

Forritunarmál Einstaklingsverkefni 6

brj46

October 202

1

Skrifum fall lengd í CAML sem notar *list_it*

```
(*
Notkun: lengd x
Fyrir: x er listi af hvaða tagi sem er.
Gildi: Skilar lengd lista x sem heiltölu.
       Fallið notar list_it til að framkvæma vinnsluna.
       list_it er ítrari sem tekur við þremur viðföngum:
         - Lambda-fall
         - Upphafsgildi acc
         - Listann sem á að ítra yfir.
*)
let list_it = List.fold_left;;

let lengd x =
  list_it (fun acc _ -> acc + 1) 0 x
;;
```

Hér er skjáskot af keyrslu:

```
# let list_it = List.fold_left;;
val list_it : ('a -> 'b -> 'a) -> 'a -> 'b list -> 'a = <fun>
# let lengd x =
  list_it (fun acc _ -> acc + 1) 0 x
;;
val lengd : 'a list -> int = <fun>
# lengd [1; 2; 3; 4; 5; 9];;
- : int = 6
#
```

2

Skrifum fall *powerList* í **CAML** sem tekur lista $x = [x_1; x_2; \dots; x_n]$ sem viðfang og skilar lista af listum sem inniheldur alla mögulega lista sem eru undirlistar listans x í þeim skilningi að undirlistarnir innihalda gildi úr listanum x í sömu röð og í x , nema hvað fjarlægja má núll eða fleiri gildi úr listanum x .

Fall:

(*
Notkun: powerList i j
Fyrir: x er listi af heiltölum.
Gildi: Listi allra undirlista lista x þar sem undirlistar
innihalda gildi úr listanum x í sömu röð og í x.
Til dæmis, powerList [1;2] skilar [[]; [2]; [1]; [1;2]].

*)

```
let rec powerList x =  
  match x with  
  | [] -> [[]]  
  | h::t ->  
    let undirlistar = powerList t in  
    List.map (fun l -> h::l) sublists @ sublists  
;;
```

Dæmi um keyrslu:

```
# let rec powerList x =  
  match x with  
  | [] -> [[]]  
  | h::t ->  
    let undirlistar = powerList t in  
    List.map (fun l -> h::l) undirlistar @ undirlistar  
;;  
val powerList : 'a list -> 'a list list = <fun>  
# powerList [1; 2];;  
- : int list list = [[1; 2]; [1]; [2]; []]
```