DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Trabajo práctico 1

Programación Funcional

20 de septiembre de 2024 — Paradigmas de Programación

Grupo CHAD sociedad anónima

Integrante	LU	Correo electrónico
Condori Llanos, Alex	163/23	nocwe11@gmail.com
Della Rosa, Facundo César	1317/23	dellarosafacundo@gmail.com
López Porto, Gregorio	1376/23	gregoriolopezporto@gmail.com
Winogron, Iván	459/23	Ivowino2000@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

Ejercicio 9

Enunciado

De acuerdo a las definiciones de las funciones para árboles ternarios de más arriba, se pide demostrar lo siguiente:

$$\forall t :: \text{AT a} \ . \ \forall x :: \text{a} \ . \ (\text{elem x (preorder t)} = \text{elem x (postorder t)})$$

Definiciones

elem :: Eq a $\implies$ a $\rightarrow$ [a] $\rightarrow$ Bool
{E0} elem e [ ] = False
{E1} elem e (x:xs) = (e == x) || elem e xs

preorder :: Procesador (AT a) a
{PRE1} preorder = foldAT (\x ri rc rd $\rightarrow$ [x] ++ ri ++ rc ++ rd) [ ]

postorder :: Procesador (AT a) a
{POST1} postorder = foldAT (\x ri rc rd $\rightarrow$ ri ++ rc ++ rd ++ [x]) [ ]

$foldAT :: (a \rightarrow b \rightarrow b \rightarrow b \rightarrow b) \rightarrow b \rightarrow ATa \rightarrow b$
{F0} foldAT f b Nil = b
{F1} foldAT f b (Tern r ri rc rd) = f a (foldAT f b ri) (foldAT f b rc) (foldAT f b rd)

Demostración

Hacemos la demostración utilizando inducción estructural en el arbol ternario t. Sea P(t) la propiedad:

$$\text{P(t)} = \text{elem x (preorder t)} = \text{elem x (postorder t)}$$

Primero pruebo la propiedad para el caso base:

$$\text{P(Nil)} = \text{elem x (preorder Nil)} = \text{elem x (postorder Nil)}$$

elem x (preorder Nil) $\underset{\{PRE1\}}{=}$ elem x (foldAT (\x ri rc rd $\rightarrow$ [x] ++ ri ++ rc ++ rd) [ ] Nil) $\underset{\{F0\}}{=}$ elem x [ ]
elem x (postorder Nil) $\underset{\{POST1\}}{=}$ elem x (foldAT (\x ri rc rd $\rightarrow$ concat ri ++ rc ++ rd ++ [x]) [ ] Nil) $\underset{\{F0\}}{=}$ elem x [ ]

Entonces elem x (preorder Nil) = elem x (postorder Nil) = elem x [ ]. Luego vale la propiedad para el caso base P(Nil).
Pruebo el paso inductivo:

$$\forall \text{ h1} :: \text{AT a}, \forall \text{h2} :: \text{AT a}, \forall \text{h3} :: \text{AT a}, \forall \text{r} :: \text{a},$$
$$\text{P(h1)} \land \text{P(h2)} \land \text{P(h3)} \land \implies \text{P(Tern r h1 h2 h3)}$$

Defino las propiedades
P(h1) = elem x (preorder h1) = elem x (postorder h1)
P(h2) = elem x (preorder h2) = elem x (postorder h2)
P(h3) = elem x (preorder h3) = elem x (postorder h3)
P(Tern r h1 h2 h3) = elem x (preorder (Tern r h1 h2 h3)) = elem x (postorder (Tern r h1 h2 h3))
Entonces, mi hipótesis inductiva es que valen las propiedades P(h1), P(h2), P(h3), y quiero ver que vale P(Tern r h1 h2 h3).

elem x (postorder (Tern r h1 h2 h3)) $\underset{\{POST1\}}{=}$ elem x (foldAT (\x ri rc rd $\rightarrow$ ri ++ rc ++ rd ++ [x]) [ ]) (Tern r h1 h2 h3)

considero f = (\x ri rc rd $\rightarrow$ ri ++ rc ++ rd ++ [x]) para facilitar la lectura.
$\underset{\{F1\}}{=}$ elem x ((f a (foldAT f [ ] ri) (foldAT f [ ] rc) (foldAT f [ ] rd)) (Tern r h1 h2 h3))

$\underset{\beta\to}{=}$ elem x ((\x ri rc rd $\to$ ri ++ rc ++ rd ++ [x]) r (foldAT f [ ] h1) (foldAT f [ ] h2) (foldAT f [ ] h3))
$\underset{\beta\to}{=}$ elem x ((foldAT f [ ] h1) ++ (foldAT f [ ] h2) ++ (foldAT f [ ] h3) ++ [r])

Propongo el siguiente lema:

$$\forall \text{ a} :: [z], \forall b :: [z], \forall c :: [z], \forall d :: [z],$$
$$\text{elem x (a ++ b ++ c ++ d) = elem x a || elem x b || elem x c || elem x d}$$

considero g = (\x ri rc rd $\to$ [x] ++ ri ++ rc ++ rd) para facilitar la lectura.
$\underset{\{lema\}}{=}$ elem x (foldAT f [ ] h1) || elem x (foldAT f [ ] h2) || elem x (foldAT f [ ] h3) || elem x [r]
$\underset{\{POST1\}}{=}$ elem x (postorder h1) || elem x (postorder h2) || elem x (postorder h3) || elem x [r]
$\underset{\{HI's\}}{=}$ elem x (preorder h1) || elem x (preorder h2) || elem x (preorder h3) || elem x [r]

Utilizando la conmutatividad de la disyunción, cambio de lugar los términos
= elem x [r] || elem x (preorder h1) || elem x (preorder h2) || elem x (preorder h3)
$\underset{\{lema\}}{=}$ elem x ([r] ++ (preorder h1) ++ (preorder h2) ++ (preorder h3))
$\underset{\{\beta\leftarrow\}}{=}$ elem x ((\x ri rc rd $\to$ ([x] ++ ri ++ rc ++ rd)) r (foldAT g [ ] h1) (foldAT g [ ] h2) (foldAT g [ ] h3))
$\underset{\{\beta\leftarrow\}}{=}$ elem x ((g a (foldAT g [ ] ri) (foldAT g [ ] rc) (foldAT g [ ] rd)) (Tern r h1 h2 h3))
$\underset{\{PRE1\}}{=}$ elem x (preorder (Tern r h1 h2 h3))
Entonces, mediante una cadena de igualdades, concluyo que elem x (preorder (Tern r h1 h2 h3)) = elem x (postorder (Tern r h1 h2 h3)), que es lo que quería probar.

Demostración del lema elem x (a ++ b ++ c ++ d) = elem x a || elem x b || elem x c || elem x d
voy a demostrar primero una versión simplificada para luego demostrar el lema.
llamo Q a la propiedad: elem x (l_1 ++ l_2) = elem x l_1 || elem x l_2
Para demostrar que las funciones son iguales, basta con ver que

$$\forall\ l_1 :: [a], \forall l_2 :: [a]\ .\ \text{elem x } (l_1 \text{ ++ } l_2) = \text{elem x } l_1 \text{ || elem x } l_2$$

Para la demostración voy a hacer inducción estructural sobre la lista l_1.
caso base Q([]) = elem x ([] ++ l_2) $\underset{\{lista\}}{=}$ elem x l_2.
por otro lado, elem x [] || elem x l_2 $\underset{\{E0\}}{=}$ False || elem x l_2 = elem x l_2
luego vale la propiedad Q para el caso base. Veo ahora el paso inductivo

$$\forall \text{ ys} :: [a], \forall y :: a\ .\ Q(ys) \implies Q(y{:}ys)$$

defino las propiedades
Q(ys) = elem x (ys ++ l_2) = elem x ys || elem x l_2
Q(y:ys) = elem x ((y:ys) ++ l_2) = elem x y:ys || elem x l_2

Entonces, mi hipótesis inductiva es que vale la propiedad Q(ys) y mi objetivo es ver que vale Q(y:ys).
elem x ((y:ys) ++ l_2) $\underset{\{E1\}}{=}$ x == y || elem x (ys ++ l_2) $\underset{\{HI\}}{=}$ x == y || elem x ys || elem x l_2
= (x == y || elem x ys) || elem x l_2 $\underset{\{E1\}}{=}$ elem x (y:ys) || elem x l_2
entonces vale la propiedad Q.
Ahora para demostrar el lema, considero l_1 = a, l_2 = (b ++ c ++ d) y aplico la propiedad Q:
elem x (a ++ b ++ c ++ d) $\underset{\{Q\}}{=}$ elem x a || elem x (b ++ c ++ d)
Siguiendo la misma lógica, considero l_1 = b, l_2 = (c ++ d) y aplico nuevamente la propiedad Q sobre el segundo término:
elem x a || elem x (b ++ c ++ d) $\underset{\{Q\}}{=}$ elem x a || elem x b || elem x (c ++ d)
Aplico la propeidad Q nuevamente considerando l_1 = c, l_2 = d:
elem x a || elem x b || elem x (c ++ d) $\underset{\{Q\}}{=}$ elem x a || elem x b || elem x c || elem x d
Luego, queda demostrado el lema elem x (a ++ b ++ c ++ d) = elem x a || elem x b || elem x c || elem x d.