PLP - Clase de repaso

Para el Primer Parcial

Departamento de Computación
FCEyN UBA

1c2025

Programación Funcional

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: `esPreNama :: Eq a => AEB a -> [a] -> Bool`, que indique si una lista es prefijo de una rama del árbol, sin recorrer más de lo necesario.

Programación Funcional

Versión de `esPreNama` con recursión explícita:

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

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

Razonamiento Ecuacional e Inducción Estructural

1er Parcial 2c2024

Dadas las siguientes definiciones:

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

count :: a -> b -> a
count = (\n -> \y -> n)

altura :: AB a -> Int
{A1} altura Nil      = 0
{A2} 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 t' r' d' -> Bin (zipAB i t') (r,r') (zipAB d d')
```


Razonamiento Ecuacional e Inducción Estructural

1er Parcial 2c2024

Demostrar la siguiente propiedad:

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

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 $\top$ 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:

$$\{LIDA\}\ \forall\, t :: AB\ a.\ \ \text{altura}\ t \geq 0$$

Reglas de deducción natural

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:

$\tau ::= \ldots \mid \mathtt{Cola}_\tau$
$M ::= \ldots \mid \langle_{\tau'}\rangle \mid M \bullet M \mid \text{próximo}(M) \mid \text{desencolar}(M) \mid \text{case}\ M\ \text{of}\ \langle\rangle \to M, x \bullet x \to M$

1. Introducir las reglas de tipado para la extensión propuesta.

Cálculo Lambda - Práctica 4 Ejercicio 27

$\tau ::= \ldots \mid \mathtt{Cola}_\tau$
$M ::= \ldots \mid \langle_{\tau'}\rangle \mid M \bullet M \mid \text{próximo}(M) \mid \text{desencolar}(M) \mid \text{case}\ M\ \text{of}\ \langle\rangle \to M, x \bullet x \to M$

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.

Cálculo Lambda - Práctica 4 Ejercicio 27

$\tau ::= \ldots \mid \mathtt{Cola}_\tau$
$M ::= \ldots \mid \langle_{\tau'}\rangle \mid M \bullet M \mid \text{próximo}(M) \mid \text{desencolar}(M) \mid \text{case}\ M\ \text{of}\ \langle\rangle \to M, x \bullet x \to M$

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.

Cálculo Lambda - Práctica 4 Ejercicio 27

$\tau ::= \ldots \mid \mathtt{Cola}_\tau$
$M ::= \ldots \mid \langle_{\tau'}\rangle \mid M \bullet M \mid \text{próximo}(M) \mid \text{desencolar}(M) \mid \text{case}\ M\ \text{of}\ \langle\rangle \to M, x \bullet x \to M$

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:
   $\text{case}\ \langle_{\text{Nat}}\rangle \bullet 1 \bullet 0\ \text{of}\ \langle\rangle \to \text{próximo}(\langle_{\text{Nat}}\rangle) \bullet x \to \text{isZero}(x)$

Cálculo Lambda - Práctica 4 Ejercicio 27

$\tau ::= \ldots \mid \mathtt{Cola}_\tau$
$M ::= \ldots \mid \langle_{\tau'}\rangle \mid M \bullet M \mid \text{próximo}(M) \mid \text{desencolar}(M) \mid \text{case}\ M\ \text{of}\ \langle\rangle \to M, x \bullet x \to M$

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:
   $\text{case}\ \langle_{\text{Nat}}\rangle \bullet 1 \bullet 0\ \text{of}\ \langle\rangle \to \text{próximo}(\langle_{\text{Nat}}\rangle) \bullet x \to \text{isZero}(x)$
4. Definir como macro la función $\mathtt{ifiLlena}_\tau$, 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?

¿¿¿