TÖL304G
Forritunarmál
Verkefnablað 2

Snorri Agnarsson

28. ágúst 2024

Efnisyfirlit

1 Inngangur

Þið eigið að prófa öll Scheme verkefnin ykkar í einhverju Scheme kerfi, til dæmis DrRacket og sýna útkomur prófana.

Hér eru nokkur Scheme föll og lykilorð sem mögulegt er að þið viljið nota til að leysa þessi verkefni: define, lambda, if, and, or, car, cdr, cons, null?, list, =, *, +.

Athugið að Scheme report[1] inniheldur lýsingar á öllum þessum föllum og lykilorðum og einnig má finna nákvæma skjölun í fylgigögnum fyrir MIT-Scheme og DrRacket.

[1] http://cs.hi.is/snorri/downloads/r5rs.pdf

2 Hópverkefni

Í eftirfarandi verkefnum megið þið einungis nota einföldu innbyggðu föllin `car, cons, cdr, null?, list` og `*` auk lykilorðanna `lambda, define` og `if`. Það ætti ekki að valda vandræðum. Einnig má nota hvaða lesfasta (*literal*) sem verða vill, svo sem `'()` og talnafasta. Þið megið að sjálfsögðu kalla á föllin sem þið skrifið og skilið.

1. Skrifið Scheme fall `last` sem tekur lista sem viðfang, sem ekki má vera tómur, og skilar aftasta gildinu í listanum. Til dæmis skulu segðirnar `(last '(1 2 3))` og `(last (list 1 2 3))` skila `3`.

   ```
   ;; Notkun: (last x)
   ;; Fyrir:  x=(x1 x2 ... xN) er listi, ekki tómur.
   ;; Gildi:  xN, þ.e. aftasta gildi x.
   (define (last x)
      ...
   )
   ```

2. Skrifið Scheme fall `remove-last` sem tekur lista sem viðfang, sem ekki má vera tómur, og skilar lista allra gilda nema aftasta í viðfangslistanum. Til dæmis skal segðin `(remove-last '(1 2 3))` skila `(1 2)`.

   ```
   ;; Notkun: (remove-last x)
   ;; Fyrir:  x=(x1 x2 ... xN) er listi, ekki tómur.
   ;; Gildi:  (x1 x2 ... xN-1), þ.e. listi allra
   ;;         gilda í x fyrir utan aftasta.
   (define (remove-last x)
      ...
   )
   ```

3. Skrifið Scheme fall `product`, sem tekur eitt viðfang x, sem skal vera listi talna $x_1, \ldots, x_n$, og skilar $\Pi_{i=1}^{n} x_i$. Þið skuluð leyfa að listinn sé tómur og skila viðeigandi gildi í því tilviki. Fallið skal vera halaendurkvæmt.[2]

   ```
   ;; Notkun: (product x)
   ;; Fyrir:  x=(x1 x2 ... xN) er listi talna.
   ;; Gildi:  Talan x1*x2*...*xN.
   (define (product x)
      ...
   )
   ```

[2] Það dugar að útreikningarnir séu framkvæmdir af halaendurkvæmu hjálparfalli, jafnvel þótt `product` sé ekki beint halaendurkvæmt. Markmiðið er að takmarka dýpt hlaðans sem forritið notar fyrir millniðurstöður, sem við munum sjá að er hlaði svokallaðra vakningarfærslna (*activation records*).

4. Skrifið Scheme fall `myappend` sem tekur tvo lista, `x` og `y` sem viðföng, og skilar lista sem inniheldur fremst öll gildin úr `x` (í sömu röð og í `x`) og síðan öll gildin úr `y` (í sömu röð og í `y`). Fallið `myappend` skal útfæra með því að nota föllin `last` og `remove-last`, að ofan, og einnig má nota `define`, `if`, `cons`, `null?`, en ekki önnur föll eða lykilorð. Fallið `myappend` verður eðlilega halaendurkvæmt, þannig að oftast þegar kallað er á það mun það enda á að kalla á sjálft sig. Tímaflækja þessa falls er hins vegar ekkert til að hrópa húrra fyrir. Athugið líka að þótt `myappend` sé halaendurkvæmt þá er næstum öruggt að hjálparfallið `remove-last` verður trúlega ekki halaendurkvæmt þannig að heildarlausnin er þá ekki halaendurkvæm.

```
;; Notkun: (myappend x y)
;; Fyrir:  x=(x1 x2 ... xN) er listi,
;;         y=(y1 y2 ... yM) er einnig listi.
;; Gildi:  Listinn (x1 x2 ... xN y1 y2 ... yM).
(define (myappend x)
   ...
}
```

3 Dafny

Það verða engar spurningar um forritunarmálið Dafny á prófi en við munum hafa Dafny verkefni. Hér er dæmi um Dafny forrit sem sannar með þrepun að fyrir heiltölur $n \geq 0$ gildir

$$\left(\sum_{i=1}^{n} i\right)^2 = \sum_{i=1}^{n} i^3$$

```
// Computes n^2, the square of the argument n.
function Square( n: int ): int
{
  n*n
}

// For a given n>=0 computes 1+2+...+n.
function SumInts( n: int ): int
  requires n >= 0
  decreases n
{
  if n == 0 then 0 else SumInts(n-1)+n
}

// For a given n>=0 computes 1^3+2^3+3^3+...+n^3
// and ensures that this is equal to the value
// (1+2+3+...+n)^2.
function SumCubes( n: int ): int
  requires n >= 0
  decreases n
  ensures SumCubes(n) == Square(SumInts(n))
{
  if n == 0 then 0 else SumCubes(n-1)+n*n*n
}
```

Þetta má einnig finna á vefnum[3].

Dafny þýðandinn samþykkir þennan forritstexta og það þýðir að eftirskilyrði falls-ins SumCubes eru uppfyllt að því skilyrði að forskilyrðin séu uppfyllt.

4 Einstaklingsverkefni

Klárið Dafny föllin þrjú sem eru ókláruð á þessari vefsíðu[4]. Skilið PDF útprenti af lausninni í Gradescope og skilið einnig (fremst í sama útprenti) permalink á lausn-ina ykkar. Þið getið fengið permalink á lausnina, þegar hún er tilbúin, með því að styðja á hnappinn sem merktur er með keðju, sækja of langa permalinkinn þar og nota tinyurl.com[5] til að smíða styttri, nothæfan, permalink.

Til hliðsjónar getið þið kíkt á þessa vefsíðu[6], sem við munum kíkja á í fyrirlestri.

[3] https://tinyurl.com/cwhpycau
[4] https://tinyurl.com/ydusfckf
[5] https://tinyurl.com
[6] https://tinyurl.com/y23c9ku3