

TÖL304G Heimadæmi 6

Valtýr og Haraldur (vok4, hbd4)

10. október 2019

1. Dæmi

Ath. að tag breyta ræðst af tag lista og upphafsgildis og skilalagi fallanna tveggja f og op.

```
(*
** Notkun: mapreduce f op u x
** Fyrir: f er fall af einni viðfangsbreytu
**        op er fall af tveimur viðfangsbreytum
**        u er upphafsgildi af sama tagi og op skilar
**        x er listi
** Gildi: u op f(x1) op ... op f(xn)
*)
let mapreduce f op u x =
  (*
  ** Notkun: help c l
  ** Fyrir: c er uppsafnað gildi úr fyrri keyrslum
  **        (upphafsgildi í fyrstu keyrslu)
  **        l er listi
  ** Gildi: Skilar ekki fyrr en listi er tómur
  **        Þá skilar fallið u op f(x1) op ... op f(xn)
  *)
  let rec help c l =
    match l with
    | [] -> c
    | first::rest -> help (op c (f first)) rest
  in help u x
;;

Keyrsla:

let
  inc x = x+1
and
  add x y = x+y
in
  mapreduce inc add 0 [0;1;2;3;4];;

- : int = 15
```

2. Dæmi

```
(*
** Notkun: fromTo i j
** Fyrir: i og j eru heiltölur, i<=j
** Gildi: Listinn [i;i+1;...;j-1].
*)
let fromTo i j =
  (*
  ** Notkun: hjalp x lst
  ** Fyrir: x er heiltala
  **          lst er listi af heiltölum frá j upp að x
  ** Gildi: Listinn [i;i+1;...;j-1]
  **          (Skilar ekki neinu fyrr en x er jafn j)
  *)
  let rec hjalp x lst =
    match x with
    | x when x == j -> lst
    | _               -> (hjalp (x+1) (List.append lst [x]))
  in hjalp i []
;;

Keyrsla:

# fromTo 6 10;;
- : int list = [6; 7; 8; 9]
```

3. Dæmi

```

(*
** Notkun: insertAt x i z
** Fyrir: x=[x1;x2;...;xN] er listi gilda
**        af einhverju tagi 'a. z er gildi
**        af sama tagi 'a. i er heiltala,
**        0 <= i <= N, Thar sem N er lengd
**        listans x.
** Gildi: Listinn
**        [x1;x2;...;x_i;z;x_{i+1};...;xN],
**        Þ.e. listinn sem að kemur thegar
**        gildinu z er skeytt inn i listann
**        x rett fyrir aftan i fremstu
**        gildin.
*)
let insertAt x i z =
  (*
  ** Notkun: hjalp j nytt rest
  ** Fyrir: j er heiltala sem bendir á það
  **        verið er að skoða að hverju sinni.
  **        nytt er nýr listi
  **        rest er afgangur af lista
  ** Gildi: Listinn [x1;x2;...;x_i;z;x_{i+1};...;xN]
  **        Þ.e. listinn x með z bætt inn í
  **        staðsetningu i.
  *)
  let rec hjalp j nytt rest =
    if j == i then
      List.append nytt (List.append [z] rest)
    else match rest with
    | head::tail -> (hjalp (j + 1) (List.append nytt [head]) tail)
    | []          -> raise (Invalid_argument "attempt to take tail of empty list")
  in hjalp 0 [] x
;;

Keyrsla:

# insertAt [0;1;2;3] 2 4;;
- : int list = [0; 1; 4; 2; 3]

```

4. Dæmi

```
(*  
** Notkun: extendPermutation n z  
** Fyrir:  n >= 0 er heiltala.  
**        z er einhver umröðun listans [1;2;...;n-1].  
** Gildi:  Listi allra þeirra lista sem út koma þegar  
**        tölunni n er skeytt inn í listann z á  
**        einhverjum stað, allt frá byrjun til enda.  
** Aths.: Þetta er "one-liner" ef við áttum okkur á  
**        hvernig nota megja map og fromTo til að leysa  
**        verkefnið.  
*)  
let extendPermutation n z =  
  List.map (fun i -> (insertAt z i n)) (fromTo 0 n)  
;;
```

Keyrsla:

```
# extendPermutation 3 [1;2;3];;  
- : int list list = [[3; 1; 2; 3]; [1; 3; 2; 3]; [1; 2; 3; 3]]
```

5. Dæmi

```
(*  
** Notkun: length x  
** Fyrir: x=[x1;x2;...;xN] er listi.  
** Gildi: N.  
** Aths.: Þetta skal útfæra sem "one-liner" með hjálp  
**        fallsins it_list.  
*)  
let length x =  
  List.fold_left (fun acc x -> acc + 1) 0 x  
;;
```

Keyrsla:

```
# length [1;2;3;4];;  
- : int = 4
```

6. Dæmi

```
# length (permutations 6);;  
- : int = 720
```