Lengoaiak, Konputazioa eta Sistema Adimendunak

Kudeaketaren eta Informazio Sistemen Informatikaren Ingeniaritzako Gradua
Bilboko Ingeniaritza Eskola (UPV/EHU)
Lengoaia eta Sistema Informatikoak Saila
2. maila — 2017-18 ikasturtea
5. eta 7. gaiak: Konputazioaren konplexutasuna eta Haskell
1,6 puntu

2017-11-08

1 Murgilketa (0,300 puntu)

Datu bezala zenbaki osozko zerrendez eratutako s zerrenda bat emanda, lehenengo elementua s-ko aurreko zerrendetan ez duten zerrendez eratutako zerrenda itzuliko duen $lehenengoa\_aurrekoetan\_ez$ izeneko funtzioa definitu nahi da Haskellez. s zerrendan zerrenda hutsa agertzen bada, errore-mezua aurkeztu beharko da. s zerrenda hutsa baldin bada, zerrenda hutsa itzuli beharko da.

$$lehenengoa\_aurrekoetan\_ez :: [[Integer]] -> [[Integer]]$$
$$lehenengoa\_aurrekoetan\_ez\ \ s\ \ \ldots$$

Adibideak:

$lehenengoa\_aurrekoetan\_ez\ \ [[3,\underline{5},\underline{5}],[2,8],[\underline{5},9,1],[7,8,2]]\ = [[3,5,5],[2,8],[7,8,2]]$
$[5,9,1]$ desagertu da bere lehenengo elementua, hau da, 5, aurreko (ezkerrerago dagoen) zerrenda batean agertu delako.

$lehenengoa\_aurrekoetan\_ez\ \ [[3,\underline{5},\underline{6}],[\underline{6},3],[2,8],[\underline{5},9,1],[7,8,2]]\ = [[3,5,6],[2,8],[7,8,2]]$
$[6,3]$ eta $[5,9,1]$ desagertu dira beraien lehenengo elementuak, hau da, 6 eta 5, aurreko (ezkerrerago dagoen) zerrendaren batean agertu direlako.

$lehenengoa\_aurrekoetan\_ez\ \ [[3,\underline{5},\underline{6}],[\underline{6},3],[2,\underline{8}],[\underline{5},9,1],[\underline{8},7,2]]\ = [[3,5,6],[2,8]]$
$[6,3]$, $[5,9,1]$ eta $[8,7,2]$ desagertu dira beraien lehenengo elementuak, hau da, 6, 5 eta 8, aurreko (ezkerrerago dagoen) zerrendaren batean agertu direlako.

$lehenengoa\_aurrekoetan\_ez\ \ [[3,5,5],[2,8],[7,8,2]]\ = [[3,5,5],[2,8],[7,8,2]]$
Zerrenda denak mantendu dira lehenengo elementuak, hau da, 3, 2 eta 7 aurreko (ezkerrerago dagoen) zerrendaren batean ez direlako agertu.

$lehenengoa\_aurrekoetan\_ez$ funtzioa zuzenean definitu beharrean, **murgilketa**ren teknika erabiliz, jarraian zehazten diren ezaugarriak dituen $lehenengoa\_aurrekoetan\_ez\_lag$ izeneko funtzioa definitu behar da. Funtzio horrek, datu bezala, zenbaki osozkoak diren zerrendez osatutako s zerrenda eta zenbaki osozko r zerrenda jasoko ditu. Emaitza bezala, lehenengo elementua ez r zerrendan eta ez s-ko aurreko zerrendetan ez duten zerrendez eratutako zerrenda itzuliko du. s zerrendan zerrenda hutsa agertzen bada, errore-mezua aurkeztu beharko da. s zerrenda hutsa baldin bada, zerrenda hutsa itzuli beharko da.

$$lehenengoa\_aurrekoetan\_ez\_lag :: [[Integer]] -> [Integer] -> [[Integer]]$$
$$lehenengoa\_aurrekoetan\_ez\_lag\ \ s\ \ r\ \ \ldots$$

Adibideak:

$lehenengoa\_aurrekoetan\_ez\_lag\ \ [[3,\underline{5},\underline{5}],[2,8],[\underline{5},9,1],[7,8,2]]\ \ [\,]\ = [[3,5,5],[2,8],[7,8,2]]$
$[5,9,1]$ desagertu da bere lehenengo elementua, hau da, 5, aurreko (ezkerrerago dagoen) zerrenda batean agertu delako.

$lehenengoa\_aurrekoetan\_ez\_lag\ \ [[3,\underline{5},\underline{5}],[2,8],[\underline{5},9,1],[7,8,2]]\ \ [9,0,1,4]\ = [[3,5,5],[2,8],[7,8,2]]$
$[5,9,1]$ desagertu da bere lehenengo elementua, hau da, 5, aurreko (ezkerrerago dagoen) zerrenda batean agertu delako.

$lehenengoa\_aurrekoetan\_ez\_lag\ \ [[\underline{3},\underline{5},\underline{5}],[2,8],[\underline{5},9,1],[\underline{7},8,2]]\ \ [9,\underline{3},\underline{7},4]\ = [[2,8]]$
$[5,9,1]$ bere lehenengo elementua aurreko zerrenda batean agertzen delako desagertu da. $[3,5,5]$ eta $[7,8,2]$ beraien lehenengo elementuak $[9,3,7,4]$ zerrendan agertzen direlako desagertu dira.

$lehenengoa\_aurrekoetan\_ez\_lag$ funtzioa $lehenengoa\_aurrekoetan\_ez$ baino orokorragoa da; izan ere, s-ko zerrendak ezabatzerakoan kontuan hartu beharreko beste balio batzuk ipintzea ahalbidetzen du r parametroaren bidez. Parametro berri hori aurreko zerrenda denetako elementuak gordez joateko erabiliko da. Horrela, s-ko zerrenda bakoitzera iritsitakoan, aurretik zeuden zerrenda denetako elementuak eskura izango ditugu r parametroan.

Gainera, $lehenengoa\_aurrekoetan\_ez$ funtzioaren **konputazio-kostua aztertu** behar da.

2 Bukaerako errekurtsibitatea (0,300 puntu)

Har dezagun honako funtzio hau:

$$
\begin{array}{l}
posiziokoa\_ezabatu :: Integer \mathrel{->} [Integer] \mathrel{->} [Integer] \\
posiziokoa\_ezabatu\ \ n\ \ r \\
\quad \mid (n <= 0) \mid\mid (n > (genericLength\ \ r)) \ \ = \ \ error\ \ \text{“}posizioa\ ez\ da\ egokia.\text{”} \\
\quad \mid otherwise \qquad\qquad\qquad\quad = \ \ (genericTake\ \ (n-1)\ \ r)\ \ +\!+\ (genericDrop\ \ n\ \ r)
\end{array}
$$

$posiziokoa\_ezabatu$ funtzioak r zerrendako n-garren elementua ezabatuz lortuko den zerrenda itzuliko du, n-ren balioa egokia denean.

$posiziokoa\_ezabatu\ \ 2\ \ [9,\text{-}7,6,8]\ = [9,6,8]$ bigarren elementua kendu da.
$posiziokoa\_ezabatu\ \ 4\ \ [9,\text{-}7,6,8]\ = [9,\text{-}7,6]$ laugarren elementua kendu da.

$posiziokoa\_ezabatu$ funtzioak ez du bukaerako errekurtsibitaterik. Bukaerako errekurtsibitatea edukitzeko, **honako bi funtzio hauek definitu** behar dira:

- $posiziokoa\_ezabatu\_lag$ funtzioa: funtzio horrek $posiziokoa\_ezabatu$ funtzioak jasotzen dituen n zenbaki osoa eta r zenbaki osozko zerrendaz gain, zenbat elementu zeharkatu diren jakiteko balio duen z zenbaki osoa eta zeharkatutako elementuak gordez joateko ℓ zenbaki osozko zerrenda jasoko ditu parametro bezala. Formalki, $posiziokoa\_ezabatu\_lag$ funtzioak, ℓ zerrenda eta r zerrendako $n-z$-garren elementua ezabatu ondoren gelditzen den zerrenda elkartuz lortuko den zerrenda itzuliko du. Hala ere, hori egin ahal izateko, $n-z$ espresioaren balioak egokia izan beharko du, hau da, $n-z$ espresioak 1 baino handiagoa edo berdina eta r-ren luzera baino txikiagoa edo berdina izan beharko du. Bestela, errore-mezua aurkeztu beharko du.

  $posiziokoa\_ezabatu\_lag\ \ 2\ \ [9,\text{-}7,6,8]\ \ 0\ \ [\,]\ = [9,6,8]$
  $(2-0)$-garren elementua kendu da eta
  $[\,]\ +\!+\ [9,6,8]$ zerrenda itzuli da.

  $posiziokoa\_ezabatu\_lag\ \ 2\ \ [9,\text{-}7,6,8]\ \ 1\ \ [4,4,4]\ = [4,4,4,\text{-}7,6,8]$
  $(2-1)$-garren elementua kendu da eta
  $[4,4,4]\ +\!+\ [\text{-}7,6,8]$ zerrenda itzuli da.

- $posiziokoa\_ezabatu\_be$ funtzioa: funtzio horrek $posiziokoa\_ezabatu$ funtzioak egiten duen gauza bera egin beharko du $posiziokoa\_ezabatu\_lag$ funtzioari egokiak diren parametroekin deituz.

  $posiziokoa\_ezabatu\_be\ \ 2\ \ [9,\text{-}7,6,8]\ = [9,6,8]$
  Bigarrena ezabatu da.

Beraz, $posiziokoa\_ezabatu$ funtzioak egiten duena $posiziokoa\_ezabatu\_be$ eta $posiziokoa\_ezabatu\_lag$ funtzioak erabiliz egin ahal izango da.

Gainera, $posiziokoa\_ezabatu$ eta $posiziokoa\_ezabatu\_be$ funtzioen **konputazio-kostuak aztertu eta bata bestearekin alderatu** behar dira.

3 Zerrenda-eraketa (1,000 puntu)

3.1. (0,100 puntu) Zenbaki osoz eratutako zerrendez osatutako s zerrenda emanda, s-ko zerrenden luzerez eratutako zerrenda itzuliko duen $luzerak$ izeneko funtzioa definitu Haskell lengoaia erabiliz.

$$\begin{array}{l} luzerak :: [[Integer]] \text{ -> } [Integer] \\ luzerak\ \ s\ \ldots \end{array}$$

Adibideak:

$$\begin{array}{l} luzerak\ \ [[6,2,7],[8],[8],[\,],[3,3,5,3]]\ =\ [3,1,1,0,4] \\ luzerak\ \ [\,]\ =\ [\,] \end{array}$$

Aukera bat, aurredefinitutako $genericLength$ funtzioa erabiltzea da.

3.2. (0,100 puntu) Zenbaki osozko s zerrenda bat emanda, s-ko elementu guztiak berdinak al diren erabakiko duen $guztiak\_berdinak$ izeneko funtzioa definitu Haskellez. s hutsa baldin bada, $True$ balioa itzuli beharko da.

$$\begin{array}{l} guztiak\_berdinak :: [Integer] \text{ -> } Bool \\ guztiak\_berdinak\ \ s\ \ldots \end{array}$$

Adibideak:

$$\begin{array}{l} guztiak\_berdinak\ \ [5,5,5]\ =\ True \\ guztiak\_berdinak\ \ [5,2,5,6]\ =\ False \end{array}$$

Aukera bat, aurredefinitutako $null$, $genericLength$ eta $head$ funtzioak erabiltzea da.

3.3. (0,100 puntu) Osoa den n zenbaki bat eta zenbaki osozko zerrendez eratutako s zerrenda bat emanda, s-ko zerrenda bakoitzetik n-garren elementua hartuz osatzen den zerrenda itzuliko duen $posiziokoak\_hautatu$ izeneko funtzioa definitu Haskellez. n zero edo txikiagoa baldin bada, errore-mezua aurkeztu beharko da. Bestalde, s-ko zerrendaren batean elementu-kopurua n baino txikiagoa baldin bada, errore-mezua aurkeztu beharko da. s zerrenda hutsa baldin bada, zerrenda hutsa itzuli beharko da.

$$\begin{array}{l} posiziokoak\_hautatu :: Integer \text{ -> } [[Integer]] \text{ -> } [Integer] \\ posiziokoak\_hautatu\ \ n\ \ s\ \ldots \end{array}$$

Adibideak:

$$\begin{array}{l} posiziokoak\_hautatu\ \ 2\ \ [[6,\underline{4},7],[7,\underline{8}],[9,\underline{5},1],[4,\underline{5},8,8]]\ =\ [4,8,5,5] \\ posiziokoak\_hautatu\ \ 1\ \ [[\underline{6},4,7],[\underline{7},8],[\underline{9},5,1],[\underline{4},5,8,8]]\ =\ [6,7,9,4] \end{array}$$

Errore-kasurako, aurredefinitutako $genericLength$ eta 3.1 atalean definitutako $luzerak$ funtzioa erabil daitezke. Beste aukera bat, aurredefinitutako not, $null$ eta 3.1 atalean definitutako $luzerak$ funtzioa erabiltzea litzateke. Kasu orokorrerako, aurredefinitutako $genericDrop$ eta $head$ funtzioak erabiltzea da aukera bat.

3.4. (0,100 puntu) Zenbaki osozko zerrendez eratutako s zerrenda bat emanda, honako kalkulu hau burutuko duen $zutabeak$ izeneko funtzioa definitu Haskellez: s-ko zerrenda guztiek luzera bera ez badute, errore-mezua aurkeztu; bestalde, s-ko zerrenda guztiek luzera bera badute, zerrenda horietan posizio berean dauden elementuez osatutako zerrendez eratutako zerrenda itzuli; s hutsa bada edo s osatzen duten zerrenda guztiak hutsak badira, s bera itzuli.

$$\begin{array}{l} zutabeak :: [[Integer]] \text{ -> } [[Integer]] \\ zutabeak\ \ s\ \ldots \end{array}$$

Adibideak:

$$zutabeak\ \ [[6,4,7],[7,8,6],[9,5,1],[0,\text{-}5,2]]\ =\ [[6,7,9,0],[4,8,5,\text{-}5],[7,6,1,2]]$$

Datutzat emandako zerrenda matrize bezala ulertuko bagenu, hau da, s-ko zerrenda bakoitza errenkada bat bezala ulertuko bagenu, matrize horretako zutabeak kalkulatu dira.

$$\begin{array}{lll} zutabeak & [[6,4,7], & \\ & [7,8,6], & \\ & [9,5,1], & \\ & [0,\text{-}5,2]] & = [[6,7,9,0],[4,8,5,\text{-}5],[7,6,1,2]] \end{array}$$

Errore-kasurako, aurredefinitutako not funtzioa, 3.1 atalean definitutako $luzerak$ funtzioa eta 3.2 atalean definitutako $guztiak\_berdinak$ funtzioa erabiltzea da aukera bat. s hutsa deneko kasurako eta s zerrenda hutsez osatuta dagoen kasurako, aurredefinitutako $null$ eta $head$ erabiltzea da aukera bat. Kasu orokorrerako, aurredefinitutako $genericLength$ eta $head$ funtzioak eta 3.3 atalean definitutako $posiziokoak\_hautatu$ funtzioak erabiltzea litzateke aukera bat.

3.5. (0,150 puntu) Zenbaki osozko r eta s bi zerrenda emanda, zerrenda biek luzera bera badute, posizioz posizioko biderkaduren batura kalkulatuko duen $posi\_bider\_batu$ izeneko funtzioa definitu Haskellez. r eta s zerrendek luzera bera ez badute, errore-mezua aurkeztu beharko da. Zerrenda biak hutsak baldin badira ere, errore-mezua aurkeztu beharko da.

$$\begin{array}{l} posi\_bider\_batu :: [Integer] \text{ -> } [Integer] \text{ -> } Integer \\ posi\_bider\_batu\ \ r\ \ s\ \ldots \end{array}$$

Adibidea:

$$posi\_bider\_batu\ \ [2,5,\text{-}2,4]\ \ [6,7,9,0]\ =\ (2*6)+(5*7)+((\text{-}2)*9)+(4*0)\ =\ 29$$

Errore-kasurako $genericLength$ erabiltzea da aukera bat. Bestalde, kasu orokorrerako, aurredefinitutako zip eta sum funtzioak erabiltzea da aukera bat.

3.6. (0,150 puntu) Zenbaki osozko r zerrenda eta zenbaki osozko zerrendez osatutako s zerrenda emanda, r zerrenda eta s-ko zerrenda bakoitzaren arteko posizioz posizioko biderkaduren baturez osatutako zerrenda itzuliko duen $errenkada\_kalkulatu$ izeneko funtzioa definitu Haskellez. r zerrenda eta s-ko zerrenda guztiak luzera berekoak ez badira, errore-mezua aurkeztu beharko da. Bestalde, r zerrenda hutsa baldin bada eta s-ko zerrenda denak ere hutsak baldin badira, errore-mezua aurkeztu beharko da. Azkenik, r edozein eratakoa izanda ere, s hutsa baldin bada, errore-mezua aurkeztu beharko da. Laburtuz, ez da onartuko r hutsa izatea edo s hutsa izatea.

$$\begin{array}{l} errenkada\_kalkulatu :: [Integer] \text{ -> } [[Integer]] \text{ -> } [Integer] \\ errenkada\_kalkulatu\ \ r\ \ s\ \ldots \end{array}$$

Adibidea:

$$\begin{array}{l} errenkada\_kalkulatu\ \ [2,5,\text{-}2,4]\ \ [[\underline{6,7,9,0}],[\underline{\underline{4,8,5,\text{-}5}}],\underline{[7,6,1,2]}]\ = \\ [\underline{((2*6)+(5*7)+((\text{-}2)*9)+(4*0))},\underline{\underline{((2*4)+(5*8)+((\text{-}2)*5)+(4*\text{-}5))}}, \\ \underline{((2*7)+(5*6)+((\text{-}2)*1)+(4*2))}]\ =\ [\underline{29},\underline{\underline{18}},\underline{50}] \end{array}$$

Errore-kasurako, aurredefinitutako not eta $null$ funtzioak eta 3.1 atalean definitutako $luzerak$ eta 3.2 atalean definitutako $guztiak\_berdinak$ funtzioak erabiltzea da aukera bat. Kasu orokorrerako, 3.5 atalean definitutako $posi\_bider\_batu$ funtzioa erabil daiteke.

3.7. (0,150 puntu) Bi matrizeren arteko biderketa simulatuko duen $matrize\_biderketa$ izeneko funtzioa idatzi behar da Haskellez. Matrizeak zenbaki osozko zerrendez osatutako r eta s zerrenden bidez adieraziko dira. Biderketa egin ahal izateko, lehenengo matrizeko zutabe-kopuruak bigarren matrizeko errenkada-kopuruaren derdina izan beharko du. Beste era batera esanda, r-ko zerrenden luzerek s-ko zerrenda-kopuruaren berdinak izan beharko dute. r osatzen duten zerrenda guztiek luzera bera ez badute edo s osatzen duten zerrenda guztiek luzera bera ez badute, errore-mezua aurkeztu beharko da.

Bestalde, r edo s hutsa baldin bada ere, errore mezua aurkeztu beharko da. Gainera, r edo s zerrenda hutsez osatuta baldin badago ere, errore-mezua aurkeztu beharko da. Azkenik, r-ko zerrenden luzera s-ko zerrenda-kopuruaren berdina ez bada ere, errore-mezua aurkeztu beharko da.

$$matrize\_biderketa :: [[Integer]] -> [[Integer]] -> [[Integer]]$$
$$matrize\_biderketa\ r\ s\ \ldots$$

Adibidea:

$$matrize\_biderketa\ \ [[2,3,5],[1,8,6]]\ \ [[7,9],[1,7],[5,3]]\ =$$
$$[[\underline{2*7+3*1+5*5}, \underline{\underline{2*9+3*7+5*3}}], [\underline{1*7+8*1+6*5}, \underline{\underline{1*9+8*7+6*3}}]]\ =$$
$$[[\underline{42}, \underline{\underline{54}}], [\underline{45}, \underline{\underline{83}}]]$$

Matrize eran ikusiz, honako hau izango genuke:

$$\begin{pmatrix} 2 & 3 & 5 \\ 1 & 8 & 6 \end{pmatrix} \times \begin{pmatrix} 7 & 9 \\ 1 & 7 \\ 5 & 3 \end{pmatrix} = \begin{pmatrix} 2*7+3*1+5*5 & 2*9+3*7+5*3 \\ 1*7+8*1+6*5 & 1*9+8*7+6*3 \end{pmatrix} = \begin{pmatrix} 42 & 54 \\ 45 & 83 \end{pmatrix}$$

Errore-kasuetarako, aurredefinitutako $null$, $head$, not eta $genericLength$ funtzioak eta 3.1 atalean definitutako $luzerak$ funtzioa eta 3.2 atalean definitutako $guztiak\_berdinak$ funtzioa erabiltzea da aukera bat. Kasu orokorrerako, 3.4 atalean definitutako $zutabeak$ funtzioa eta 3.6 atalean definitutako $errenkada\_kalkulatu$ funtzioa erabil daitezke.

3.8. (0,150 puntu) $matrize\_biderketa$ funtzioaren konputazio-kostua aztertu.