TÖL304G
Forritunarmál
Verkefnablað 5

Snorri Agnarsson

23. september 2019

Verkefni

Munið að föll sem skilað er, þar með talið hjálparföll, hafa skýra og rétta lýsingu með „Notkun: ...“, „Fyrir: ...“ og „Gildi: ...“. Takið eftir að í sumum tilfellum þurfa forskilyrði að innihalda lýsingar á sama sniði fyrir viðföng sem eru föll, og svipað gildir í eftirskilyrði (þ.e. „Gildi: ...“) fyrir gildi sem eru föll.

Mikilvægt er að:

- Allar lausnir séu vel sniðsettar og *með réttri innfellingu*.
- Öll **föll** hafi skýra lýsingu með „Notkun:“, „Fyrir:“ og annaðhvort „Eftir:“ eða „Gildi:“, þ.e. forskilyrði og eftirskilyrði. Sé lýsingin gefin fyrirfram skal nota hana, en athugið að hjálparföll sem þið finnið sjálf upp þurfa sína lýsingu. Athugið samt að þótt gefin sé lýsing falls má skrifa aðra lýsingu ef forskilyrði er víkkað og eftirskilyrði er þrengt. Það kemur fyrir sð slíkt sé gagnlegt, til dæmis ef endurkvæm notkun fallsins krefst betra falls en upphaflega lýsingin krefst.
- Sérhver nemandi skal vinna sín einstaklingsverkefni einn og óháður öðrum nemendum. Samræður milli nemenda um verkefnin eru af hinu góða, en afritun einstaklingsverkefna er að sjálfsögðu óheimil.

Í öllum Scheme verkefnunum er ætlast til að forritunin sé án hliðarverkana nema annað sé tekið fram. Þetta þýðir einfaldlega að ekki er leyfilegt að gefa breytu nýtt gildi eða gera uppskurð á lista. Ef þið notið aldrei neina aðgerð í Scheme sem hefur táknið ’!’ í sínu nafni þá munuð þið uppfylla þetta skilyrði. Dæmi um slíka forboðna aðgerð er `set!`, sem notuð er til að gefa breytu nýtt gildi.

Hópverkefni

1. Skrifið fall `mulstreams` með eftirfarandi lýsingu:

```
;; Notkun: (mulstreams x y)
;; Fyrir:  x er óendanlegur straumur talna,
;;         x=[x1 x2 x3 ...].
;;         y er einnig óendanlegur straumur talna,
;;         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 ...]
;;           .
;;           .
;;           .
;;         ]
```

   Prófið fallið `mulstreams` með segðinni `(mulstreams heil heil)` og sýnið hluta af útkomunni. Útkoman er óendanleg margföldunartafla þannig að vonlaust er að sjá hana alla, en athugið að ef breytan `s` inniheldur óendanlegan straum af óendanlegum straumum þá mun segðin

```
(map (lambda (x) (stream-list x 5)) (stream-list s 5))
```

   skila lista af listum sem eru hornið á fylkinu. Slíkt gildi getur Racket birt okkur á læsilegan hátt.

   Vísbending: Stofninn á fallinu `mulstreams` verður eðlilega ein beiting á `cons-stream`. Spyrjið ykkur hver hausinn er á útkomunni og hvernig hægt er að reikna hann (það má gera með einu kalli á fall sem finna má í `straumar.rkt`) og hver halinn er og hvernig hægt er að reikna hann.

2. Skrifið fall `squarestream` með eftirfarandi lýsingu:

```
;; Notkun: (squarestream s)
;; Fyrir:  s er óendanlegur straumur talna,
;;         s=[x1 x2 x3 ...].
;; Gildi:  Óendanlegur straumur óendanlegra strauma
;;         talna sem er
;;         [[x1*x1 x2*x1 x3*x1 ...]
;;          [x1*x2 x2*x2 x3*x2 ...]
;;          [x1*x3 x2*x3 x3*x3 ...]
;;           .
```

```
;;            .
;;            .
;;          ]
```

Prófið fallið `squarestream` á straumnum `heil` og sýnið hluta af útkomunni.

Athugið að þægileg aðferð til að þróa föll sem byggð eru á straumaföllunum er að sækja forritstextann fyrir strauma úr Uglunni og bæta ykkar falli aftast í skrána.

3. Skrifið fall `powerlist` sem hefur eftirfarandi lýsingu:

```
;; Notkun: (powerlist x)
;; Fyrir:  x er endanlegur listi,
;;         x=(x1 x2 ... xN)
;; Gildi:  Listinn (y1 y2 y3 ...)
;;         sem inniheldur alla lista sem
;;         hægt er að smíða með því að taka
;;         núll eða fleiri gildi úr x, í
;;         sömu röð og í x, og skeyta þeim
;;         saman í lista.
```

Athugið að þetta er svipað og að reikna veldismengi mengis. Röð listanna í útkomulistanum er valfrjáls. Til dæmis myndi `(powerlist '())` skila listanum `(())` (sem samsvarar því að veldismengið af tómamenginu er mengið sem inniheldur aðeins tómamengið. Segðin `(powerlist '(1))` gæti skilað listanum `(() (1))` eða listanum `((1) ())`. Segðin `(powerlist '(1 2))` gæti skilað listanum `(() (1) (2) (1 2))`.

Fjöldi staka (innri lista) í útkomulistanum ætti ávallt að vera 2^N þar sem N er fjöldi gilda í viðfanginu x.

Innbyggðu föllin `map` og `append` eru gagnleg hér. Þetta verkefni er hægt að leysa á tiltölulega einfaldan hátt í Scheme.

Einstaklingsverkefni

1. Skrifið endurkvæmt Scheme fall sem samsvarar Dafny fallinu `RealPow_recursive` hér[1]. 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.

   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

[1] https://rise4fun.com/Dafny/SEz8

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 titl þess að sönnunin gangi upp. Þið megið reikna með því að glöggur lesandi viti að $\left(x^2\right)^z = x^{2z}$.

Prófið fallið með því að hefja töluna $1 + 10^{-10}$ í veldið 10^10 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.

2. Skrifið halaendurkvæmt Scheme fall sem samsvarar Dafny fallinu `RealPow_loop` hér[2]. 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.

   Þið munið þurfa halaendurkvæmt hjálparfall til að líkja eftir lykkunni.

   Lýsing fallanna skal vera næganlega skýr til að glöggur lesandi geti sannfært sig um að virknin sé sönnuð.

   Prófið fallið með því að hefja töluna $1 + 10^{-10}$ í veldið 10^{10} 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. Íhugið hvers vegna.

3. Skrifið fall `byltalista` með eftirfarandi lýsingu:

```
;; Notkun: (byltalista 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)
;;          .
```

[2] https://rise4fun.com/Dafny/SEz8

```
;;            .
;;            .
;;            (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. Munið að sýna prófanir.