

Programare Funcțională – Laborator 7

Închideri funcționale

Isabela Drămnesc

1 Concepte

- Funcționale
- apply, filter, map, foldr, foldl, pairs
- Închideri funcționale

2 Discutarea temei din laboratorul 6

3 Exerciții

3.1 Definiți foldr

utilizând *foldl* și *reverse*.

```
(check-expect (foldr + 0 '(2 3 4 5)) 14)
```

3.2 Definiți filter

utilizând *lambda* și *foldr*.

```
(check-expect (filter positive? '(4 -2 1 2 -1 5)) '(4 1 2 5))
```

3.3 Definiți length

utilizând *lambda* și *foldl*.

```
(check-expect (length '(a b (2 3) 5)) 4)
```

3.4 Explicați următoarele două definiții.

- (a)

```
(define (nest f n)
  (if (= n 0)
      (lambda (x) x)
      (lambda (x) ((nest f (- n 1)) (f x)))))
```
- (b)

```
(define (nestwhile f v p)
  (if (p v) v (nestwhile f (f v) p)))
```

3.5 Explicați următoarea definiție.

```
(define (comp . flst)
  (foldr (lambda (f g) (lambda (x) (f (g x))))
        (lambda (x) x)
        flst))
```

3.6 Definiți funcția $list \rightarrow set$

care transformă o listă într-o mulțime (elimină duplicatele).

Hint: Exprimați calculul ($list \rightarrow set$ lst) ca ($foldr$ f $null$ lst) cu o funcție f potrivită. Puteți folosi funcția built-in ($member$ $elem$ lst) care este true dacă $elem$ este membru în lista lst și false altfel.

```
(check-expect (list->set '(1 2 4 1 2 6 4 1)) '(2 6 4 1))
```

```
(check-expect (list->set '(1 2 a b 4 a 2)) '(1 b 4 a 2))
```

3.7 Definiți o funcție $count - symb$

care numără fiecare apariție a unui simbol într-o listă. Exemplu

```
(check-expect (count-symb '(a b a b b c x z z x))
  '((a . 2) (b . 3) (c . 1) (z . 2) (x . 2)))
```

4 TEMA

4.1 Dați definiții recursive pentru fiecare dintre următoarele funcții:

(a) (f-map2 a b f1 ... fn)

care calculează (list w1 ... wn), unde w_i este valoarea (f_i a b) pentru fiecare i de la 1 la n . Exemple:

```
(check-expect (fmap-2 4 2 + * /) '(6 8 2))
```

```
(check-expect (fmap-2 4 2) '())
```

```
(check-expect (fmap-2 10 11 * / + (lambda (x y) (+ x y 100))) '(110 10/11 21 121))
```

(b) (inc? a1 ... an)

care primește ca input $n \geq 0$ întregi și returnează true dacă și numai dacă $a_1 < a_2 < \dots < a_n$. Exemple:

```
(check-expect (inc? 5 2 1 3) #f)
```

```
(check-expect (inc? 2 5 9 10 11) #t)
```

```
(check-expect (inc?) #t)
```

(check-expect (inc? 10) #t)

(check-expect (inc? 3 2 1) #f)

(c) (dec? a1 ... an)

care primește ca input $n \geq 0$ întregi și returnează true dacă și numai dacă $a_1 > a_2 > \dots > a_n$.

(d) (sorted? cmp a1 ... an)

care returnează true dacă și numai dacă $(cmp\ a_i a_{i+1})$ este adevărat pentru $1 \leq i \leq n$.

(e) (monotone? a1 ... an)

care primește ca input $n \geq 0$ întregi și returnează true dacă și numai dacă $a_1 < a_2 < \dots < a_n$ sau $a_1 > a_2 > \dots > a_n$.

4.2 Finalizați toate exercițiile din temele suplimentare.

Notă: Termen de realizare: laboratorul următor.