

TÖL304G

Forritunarmál

Verkefnablað 7

Snorri Agnarsson

6. október 2019

Verkefni

Lýsingar falla

Í öllum verkefnunum skulu öll föll, 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.

Prófanir falla

Munið að alltaf skal sýna einhverjar ófáfengilegar prófanir á öllum föllum sem þið skilið. Ekki þarf að sýna skjáskot af prófunum, en sýna skal hvernig prófanirnar eru forritaðar og hverjar útkomurnar eru.

Gott getur verið að ræða um prófanir ykkar á milli og á Piazza, einnig fyrir einstaklingsverkefni.

Listi úr óendanlegum straumi

Eftirfarandi fall getur verið hjálplegt í prófunum.

```
;;; Notkun: x = take(n,s);  
;;; Fyrir:  n er heiltala >=0,  
;;;       s er óendanlegur straumur.  
;;; Eftir:  x er listi sem inniheldur n-1
```

```

;;;          fremstu gildi úr s í sömu röð
;;;          og í s.
rec fun take(n,s)
{
  if( n==0 ) {return []};
  return streamHead(s) : take(n-1,streamTail(s));
};

```

Hópverkefni

1. Skrifið fall `filter` í Morpho sem tekur tvö viðföng, fall p og lista $x = [x_1, \dots, x_n]$ og skilar lista þeirra gilda x_i í x þannig að kallið $p(x_i)$ skilar „satt“, þ.e. þannig að skilagildið er hvorki `null` né `false`. Til dæmis ætti segðin `filter(fun(x){x%2==0}, [1,2,3,4,5])` að skila listanum `[2,4]`.
2. Skrifið fall `filterStream` í Morpho sem tekur tvö viðföng, fall p og óendanlegan straum $x = [x_1, \dots]$ og skilar straumi þeirra gilda x_i í x þannig að kallið $p(x_i)$ skilar „satt“, þ.e. þannig að skilagildið er hvorki `null` né `false`.
3. Skrifið endurkvæma skilgreiningu á fasta í Morpho

```

;;; Notkun: primes
;;; Fyrir:   Ekkert
;;; Gildi:   Straumur allra prímtalna án
;;;          endurtekninga í vaxandi röð,
;;;          þ.e. #[2,3,5,7,11,13,17,...].
rec val primes=...;

```

þannig að breytan `primes` innihaldi vaxandi óendanlegan straum sem inniheldur allar prímtölurnar og ekkert annað, þ.e. `#[2,3,5,7,11,13,17,...]`. Þið megið nota fallið `filterStream` hér að ofan og önnur hjálparföll að vild (með réttum lýsingum, að sjálfsögðu).

Vísbending 0: Til þess að allt þetta gangi snurðulaust fyrir sig er ráðlegt að sjá til þess að allar heiltölurnar í straumunum séu *stórar heiltölur*, þ.e. heiltölur sem búið er að breyta á `BigInteger` snið. Stóra heiltalan sem samsvarar `12345` er `bigInteger(12345)`. Reikniaðgerðir á stórar heiltölur skila stórum heiltölum. Ef þetta er ekki gert þá munu straumarnir byrja á réttum gildum en fyrr eða síðar munu koma röng gildi, því réttu gildin þurfa að vera stórar heiltölur.

Vísbending 1: Straumurinn `#[2,3,5,7,9,11,13,15,17,...]` getur verið gagnlegur til að byggja á. Ef þið síið úr þeim straumi einungis þær tölur

þar sem engin fremri tala gengur upp í töluna þá fáið þið straum allra prímtalna. Sama má einnig gera með strauminn $\#[2, 3, 4, 5, 6, 7, 8, \dots]$, sem er e.t.v. aðeins einfaldara.

Vísbending 2: Það getur verið hjálplegt að skrifa fyrst hjálparfall með eftirfarandi lýsingu:

```
;;; Notkun: p = erPrím(s,n);  
;;; Fyrir:  n er heiltala >1, s er óendanlegur  
;;;        vaxandi straumur mismunandi stórra  
;;;        heiltalna >1 sem inniheldur allar  
;;;        prímtölur sem ganga upp í n.  
;;; Eftir:  p er satt þá og því aðeins að n  
;;;        sé prímtala.
```

Vísbending 3: Eðlilegt er að útfæra `erPrím` með `lykkju` (ásamt `fastayrðingu` `lykkju`), eða (þá þarf ekki `fastayrðingu` `lykkju`) með `halaendurkvæmni`. Í báðum tilvikum stytum við `s` í hverju skrefi.

Vísbending 4: Til þess að gera útreikningana hraðvirkari og einnig til að sjá til þess að möguleg gagnkvæm endurkvæmni lendi ekki í vandræðum þá er gagnlegt að athuga, inni í útfærslunni á `erPrím`, hvort `streamHead(s)*streamHead(s) > n`. Ef svo er þá er `n` áreiðanlega prímtala (íhugið hvers vegna það er satt og íhugið einnig hvers vegna þetta hraðar útreikningunum og hve mikið).

Einstaklingsverkefni

1. Skrifið fall `powerList` sem tekur heiltölu $n \geq 0$ sem viðfang og skilar lista af listum heiltalna sem inniheldur alla mögulega lista heiltalna sem eru undirlistar listans $x = [n, n-1, \dots, 2, 1]$ í þeim skilningi að undirlistarnir innihalda gildi úr listanum x í sömu röð og í x , nema hvað fjarlægja má núll eða fleiri gildi úr listanum x .

Til dæmis myndi `powerList(0)` skila `[[]]` og `powerList(2)` gæti skilað `[[] , [2] , [1] , [2, 1]]`, eða lista sömu lista í annarri röð.

Þið megið nota eins mörg hjálparföll eins og þið viljið og nota hvaða innbyggð föll í `Morpho` sem ykkur hentar.

2. Skrifið fall `zipWith` í `Morpho` sem tekur fall f (lokun) og tvo jafnlanga lista, $x = [x_1, \dots, x_n]$ og $y = [y_1, \dots, y_n]$ sem viðföng og skilar listanum $[f(x_1, y_1), \dots, f(x_n, y_n)]$. Til dæmis ætti kallið `zipWith(fun(x,y){x+y}, [1,2,3], [4,5,6])` að skila `[5,7,9]`.