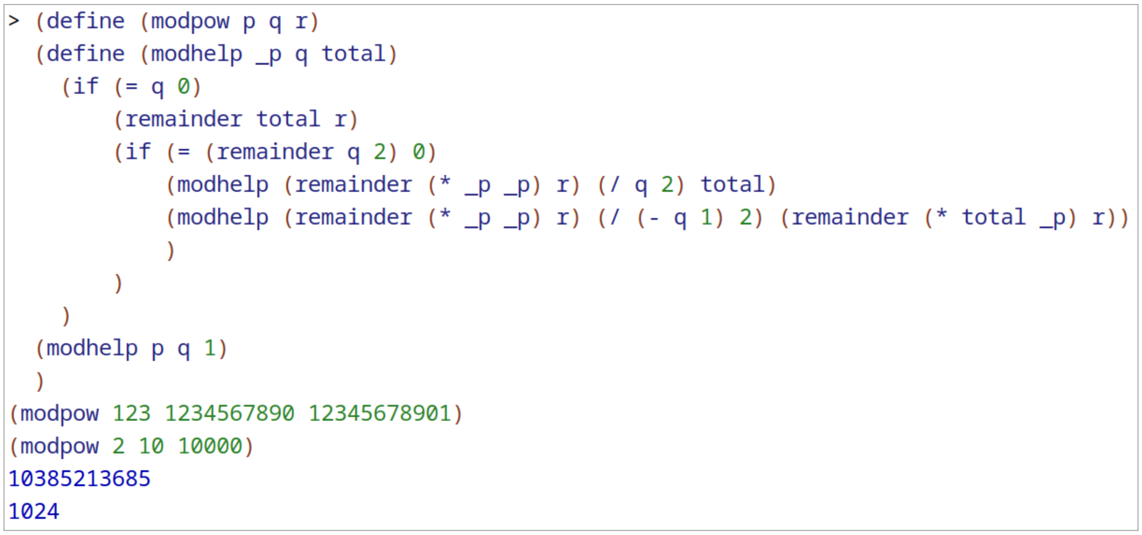
Forritunarmál Hópverkefni 5

Ragnar Björn Ingvarsson, rbi3
Daníel Snær Halldórsson, dsh11
Ólafur Sær Sigursteinsson, oss27
Máni Sverrisson, mas176
Elías Ver Bjarnason, evb17
Jonathan Jakub Otuoma, jjo1
Yi Hu, yih2
Dagmar Ýr Eyþórsdóttir, dye2
Eygló Ástþórsdóttir, eya19
Ana Margarida Delgado Costa, amd16
Kristín Fríða Sigurborgardóttir, kfs14

26. september 2024

1

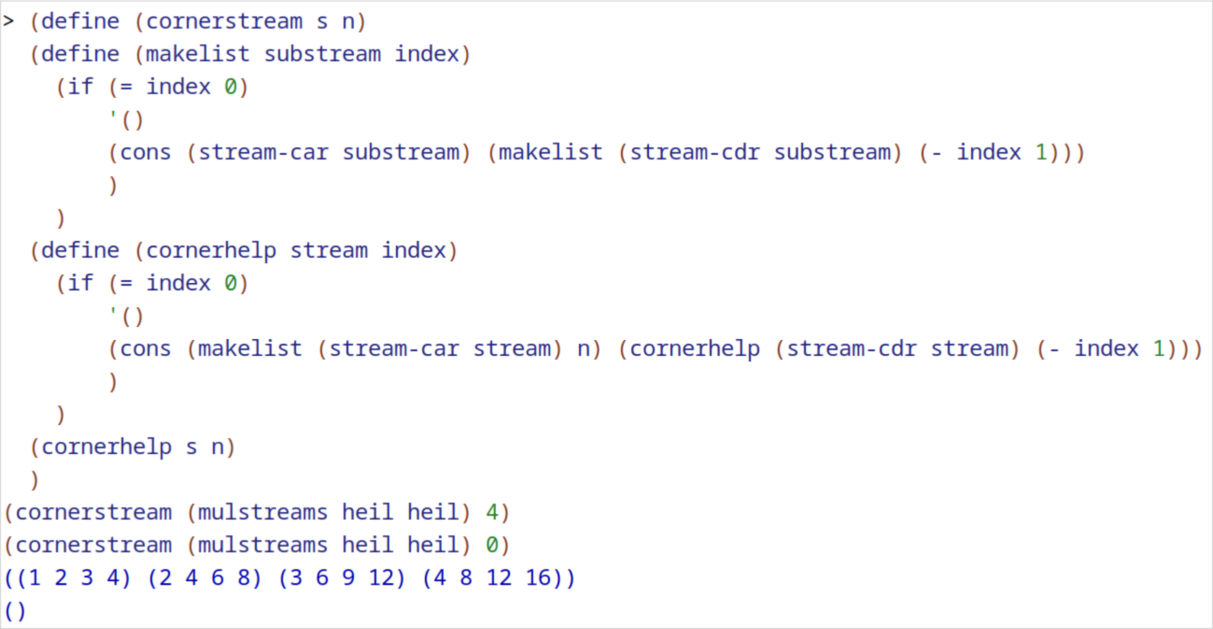
```
;; Notkun: (modpow p q r)
;; Fyrir: p, q og r eru heiltölur og q >= 0, 0 <= p < r
;;        og r > 1.
;; Gildi: p^q mod r, semsagt afgangur p^q þegar deilt
;;        með r.
(define (modpow p q r)
  ;; Notkun: (modhelp _p q total)
  ;; Fyrir: _p, q og total eru heiltölur,
  ;;        0 <= _p < r, q >= 0, 0 <= total < r
  ;; Gildi: (total * (_p^q mod r)) mod r
  (define (modhelp _p q total)
    (if (= q 0)
        (remainder total r)
        (if (= (remainder q 2) 0)
            (modhelp (remainder (* _p _p) r) (/ q 2) total)
            (modhelp (remainder (* _p _p) r) (/ (- q 1) 2) (remainder (* total _p) r))
            )
        )
    )
  (modhelp p q 1)
  )
```



```
> (define (modpow p q r)
  (define (modhelp _p q total)
    (if (= q 0)
        (remainder total r)
        (if (= (remainder q 2) 0)
            (modhelp (remainder (* _p _p) r) (/ q 2) total)
            (modhelp (remainder (* _p _p) r) (/ (- q 1) 2) (remainder (* total _p) r))
            )
        )
    )
  (modhelp p q 1)
  )
(modpow 123 1234567890 12345678901)
(modpow 2 10 10000)
10385213685
1024
```

2

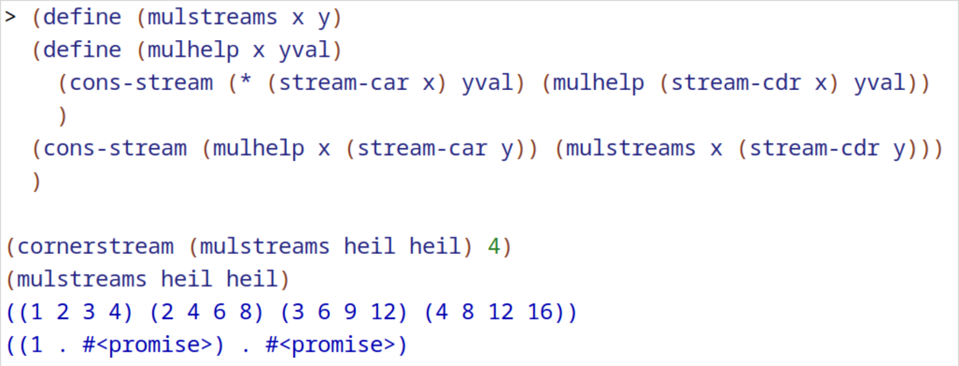
```
;; Notkun: (cornerstream s n)
;; Fyrir: s er óendanlegur straumur óendanlegra
;;        strauma,
;;        s=[[x11 x12 ...],[x21 x22 ...] ...].
;;        n er heiltala, n>=0.
;; Gildi: Listinn
;;        ((x11 x12 ... x1n)
;;        (x21 x22 ... x2n)
;;        ...
;;        (xn1 xn2 ... xnn)
;;        )
(define (cornerstream s n)
  ;; Notkun: (makelist substream index)
  ;; Fyrir: substream er óendanlegur straumur, [x1, x2, ...],
  ;;        index er heiltala, 0 <= index <= n
  ;; Gildi: Listinn (x1, x2, ... , x_index)
  (define (makelist substream index)
    (if (= index 0)
        '()
        (cons (stream-car substream) (makelist (stream-cdr substream) (- index 1)))
        )
    )
  ;; Notkun: (cornerhelp stream index)
  ;; Fyrir: stream er óendanlegur straumur óendanlegra
  ;;        strauma [[x11, x12, ...], [x21, x22, ...], ...],
  ;;        index er heiltala, 0 <= index <= n
  ;; Gildi: Listinn ((x11, x12, ..., x1n),
  ;;                 (x21, x22, ..., x2n),
  ;;                 ...
  ;;                 (x_index1, x_index2, ..., x_indexn))
  (define (cornerhelp stream index)
    (if (= index 0)
        '()
        (cons (makelist (stream-car stream) n) (cornerhelp (stream-cdr stream) (- index 1)))
        )
    )
  (cornerhelp s n)
  )
```



```
> (define (cornerstream s n)
  (define (makelist substream index)
    (if (= index 0)
        '()
        (cons (stream-car substream) (makelist (stream-cdr substream) (- index 1)))
        )
    )
  (define (cornerhelp stream index)
    (if (= index 0)
        '()
        (cons (makelist (stream-car stream) n) (cornerhelp (stream-cdr stream) (- index 1)))
        )
    )
  (cornerhelp s n)
  )
(cornerstream (mulstreams heil heil) 4)
(cornerstream (mulstreams heil heil) 0)
((1 2 3 4) (2 4 6 8) (3 6 9 12) (4 8 12 16))
()
```

3

```
;; Notkun: (mulstreams x y)
;; Fyrir: x og y eru óendanlegir straumar talna,
;; x=[x1 x2 x3 ...].
;; y=[y1 y2 y3 ...].
;; Gildi: Óendanlegur straumur óendanlegra strauma
;; talna sem er
;; [[x1*y1 x2*y1 x3*y1 ...]
;; [x1*y2 x2*y2 x3*y2 ...]
;; [x1*y3 x2*y3 x3*y3 ...]
;; ...
;; ]
(define (mulstreams x y)
  ;; Notkun: (mulhelp x yval)
  ;; Fyrir: x er óendanlegur straumur talna, [x1 x2 x3 ...],
  ;;        yval er tala
  ;; Gildi: óendanlegi straumurinn [x1*yval x2*yval ...]
  (define (mulhelp x yval)
    (cons-stream (* (stream-car x) yval) (mulhelp (stream-cdr x) yval))
    )
  (cons-stream (mulhelp x (stream-car y)) (mulstreams x (stream-cdr y)))
  )
```

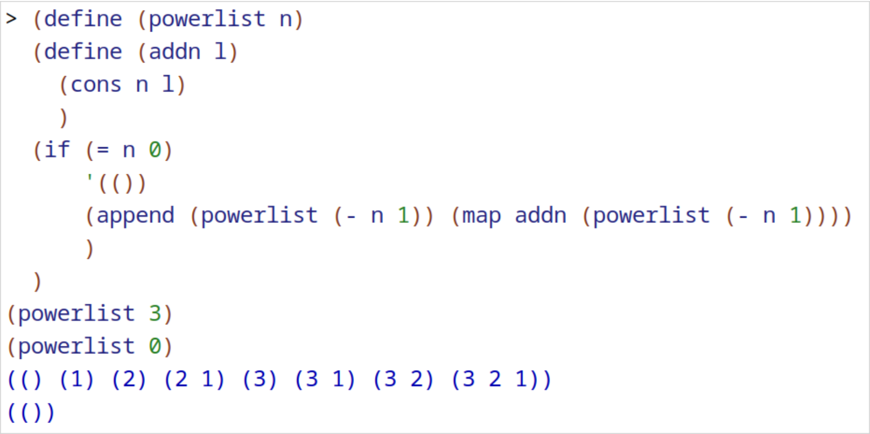


```
> (define (mulstreams x y)
  (define (mulhelp x yval)
    (cons-stream (* (stream-car x) yval) (mulhelp (stream-cdr x) yval))
    )
  (cons-stream (mulhelp x (stream-car y)) (mulstreams x (stream-cdr y)))
  )

(cornerstream (mulstreams heil heil) 4)
(mulstreams heil heil)
((1 2 3 4) (2 4 6 8) (3 6 9 12) (4 8 12 16))
((1 . #<promise>) . #<promise>)
```

4

```
;; Notkun: (powerlist n)
;; Fyrir: n er heiltala, n>=0.
;; Gildi: Listinn (y1 y2 y3 ...)
;;        sem inniheldur alla lista sem
;;        hægt er að smíða með því að taka
;;        núll eða fleiri gildi úr {1,...,n}
;;        og skeyta þeim saman í lista í
;;        minnkandi röð.
(define (powerlist n)
  ;; Notkun: (addn l)
  ;; Fyrir: l er listi, l = (l1 l2 ... lm)
  ;; Gildi: Listinn (n l1 l2 ... lm)
  (define (addn l)
    (cons n l)
    )
  (if (= n 0)
      '(())
      (append (powerlist (- n 1)) (map addn (powerlist (- n 1))))
      )
  )
```



```
> (define (powerlist n)
  (define (addn l)
    (cons n l)
    )
  (if (= n 0)
      '(())
      (append (powerlist (- n 1)) (map addn (powerlist (- n 1))))
      )
  )
(powerlist 3)
(powerlist 0)
(() (1) (2) (2 1) (3) (3 1) (3 2) (3 2 1))
(())
```