List

Friday, 21 January 2022 10.44

List<Elem>

Const. de bază:

- [] : → List lista goală
- ":" : Elem x List → List adăugare el. în listă

$1 : [] = [1]$

$4 : [1,2,3] = [4,1,2,3] =$

$= 4 : 1 : 2 : 3 : []$

Operatori

- length : List → Nat
- ++ : List x List → List (concatenare)

Axiome

- $length([]) = 0$ LEN 1
- $length(x : xs) = 1 + length(xs)$ LEN 2
- $[] ++ L = L$ CONCAT 1
- $(x : xs) ++ L = x : (xs ++ L)$ CONCAT 2

Cerința: ++ e asoc. $(A ++ B) ++ C = A ++ (B ++ C)$ ∀ A, B, C Liste

Fac ind. după A.

Cazuri de bază

$A = []$

$([] ++ B) ++ C \overset{CONCAT1}{=} B ++ C$

$[] ++ (B ++ C) \overset{CONCAT1}{=} B ++ C$

"=" ∀ B, C

Pas de inducție

Pp. că avem $A = xs \Rightarrow A = x : xs$

Ip. inductivă

Pp $P(xs)$: $(xs ++ B) ++ C = xs ++ (B ++ C)$ ∀ B, C (ii)

Avem de dem. $P(x : xs)$: $\underbrace{[(x : xs) ++ B] ++ C}_{ST} = \underbrace{(x : xs) ++ (B ++ C)}_{DR}$

$ST = [(x : xs) ++ B] ++ C \overset{CONCAT\,2}{=} [x : (xs ++ B)] ++ C =$

$\overset{CONCAT\,2}{=} x : [(xs ++ B) ++ C]$

$DR = (x : xs) ++ (B ++ C) \overset{CONCAT\,2}{=} x : [xs ++ (B ++ C)]$

$\overset{ii}{\Rightarrow} ST = DR$

$\Rightarrow A$

Range (ex rezolvat completari)

Friday, 21 January 2022 11.06

range(a : l) = (a-1 = head(l)) și range(l)

3 2 1 → 4 3 2 1

head(a : l) = a = head(l)+1

⇒ head(l) = head(a : l)-1

a - 1 = head(l) ⇒ a = head(l)+1

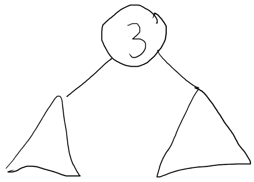
$$\text{sum}(a : l) = a + \underbrace{\text{sum}(l)}_{\text{i.i.}} = a + \text{head}(l) \cdot \frac{(\text{head}(l)+1)}{2} =$$

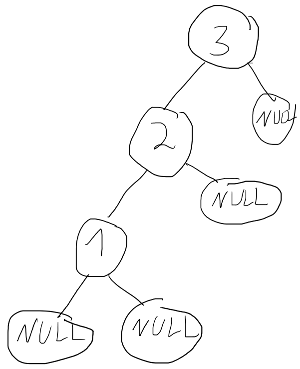
$$= \text{head}(a : l) + [\text{head}(a : l) - 1] \cdot [\text{head}(a : l) \cancel{-1} \cancel{+1}] \cdot \frac{1}{2} =$$

$$= \text{head}(a : l)\left[1 + \frac{\text{head}(a : l) - 1}{2}\right] = \frac{1}{2}\,\text{head}(a : l) \cdot [\text{head}(a : l) + 1]$$

Arbore binar

Friday, 28 January 2022 10.34





4. Fie tipul de date asbtract BinTree definit prin constructorii de bază:

$empty : \to BinTree$ # Arborele binar vid

$node : BinTree * Elem * BinTree \to BinTree$ # Subarborele stâng, radăcina, sub drept

Se consideră operatorii:

$\boldsymbol{size : BinTree \to N}$ # Numărul de elemente din arbore

$\boldsymbol{all : (Elem \to Bool) * BinTree \to Bool}$

Întoarce True dacă p(x) este adevărat pentru toate elementele x din Tree, False altfel

Axiomele pentru All:

(All1): $all(p, empty) = True$

(All2): $all(p, node(left, root, right)) = p(root)$ && $all(p, left)$ && $all(p, right)$

$\boldsymbol{countif : (Elem \to Bool) * BinTree \to N}$

countif(p, tree) = câte elemente din tree satisfac predicatul p (adică pentru câte elemente din tree p(x) este adevărat

Scrieți axiomele pentru operatorii size și countif, apoi demonstrați prin inducție structurală:

$$all(p, t) \to countif(p, t) == size(t) \quad (1)$$

nade(nade(nade(empty, 1, empty), 2, empty), 3, empty)

S1 size(empty) = 0

S2 size(nade(A, x, B)) = size(A) + 1 + size(B)

C1 countif(p, empty) = 0

C2 countif(p, nade(A, x, B)) = (p(x) ? 1 : 0) + countif(p, A) + countif(p, B)

all(p, t) -> countif(p, t) = size(t)

Pp. all(p, t) fals => nu avem de dem. nimic

Pp. all(p, t) A => avem de dem. countif(p, t) = size(t)

Caz de bază

t = empty

all(p, empty) = True

countif(p, emity) = 0 = size(empty) ✓

Ip. ind.

Pp. all(p, A) -> countif(p, A) = size(A)

și all(p, B) -> countif(p, B) = size(B)

Vreau să dem

all (p, node(A, x, B)) $\Rightarrow$ countif (p, node(A, x, B)) = size(node(A,x,B))

all (p, node(A, x, B)) $\overset{A2}{=}$

= p(x) și all (p, A) și all (p, B)

Dacă all (p, node(A,x,B)) = Fals $\Rightarrow$ nu mai am de dem.

Dacă all (p node(A,x,B) = True $\Rightarrow$ p(x) = True
all (p, A) = True
all (p, B) = True

ST = countif (p, node(A,x,B)) = size(node(A,x,B)) = DR

DR $\overset{S2}{=}$ size(A) + 1 + size(B)

ST $\overset{C2}{=}$ p(x) ? 1 : 0 + countif (p, A) + countif (p, B)

Dar, știm:

p(x) = True
all (p, A) = True $\overset{i.i.}{\Rightarrow}$ countif (p, A) = size(A)
all (p, B) = True $\overset{i.i.}{\Rightarrow}$ countif (p, B) = size(B)

$\Rightarrow$ ST = 1 + size(A) + size(B)

ST = DR $\Rightarrow$ (A)

Fold

Friday, 28 January 2022 11.07

- $[] : \to LIST$
- $(x : l) : INT * LIST \to LIST$

Fie FUN mulțimea funcțiilor definite astfel: $INT * INT \to INT$.
Se definesc funcțiile (operatori) următoare, cu axiomele aferente:

$\boldsymbol{fold : FUN * INT * LIST \to INT}$

- $fold(f, z, []) = z$
- $fold(f, z, x : l) = f(x, fold(f, z, l))$

$\boldsymbol{len : LIST \to INT}$

- $len([]) = 0$
- $len(x : l) = 1 + len(l)$

Să se demonstreze prin inducție structurală că P(l) este adevarată, pentru ∀ l ∈ LIST:
$\boldsymbol{P(l) = (len(l) = fold(inc, 0, l))}$
cu inc definită astfel $inc(x, y) = y + 1$

fold(f, accumulator_inițial, l)

l = [1, 2, 3]   f(el_cur, acc)

suma el)

f(a, b) = a + b

fold (f, 0, l)

Caz de bază

l = []

len([]) = 0
fold(inc, 0, []) = 0 ⟹ (A)

Ip. inductivă

I.I: P_n. len(l) = fold(inc, 0, l)

Dem. că len(x : l) = fold(inc, 0, x : l) = DR

ST = len(x : l) $\overset{L2}{=}$ 1 + len(l) $\overset{I.I}{=}$ 1 + fold(inc, 0, l)

DR = fold(inc, 0, x : l) $\overset{F2}{=}$ inc(x, fold(inc, 0, l)) =

$\overset{inc}{=}$ fold(inc, 0, l) + 1

⟹ ST = DR (A)