

# TÖL304G Heimadæmi 2

Valtýr, Árni og Haraldur (vok4, afm7, hbd4)

12. september 2019

## 1. Dæmi

---

Skrifið fall `fcompose` sem tekur tvö viðföng,  $f$  og  $g$  sem hvort tveggja eiga að vera föll sem taka eitt viðfang. Kallið `(fcompose f g)` skal skila samsetta fallinu  $f \circ g$ , þ.e fallinu  $h$  þar sem  $h(x)$  skilar  $f(g(x))$

*Lausn:*

```
;; Notkun: ((fcompose f g) x)
;; Fyrir:  f og eru einundarföll, f er löglegt viðfang
;;         í g, (g x) er löglegt viðfang í f.
;; Gildi:  (f (g x))
(define (fcompose f g)
  (lambda (x) (f (g x))))
)
```

## 2. Dæmi

---

Skrifið Scheme fall `sqall`, sem tekur lista  $(x_1 \dots x_n)$ , sem viðfang, og skilar listanum  $(x_1^2 \dots x_n^2)$ . Til dæmis skal kallið `(sqall (list 1 2 3 4))` skila listanum `(1 4 9 16)`.

*Lausn:*

```
;; Notkun: (sqall x)
;; Fyrir:  x=(x1 ... xN) er listi talna.
;; Gildi:  Listinn (x1^2 ... xN^2)
(define (sqall x)
  ;; Notkun: (mymap f x)
  ;; Fyrir:  f er fall og x er listi
  ;; Gildi:  Listinn ((f x1) ... (f xN))
  (define (mymap f x) (if (null? x) '() (cons (f (car x)) (mymap f (cdr x)))))

  ;; Notkun: (sqr y)
```

```
;; Fyrir: y er tala
;; Gildi: talan  $y^2$ 
(define (sqr y) (* y y))

(mymap sqr x)
)
```

### 3. Dæmi

---

Skrifið halaendurkvæmt Scheme fall `sqallrev`, sem tekur lista  $(x_1 \dots x_n)$ , sem viðfang, og skilar listanum  $(x_n^2 \dots x_1^2)$ . Til dæmis skal kallið `((myif 1 2) #t)` skila 1, en `((myif 1 2) #f)` skal skila 2.

*Lausn:*

```
;; Notkun: (sqallrev x)
;; Fyrir: x=(x1 ... xN) er listi talna.
;; Gildi: Listinn (xN^2 ... x1^2)
(define (sqallrev x)
  ;; Notkun: (revconcatsqr a b)
  ;; Fyrir: a=(...) er listi talna
  ;; b=(x1 ... xN) er listi talna.
  ;; Gildi: Listinn (xN^2 ... x1^2 ... allar tölur í a)
  (define (revconcatsqr a b)
    (if (null? b)
        a
        (revconcatsqr (cons (* (car b) (car b)) a) (cdr b))))
  (revconcatsqr '() x)
)
```

### 4. Dæmi

---

Skrifið Scheme fall `myif`, sem tekur tvö viðföng  $x$  og  $y$ , og skilar falli, sem tekur gildi  $z$ , sem viðfang, og skilar  $x$  ef  $z$  er satt og  $y$  ef  $z$  er ekki satt. Til dæmis skal kallið `((myif 1 2) #t)` skila 1, en `((myif 1 2) #f)` skal 2.

*Lausn:*

```
;; Notkun: (myif a b)
;; Fyrir:  a, b af hvaða tagi sem er
;; Gildi:  Fall af x sem skilar a ef x er satt annars b
(define (myif a b)
  (lambda (x) (if x a b))
)
```

## Viðauki

Áður en við byrjuðum að vinna hvert dæmi skrifuðum við prufunarkóða fyrir fallið (*test driven development*). Lausnirnar okkar voru svo athugaðar út frá prufunarkóðanum í hvert skiptið sem við gerðum breytingar. Sjá kóða:

```
(require rackunit "vika2.scm")

; Define helper functions for test
(define (plus5 x) (+ 5 x))
(define (times3 x) (* 3 x))

(check-equal? (plus5 5) 10)
(check-equal? (times3 3) 9)

; Test fcompose
(check-equal? ((fcompose plus5 times3) 5) 20)

; Test sqall
(check-equal? (sqall '(1 2 3 4)) '(1 4 9 16))

; Test sqallrev
(check-equal? (sqallrev '(1 2 3 4)) '(16 9 4 1))

; Test myif
(check-equal? ((myif 1 2) #t) 1)
(check-equal? ((myif 1 2) #f) 2)
```

---

```
$ raco test

raco test: "./vika2/vika2-test.rkt"
7 tests passed
```