esquema de recursión estructural `foldAEB` para este tipo.
- Usando `foldAEB`, definir la función:
  `esPreRama :: Eq a => AEB a -> [a] -> Bool`, que indique si una lista es prefijo de una rama del árbol, sin recorrer más de lo necesario.

Versión de `esPreRama` con recursión explícita:

```
data AEB a = Hoja a | Bin (AEB a) a (AEB a)

esPreRama :: Eq a => AEB a -> [a] -> Bool
esPreRama (Hoja x) = \xs -> null xs || (xs == [x])
esPreRama (Bin i r d) = \xs -> null xs ||
    (r == head xs && (esPreRama i (tail xs) ||
                      esPreRama d (tail xs)))
```

Dar una definición de `esPreRama` usando `foldAEB`.

Dadas las siguientes definiciones:

```
data AB a = Nil | Bin (AB a) a (AB a)

    const ::  a -> b -> a
{C} const = (\x -> \y -> x)

     altura ::  AB a -> Int
{A0} altura Nil         = 0
{A1} altura (Bin i r d) = 1 + max (altura i) (altura d)

     zipAB ::  AB a -> AB b -> AB (a,b)
{Z0} zipAB Nil         = const Nil
{Z1} zipAB (Bin i r d) = \t -> case t of
         Nil          -> Nil
         Bin i' r' d' -> Bin (zipAB i i') (r,r') (zipAB d d')
```

Demostrar la siguiente propiedad:

$$\forall\ \texttt{t::AB a.}\forall\ \texttt{u::AB a.}\quad \texttt{altura t} \geq \texttt{altura (zipAB t u)}$$

Se recomienda hacer inducción en un árbol, utilizando el lema de generación para el otro cuando sea necesario. Se permite definir macros (*i.e.*, poner nombres a expresiones largas para no tener que repetirlas). No es obligatorio escribir los $\forall$ correspondientes en cada paso, pero es importante recordar que están presentes. Recordar también que los $=$ de las definiciones pueden leerse en ambos sentidos. Se consideran demostradas todas las propiedades conocidas sobre enteros y booleanos, así como también:

$$\{\texttt{LEMA}\}\ \forall\ \texttt{t::AB a.}\quad \texttt{altura t} \geq 0$$

Reglas de deducción natural

$$\frac{}{\Gamma, \tau \vdash \tau}\ \text{ax}$$

$$\frac{\Gamma \vdash \tau \quad \Gamma \vdash \sigma}{\Gamma \vdash \tau \wedge \sigma}\ \wedge_i \qquad \frac{\Gamma \vdash \tau \wedge \sigma}{\Gamma \vdash \tau}\ \wedge_{e_1} \qquad \frac{\Gamma \vdash \tau \wedge \sigma}{\Gamma \vdash \sigma}\ \wedge_{e_2}$$

$$\frac{\Gamma, \tau \vdash \sigma}{\Gamma \vdash \tau \Rightarrow \sigma}\ \Rightarrow_i \qquad \frac{\Gamma \vdash \tau \Rightarrow \sigma \quad \Gamma \vdash \tau}{\Gamma \vdash \sigma}\ \Rightarrow_e$$

$$\frac{\Gamma \vdash \tau}{\Gamma \vdash \tau \vee \sigma}\ \vee_{i_1} \qquad \frac{\Gamma \vdash \sigma}{\Gamma \vdash \tau \vee \sigma}\ \vee_{i_2} \qquad \frac{\Gamma \vdash \tau \vee \sigma \quad \Gamma, \tau \vdash \rho \quad \Gamma, \sigma \vdash \rho}{\Gamma \vdash \rho}\ \vee_e$$

$$\frac{\Gamma, \tau \vdash \bot}{\Gamma \vdash \neg\tau}\ \neg_i \qquad \frac{\Gamma \vdash \tau \quad \Gamma \vdash \neg\tau}{\Gamma \vdash \bot}\ \neg_e$$

$$\frac{\Gamma \vdash \bot}{\Gamma \vdash \tau}\ \bot_e$$

Lógica intuicionista

$$\frac{\Gamma \vdash \neg\neg\tau}{\Gamma \vdash \tau}\ \neg\neg_e \qquad \frac{\Gamma, \neg\tau \vdash \bot}{\Gamma \vdash \tau}\ \text{PBC} \qquad \frac{}{\Gamma \vdash \tau \vee \neg\tau}\ \text{LEM}$$

Lógica clásica (las tres son equivalentes entre sí)

Cálculo Lambda - Práctica 4 Ejercicio 27

Se desea extender el Cálculo Lambda tipado con colas bidireccionales (también conocidas como *deque*).
Se extenderán los tipos y términos de la siguiente manera:

$\tau ::= \cdots \mid \text{Cola}_\tau$

$M ::= \cdots \mid \langle\rangle_\tau \mid M \bullet M \mid \text{próximo}(M) \mid \text{desencolar}(M)$
$\quad\quad \mid \text{case } M \text{ of } \langle\rangle \rightsquigarrow M; c \bullet x \rightsquigarrow M$

donde $\langle\rangle_\tau$ es la cola vacía en la que se pueden encolar elementos de tipo τ; $M_1 \bullet M_2$ representa el agregado del elemento M_2 al **final** de la cola M_1; los observadores próximo(M_1) y desencolar(M_1) devuelven, respectivamente, el primer elemento de la cola (el primero que se encoló), y la cola sin el primer elemento (estos dos últimos solo tienen sentido si la cola no es vacía); y el observador case M_1 of $\langle\rangle \rightsquigarrow M_2; c \bullet x \rightsquigarrow M_3$ permite operar con la cola en sentido contrario, accediendo al último elemento encolado (cuyo valor se ligará a la variable x en M_3) y al resto de la cola (que se ligará a la variable c en el mismo subtérmino).

Cálculo Lambda - Práctica 4 Ejercicio 27

$\tau ::= \ldots \mid \mathsf{Cola}_\tau$
$M ::= \ldots \mid \langle\rangle_\tau \mid M \bullet M \mid \text{próximo}(M) \mid \text{desencolar}(M)$
$\quad \mid \text{case } M \text{ of } \langle\rangle \rightsquigarrow M; c \bullet x \rightsquigarrow M$

1. Introducir las reglas de tipado para la extensión propuesta.
2. Definir el conjunto de valores y las nuevas reglas de reducción. Pueden usar los conectivos booleanos de la guía. **No es necesario escribir las reglas de congruencia,** basta con indicar cuántas son.
   Pista: puede ser necesario mirar más de un nivel de un término para saber a qué reduce.
3. Mostrar paso por paso cómo reduce la expresión:
   $\text{case } \langle\rangle_{\mathsf{Nat}} \bullet \underline{1} \bullet 0 \text{ of } \langle\rangle \rightsquigarrow \text{próximo}(\langle\rangle_{\mathsf{Bool}}); c \bullet x \rightsquigarrow \text{isZero}(x)$
4. Definir como macro la función $\texttt{último}_\tau$, que dada una cola devuelve el último elemento que se encoló en ella. Si la cola es vacía, puede colgarse o llegar a una forma normal bien tipada que no sea un valor. Dar un juicio de tipado válido para esta función (no es necesario demostrarlo).

¿Preguntas?

¿ ¿ ¿ ¿ ¿ ¿ ¿ ¿ ¿? ? ? ? ? ? ? ? ?