

## TÖL304G – Forritunarmál – Verkefni 5

1. Skrifðu halaendurkvæmt Scheme fall sem samsvarar Dafny fallinu `RealPowLoop` hér! . Skylda er að sjá til þess að dýpt endurkvæmni sé í mesta lagi í hlutfalli við  $\log_2(x)$ , alls ekki í hlutfalli við  $x$ , þegar reiknað er  $z^x$ .

Þið munuð væntanlega vilja nota Scheme föllin `remainder` og `quotient`. Fyrir heiltölur  $x$  og  $y$  þannig að  $x, y > 0$  gildir að `(remainder x y)` skilar afgangnum þegar  $x$  er deilt með  $y$ , og `(quotient x y)` skilar útkomunni úr heiltöludeilingu á  $x$  með  $y$ . Prófið þessi föll í Scheme til að sjá hverju þau skila og berið `quotient` saman við `(/ x y)`.

Lýsing fallsins skal vera næganlega skýr til að glöggur lesandi geti sannfært sig um að virknin sé sönnuð án þess að bæta þurfi við forskilyrðum eða eftirskilyrðum til þess að sönnunin gangi upp. Þið megið reikna með því að glöggur lesandi viti að  $(x^2)^z = x^{2z}$ .

Prófið fallið með því að hefja töluna  $1 + 2^{-30}$  í veldið  $2^{30}$  og sýnið hvernig þið prófið og útkomuna úr prófinu. Þið skuluð sjá til þess að talan sem hafin er í veldi sé fleytitala frekar en ræð tala.

```
1 ;; Notkun: (real-pow-loop z x)
2 ;; Fyrir: z er kommutala og x er heiltala, x >= 0.
3 ;; z er gagnagrunnurinn og x er veldið sem z er reiknað.
4 ;;
5 ;; Gildi: Talan sem er z reiknar til veldis x, þ.e. z^x.
6 ;; Útfærð með endurkvæmni sem byggir á tvöföldunaraðferðinni,
7 ;; sem gerir reikninginn mörgum sinnum hraðari en
8 ;; beinn veldisreikningur.
9
10 |
11 (define (real-pow-loop z x)
12   (define (helper z x acc)
13     (cond ((= x 0) acc)
14           ((even? x) (helper (* z z) (quotient x 2) acc))
15           (else (helper z (- x 1) (* z acc)))))
16   (helper (exact->inexact z) x 1.0))
17
18 (define result (real-pow-loop (+ 1 (/ 1.0 (expt 2 30))) (expt 2 30)))
19 (display result)
20 (newline)
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Welcome to DrRacket, version 8.10 [cs].  
Language: R5RS; memory limit: 128 MB.  
718281808182473

2. Skriðið fall `transpose-list` með eftirfarandi lýsingu.

Write a function `transpose-list` with the following description.

```
;; Notkun: (transpose-list z)
;; Fyrir:  z er listi jafnlangra lista,
;;         z=((x11 x12 ... x1N)
;;            (x21 x22 ... x2N)
;;            (x31 x32 ... x3N)
;;            .
;;            .
;;            .
;;            (xM1 xM2 ... xMN)
;;         )
;; Gildi:  Listinn sem er byltingin
;;         (transpose) af z, þ.e.
;;         ((x11 x21 ... xM1)
;;          (x12 x22 ... xM2)
;;          (x13 x23 ... xM3)
;;          .
;;          .
;;          .
;;          (x1N x2N ... xMN))
```

Athugið að fallið þarf ekki að virka ef innri listarnir eru ekki jafnlangir því samkvæmt forskilyrði á slíkt ekki að gerast. Ef innri listarnir eru tómir er eðlilegt að fallið skili tóma listanum. Sama gildir ef ytri listinn er tómur. Til hliðsjónar má kíkja á Dafny útfærslu á vefnum<sup>3</sup> eða í skránni `Analogues.dfy` í Canvas. Munið að sýna prófanir.

```
1 (define (transpose-list z)
2   (if (or (null? z) (null? (car z)))
3       '()
4       (cons (map car z) (transpose-list (map cdr z)))))
5
6 (define test-list-1
7   '((1 2 3)
8     (4 5 6)
9     (7 8 9)))
10
11 (define test-list-2
12   '((1 4 7)
13     (2 5 8)
14     (3 6 9)))
15
16 (display (transpose-list test-list-1))
17 (newline) ; Gefur ((1 4 7) (2 5 8) (3 6 9))
18
19 (display (transpose-list test-list-2))
20 (newline) ; Gefur ((1 2 3) (4 5 6) (7 8 9))
21
22 (display (transpose-list '()))
23 (newline) ; Gefur ()
24
25 (display (transpose-list '((1) (2) (3))))
26 (newline) ; Gefur ((1 2 3))
27
Welcome to DrRacket, version 8.10 [cs].
Language: R5RS; memory limit: 128 MB.
((1 4 7) (2 5 8) (3 6 9))
((1 2 3) (4 5 6) (7 8 9))
()
((1 2 3))
> |
```