Lengoaiak, Konputazioa eta Sistema Adimendunak

Bilboko IITUE

2014-01-13

2. gaia: Lengoaiak – 0,9 puntu

1 A^* zenbagarria da eta 2^{A^*} zenbaezina da (0,325 puntu)

1.1. (0,025 puntu) Har dezagun $A = \{a, b\}$ alfabetoa. A^*-ko hitzak zenbatuz joateko era egokia zein den zehaztu. Horretarako, lau elementuz osatutako lehenengo hitzera arteko hitz denak orden egokian eman (lau elementuz osatutako lehenengo hitza ere eman).

1.2. (0,300 puntu) Har dezagun edozein A alfabeto. Kontraesanaren teknika erabiliz, 2^{A^*} zenbaezina dela frogatu.

2 Lengoaien definizioa (0,575 puntu)

Har dezagun $A = \{a, b, c\}$ alfabetoa:

2.1. (0,050 puntu) Posizio bikoiti denetan c sinboloa duten hitzez osatutako L_1 lengoaiaren definizio formala eman. Dena den, posizio bakoitietan ere c ager daiteke. Ezkerreko ertzean dagoen elementua 1 posizioan dagoela kontsideratu behar da. Adibidez, $bcacbcb$, $bcacbcbc$, ε, ccc, $cccc$, $ccacbc$, a, b eta $ccbcbc$ hitzak L_1 lengoaiakoak dira baina cba, aa, $cabbbc$ eta aaa ez dira L_1 lengoaiakoak.

2.2. (0,100 puntu) Hutsak ez diren eta abc hitza nahi adina aldiz elkartuz eratzen diren hitzez osatutako L_2 lengoaiaren definizio formala eman. Adibidez, abc, $abcabc$ eta $abcabcabc$ lengoaiakoak dira baina ε, $abba$, $ccccc$ eta $abcc$ ez dira L_2 lengoaiakoak.

2.3. (0,100 puntu) Hutsak ez diren eta a bakoitzaren jarraian gutxienez a-ren desberdinak diren bi elementu dituzten hitzez osatutako L_3 lengoaiaren definizio formala eman. Adibidez, $bcbbccc$, abc, ccc, $abbbacbabb$, $baccbbbacb$ eta $ccbbc$ lengoaiakoak dira baina ε, aa, aba, $acbcca$ eta ab ez dira L_3 lengoaiakoak.

2.4. (0,075 puntu) abc azpikatea gutxienez behin duten hitzez osatutako L_4 lengoaiaren definizio formala eman. Adibidez, $ccabcaa$, $aaabcacabcccb$, $accabcaab$, $ccbaaabcaaab$ eta $aabcaaa$ L_4 lengoaiakoak dira baina ε, $bacbcc$ eta $aabbaa$ ez dira L_4 lengoaiakoak.

2.5. (0,050 puntu) abc azpikatea gutxienez behin duten eta a bakoitzaren jarraian gutxienez a-ren desberdinak diren bi elementu dituzten hitzez osatutako L_5 lengoaiaren definizio formala eman. Adibidez, $babccc$, $abcaccacbabccbcb$ eta $ccabcccc$ hitzak L_5 lengoaiakoak dira baina ε, $abcbbba$, $baaabb$ eta $aaabccc$ ez dira L_5 lengoaiakoak.

2.6. (0,075 puntu) abc azpikatea ez duten hitzez osatutako L_6 lengoaiaren definizio formala eman. Adibidez, ε, $aababbccc$, ac eta $bbbb$ hitzak L_6 lengoaiakoak dira. Bestalde, $ccbbcabcaa$, $aabcbc$ eta $aabccc$ hitzak ez dira L_6 lengoaiakoak.

2.7. (0,050 puntu) Osagai denak a edo bestela a-rik ez duten hitzez osatutako L_7 lengoaiaren definizio formala eman. Adibidez, a, aaa, $bcbbbc$, ε, c, bbc, ccc eta $aaaa$ hitzak L_7 lengoaiakoak dira baina $babb$, $cbbcac$ eta $caacc$ ez dira L_7 lengoaiakoak.

2.8. (0,075 puntu) Hiru edo handiagoa den luzera bakoitia eta a sinboloa hasierako, erdiko eta azkeneko posizioetan bakarrik duten hitzez osatutako L_8 lengoaiaren definizio formala eman. Adibidez, aaa, $abaca$, $ababa$ eta $accbaccca$ hitzak L_8 lengoaiakoak dira baina a, $acbbcc$, bba, ε, b eta $aaabccc$ ez dira L_8 lengoaiakoak.

3. gaiko lehenengo zatia: AFD-ak eta minimizazioa – 1,6 puntu

1 Automata finitu deterministen (AFD-en) diseinua (0,900 puntu)

$A = \{a, b, c\}$ alfabetoaren gainean definitutako honako hiru lengoaientzat AFD bana diseinatu:

1.1 c-rik ez eta, edozein ordenatan, gutxienez a bat eta gutxienez b bat dituzten hitzen lengoaia (0,300 puntu)

c sinboloaren agerpenik ez eta gutxienez a sinboloaren agerpen bat eta gutxienez b sinboloaren agerpen bat dituzten hitzez osatutako L_1 lengoaia. a eta b sinboloen agerpenei dagokionez, ordenak ez du garrantzirik. Adibidez, $bbbab$, $ababbb$, ba, ab eta $bbbaaaa$ hitzak L_1 lengoaiakoak dira baina aac, $aabcbc$, $aacc$, aaa, $bbbb$ eta ε hitzak ez dira L_1 lengoaiakoak. L_1 lengoaiaren definizio formala honako hau da:

$$L_1 = \{w \mid w \in A^* \wedge |w|_a \geq 1 \wedge |w|_b \geq 1 \wedge |w|_c = 0\}$$

1.2 c-rik agertzen bada, a-rik eta b-rik ez duten hitzen lengoaia (0,300 puntu)

c sinboloaren agerpenik baldin badute, a eta b sinboloen agerpenik ez duten hitzen L_2 lengoaia. Beraz hitz batean a eta b nahasian ager daitezke, baina c agertzen bada, orduan hitza c-ren errepikapenez osatutakoa izango da, hau da, ez du a-rik eta b-rik izango. Adibidez, $abaabba$, $aaba$, aaa, ε, ccc, bb eta $bbaaab$ hitzak L_2 lengoaiakoak dira baina $aacb$, $bccbbb$, $cccaa$ eta $ccaaabab$ ez. Jarraian L_2 lengoaiaren bi definizio formal erakusten dira:

$$L_2 = \{w \mid w \in A^* \wedge (|w|_c = |w| \vee |w|_c = 0)\}$$
$$L_2 = \{w \mid w \in A^* \wedge (|w|_c \geq 1 \rightarrow (|w|_a = 0 \wedge |w|_b = 0))\}$$

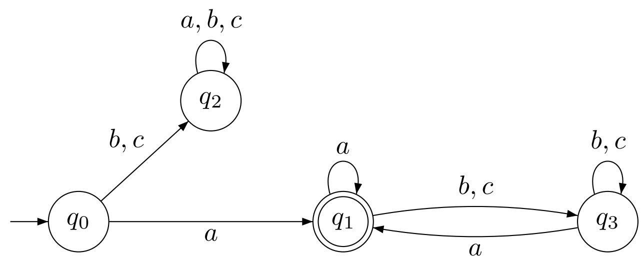
1.3 Bi zati osatuz agertzen diren bi sinboloren errepikapenez osatutako hitzen lengoaia (0,300 puntu)

Bi zati osatuz agertzen diren alfabetoko bi sinbolo desberdinen errepikapenez eratutako hitzen L_3 lengoaia. Zati bakoitzak gutxienez elementu bat izan beharko du. Adibidez, $aaabbbb$, $bbaaaa$, $ccccaa$, $bbbccc$, $aaac$ eta $cbbb$ hitzak L_3 lengoaiakoak dira baina $aaba$, a, aa, $abbabcaa$ eta ε ez. L_3 lengoaiaren definizio formala honako hau da:

$$\begin{aligned} L_3 = \{w \mid w \in A^* \wedge \exists \alpha, \beta, u, v \quad & (\alpha \in A \wedge \beta \in A \wedge u \in A^* \wedge v \in A^* \wedge \alpha \neq \beta \\ & \wedge |u| \geq 1 \wedge |v| \geq 1 \wedge |u| = |u|_\alpha \wedge |v| = |v|_\beta \wedge w = uv)\} \end{aligned}$$

2 Konputazio deterministen garapena (0,150 puntu)

Jarraian erakusten den AFD-a kontuan hartuz, hor zehazten diren konputazioak garatu urratsez urrats, bukaeran AFD-ak “Bai” ala “Ez” erantzungo duen esanez:

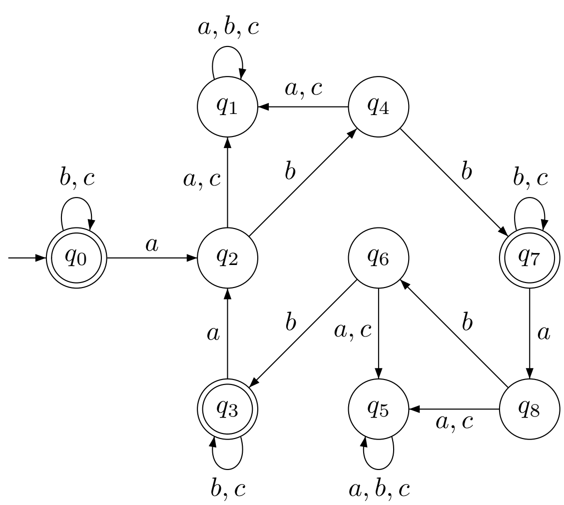


1. $\delta^*(q_0, aba)$
2. $\delta^*(q_0, aaa)$
3. $\delta^*(q_0, \varepsilon)$
4. $\delta^*(q_0, abb)$
5. $\delta^*(q_0, a)$

Kasu bakoitzak 0,030 balio du.

3 AFD-en minimizazioa (0,550 puntu)

$A = \{a, b, c\}$ alfabetoaren gainean definitutako honako AFD hau minimizatu:



3. gaiko bigarren zatia: AFED-ak – Puntu 1

1 Automata finitu ez determinististen (AFED-en) diseinua (0,860 puntu)

$A = \{a, b, c\}$ alfabetoaren gainean definitutako honako lau lengoaientzat AFED bana diseinatu:

1.1 Hurrenez hurren a-z, b-z, c-z, b-z eta a-z eratutako bost bloke ez hutsez osatutako hitzen lengoaia (0,215 puntu)

Jarraian zehazten den eran bost blokez eratutako hitzez osatutako L_1 lengoaia: lehenengo eta bosgarren blokeak a sinboloaren errepikapenez eratuta egon behar dute, bigarrenak eta laugarrenak b sinboloaren errepikapenez osatutakoak izan behar dute eta hirugarrenak c sinboloaren errepikapenez eratutakoa izan beharko du. Bloke bakoitzak gutxienez elementu bat izan beharko du eta bloke desberdinek luzera desberdina izan dezakete. Adibidez, $aabcccbbaaaa$, $abcba$, $aaaabccbaaa$ eta $abccccbaa$ hitzak L_1 lengoaiakoak dira baina aac, $aabcbc$, a, $aaaa$, aaa, ba, $aaabbbb$, $aabbac$, $bbacccabb$ eta ε hitzak ez dira L_1 lengoaiakoak. L_1 lengoaiaren definizio formala honako hau da:

$$L_1 = \{w \mid w \in A^* \wedge \exists u, v, x, y, z(\begin{array}{l} u \in A^* \wedge v \in A^* \wedge x \in A^* \wedge y \in A^* \wedge z \in A^* \wedge \\ |u| \geq 1 \wedge |v| \geq 1 \wedge |x| \geq 1 \wedge |y| \geq 1 \wedge |z| \geq 1 \wedge \\ |u| = |u|_a \wedge |v| = |v|_b \wedge |x| = |x|_c \wedge |y| = |y|_b \wedge |z| = |z|_a \wedge \\ w = uvxyz)\} \end{array}$$

1.2 Hutsak izan daitezkeen eta hurrenez hurren a-z, b-z, c-z, b-z eta a-z eratuta dauden bost blokez osatutako hitzen lengoaia (0,215 puntu)

Jarraian zehazten den eran bost blokez eratutako hitzez osatutako L_2 lengoaia: lehenengo eta bosgarren blokeak a sinboloaren errepikapenez eratuta egon behar dute, bigarrenak eta laugarrenak b sinboloaren errepikapenez osatutakoak izan behar dute eta hirugarrenak c sinboloaren errepikapenez eratutakoa izan beharko du. Blokeak hutsak izan daitezke eta bloke desberdinek luzera desberdina izan dezakete. Adibidez, $aabcccbbaaaa$, $abcba$, $aaaabccbaaa$, $abccccbaa$, $aaaa$, ccc, $aacc$, bbb, $bbbccb$, ε eta $bbaaa$ hitzak L_2 lengoaiakoak dira baina $aacbc$, $aabcbc$, $acbaabb$, bab, $aabbac$, $ccbbaac$ eta $cccaabbb$ hitzak ez dira L_2 lengoaiakoak. L_2 lengoaiaren definizio formala honako hau da:

$$L_2 = \{w \mid w \in A^* \wedge \exists u, v, x, y, z(\begin{array}{l} u \in A^* \wedge v \in A^* \wedge x \in A^* \wedge y \in A^* \wedge z \in A^* \wedge \\ |u| = |u|_a \wedge |v| = |v|_b \wedge |x| = |x|_c \wedge |y| = |y|_b \wedge |z| = |z|_a \wedge \\ w = uvxyz)\} \end{array}$$

1.3 bb azpikatea ez duten hitzen lengoaia (0,215 puntu)

bb azpikatea ez duten hitzez osatutako L_3 lengoaia. Adibidez, $aaaca$, $accb$, $aabaabab$, $baacab$, $ccab$, bcc, ε, a, b eta $acccaaccb$ hitzak L_3 lengoaiakoak dira baina $aabb$, $bbbb$, $abbaabba$, $caaabbac$, $cccbcbb$ eta $babbbcb$ ez. L_3 lengoaiaren definizio formala honako hau da:

$$L_3 = \{w \mid w \in A^* \wedge \neg\exists u, v(u \in A^* \wedge v \in A^* \wedge w = ubbv)\}$$

1.4 a kopuru bikoitia edo a-z hasi eta a-z bukatzen diren hitzen lengoaia (0,215 puntu)

Jarraian zehazten diren bi baldintzetatik gutxienez bat betetzen duten hitzez osatutako L_4 lengoaia:

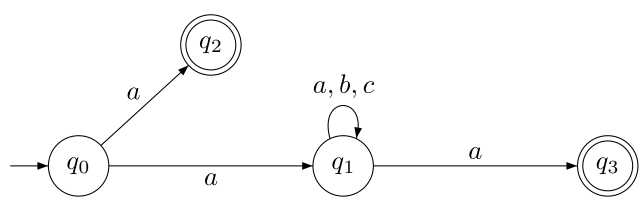
- a kopuru bikoitia izatea
- a-z hasi eta a-z bukatzea

Adibidez, $aaabac$, $ccaaacba$, $bbaaaab$, ε, a, b, aa, $abacaaaca$ eta $acbaaacca$ hitzak L_4 lengoaiakoak dira baina $aabab$, $bbbab$, $baaa$, $aacbbaacab$ eta $babcb$ ez. L_4 lengoaiaren definizio formala honako hau da:

$$L_4 = \{w \mid w \in A^* \wedge (|w|_a \text{ mod } 2 = 0 \vee \exists u(u \in A^* \wedge w = aua))\}$$

2 Konputazio ez deterministen garapena (0,140 puntu)

Jarraian erakusten den AFED-a kontuan hartuz, hor zehazten diren konputazioak garatu urratsez urrats, bukaeran AFED-ak "Bai" ala "Ez" erantzungo duen esanez:



1. $\nu^*(\{q_0\}, aba)$
2. $\nu^*(\{q_0\}, aaa)$
3. $\nu^*(\{q_0\}, abb)$
4. $\nu^*(\{q_0\}, \varepsilon)$
5. $\nu^*(\{q_0\}, a)$

1, 2 eta 3 kasuek 0,030 balio dute bakoitzak eta 4 eta 5 kasuek 0,025 bakoitzak.

4. gaia: Lengoaia erabakigarriak, lengoaia bereizgarriak eta lengoaia bereiztezinak – 1,5 puntu

1 L_{bai} lengoaia bereizgarria da (0,200 puntu)

$$L_{bai} = \{\langle T, w\rangle \mid T \text{ Turing-en makinak } w \text{ hitzarentzat "Bai" erantzuten du}\}$$

L_{bai} lengoaia bereizteko jarraitu beharreko algoritmoa edo eskema emanez, L_{bai} lengoaia bereizgarria dela frogatu.

2 L_{bai} lengoaia erabakiezina da (0,300 puntu)

Kontraesanaren teknika erabiliz, L_{bai} lengoaia erabakigarria ez dela frogatu.

3 L_{halt} lengoaia bereizgarria da (0,200 puntu)

$$L_{halt} = \{\langle T, w\rangle \mid T \text{ Turing-en makinak } w \text{ hitza ematen zaionean, “Bai” edo “Ez” erantzuten du}\}$$

L_{halt} lengoaia bereizteko jarraitu beharreko algoritmoa edo eskema emanez, L_{halt} lengoaia bereizgarria dela frogatu.

4 L_{halt} lengoaia erabakiezina da (0,300 puntu)

Kontraesanaren teknika erabiliz, L_{halt} lengoaia erabakigarria ez dela frogatu.

5 Bereiztezinak diren lengoaiak badaude (0,200 puntu)

$A = \{0, 1\}$ alfabetoa, A^*-ren zenbagarritasuna, 2^{A^*}-ren zenbaezintasuna eta Turing-en makinak A^*-ko hitzen bidez adierazi daitezkeela kontuan hartuz, bereiztezinak diren lengoaiak badaudela frogatu.

6 $\overline{L_{bai}}$ bereiztezina da (0,300 puntu)

Har dezagun $\overline{L_{bai}}$ lengoaia:

$$\overline{L_{bai}} = \{\langle T, w\rangle \mid T \text{ Turing-en makinak } w \text{ hitzarentzat ez du “Bai” erantzuten}\}$$

Kontraesanaren teknika erabiliz, L_{bai} lengoaiaren osagarria, hau da, $\overline{L_{bai}}$ lengoaia, bereiztezina dela frogatu.

6. gaia: Sistema Adimendunak – 0,9 puntu

1 DNF monotonoen algoritmoa (0,300 puntu)

Demagun erabiltzaileak DNF monotonoa den honako g formula hau duela buruan:

$$g = (x_1 \wedge x_2) \vee (x_1 \wedge x_3) \vee (x_1 \wedge x_5)$$

Aldagai kopurua 5 dela jakinda, hau da, $n = 5$ dela jakinda, algoritmoak g-ren baliokidea den h formula bat eraiki arte erabiltzailearen eta algoritmoaren artean gertatuko den elkarrekintza urratsez urrats zehaztu. Beraz, adibide osoa garatu beharko da eta prozesu horretan algoritmoarentzat pista edo laguntza izango diren balorazio egokiak asmatu beharko dira.

2 k-CNF-en algoritmoa (0,300 puntu)

Demagun erabiltzaileak 2-CNF-a den honako g formula hau duela buruan:

$$g = (x_1 \vee x_3) \wedge (\neg x_1 \vee \neg x_3)$$

Aldagai kopurua 3 dela jakinda, hau da, $k = 2$ eta $n = 3$ direla jakinda, algoritmoak g-ren baliokidea den h formula bat eraiki arte erabiltzailearen eta algoritmoaren artean gertatuko den elkarrekintza urratsez urrats zehaztu. Beraz, adibide osoa garatu beharko da eta prozesu horretan algoritmoarentzat pista edo laguntza izango diren balorazio egokiak asmatu beharko dira.

3 k-DNF-en algoritmoa (0,300 puntu)

Demagun erabiltzaileak 1-DNF-a den honako g formula hau duela buruan:

$$g = (x_1) \vee (\neg x_4) \vee (x_5)$$

Aldagai kopurua 5 dela jakinda, hau da, $k = 1$ eta $n = 5$ direla jakinda, algoritmoak g-ren baliokidea den h formula bat eraiki arte erabiltzailearen eta algoritmoaren artean gertatuko den elkarrekintza urratsez urrats zehaztu. Beraz, adibide osoa garatu beharko da eta prozesu horretan algoritmoarentzat pista edo laguntza izango diren balorazio egokiak asmatu beharko dira.

7. gaia: Haskell – 1,6 puntu

1 Murgilketa (0,300 puntu)

Osoa den x zenbakia emanda, bere zatitzaile bikoitien kopurua zatitzaile bakoitien kopurua baino handiagoa baldin bada $True$ eta bestela $False$ itzuliko duen $bik\_gehiago$ funtzioa definitu behar da. Zatitzaile bikoitien eta bakoitien kopurua berdina bada $False$ itzuli beharko da. x parametroaren balio 1 baino txikiagoa baldin bada, errore-mezua aurkeztu beharko da.

$$bik\_gehiago :: Int \text{ -> } Bool$$
$$bik\_gehiago\ \ x\ \ \ldots$$

Adibideak:

$bik\_gehiago\ 3$	$=\ False$	3 zenbakiaren zatitzaile bikoitien kopurua (zero) ez delako zatitzaile bakoitien kopurua (bi, 1 eta 3) baino handiagoa.
$bik\_gehiago\ 4$	$=\ True$	4 zenbakiak bi zatitzaile bikoiti (2, 4) eta zatitzaile bakoiti bat (1) dituelako.
$bik\_gehiago\ 6$	$=\ False$	6 zenbakiak bi zatitzaile bikoiti (2, 6) eta bi zatitzaile bakoiti (1, 3) dituelako.

Murgikeltaren teknika jarraituz, osoak diren x, bm, bik eta bak lau zenbaki emanda, bm-tik hasita x zenbakiaren zatitzaile bikoitien kopurua gehi bik zatitzaile bakoitien kopurua gehi bak baino handiagoa baldin bada $True$ eta bestela $False$ itzuliko duen $bik\_gehiago\_lag$ funtzioa definitu behar da. bm-tik hasita, x zenbakiaren zatitzaile bikoitien kopurua gehi bik eta zatitzaile bakoitien kopurua gehi bak berdinak badira, $False$ itzuli beharko da. x balioa 1 baino txikiagoa baldin bada edo bm balioa 1 baino txikiagoa baldin bada, errore-mezua aurkeztu beharko da. Bestalde, bm balioa x baino handiagoa bada ere, errore-mezua aurkeztu beharko da.

$$bik\_gehiago\_lag :: Int \text{ -> } Int \text{ -> } Int \text{ -> } Int \text{ -> } Bool$$
$$bik\_gehiago\_lag\ \ x\ \ bm\ \ bik\ \ bak\ \ \ldots$$

Adibideak:

$bik\_gehiago\_lag\ \ 3\ \ 1\ \ 0\ \ 0\ \ =\ \ False$
1etik hasita, 3ren zatitzaile bikoitien kopurua gehi 0 ez delako 3ren zatitzaile bakoitien kopurua gehi 0 baino handiagoa.

$bik\_gehiago\_lag\ \ 3\ \ 1\ \ 8\ \ 0\ \ =\ \ True$
1etik hasita, 3ren zatitzaile bikoitien kopurua gehi 8 3ren zatitzaile bakoitien kopurua gehi 0 baino handiagoa delako.

$bik\_gehiago\_lag\ \ 50\ \ 1\ \ 0\ \ 0\ \ =\ \ False$
1etik hasita, 50en zatitzaile bikoitien (2, 10, 50) kopurua gehi 0 ez delako 50en zatitzaile bakoitien (1, 5, 25) kopurua gehi 0 baino handiagoa.

$bik\_gehiago\_lag\ \ 50\ \ 6\ \ 0\ \ 0\ \ =\ \ True$
6tik hasita, 50en zatitzaile bikoitien (10, 50) kopurua gehi 0 50en zatitzaile bakoitien (25) kopurua gehi 0 baino handiagoa delako.

$bik\_gehiago\_lag$ funtzioa $bik\_gehiago$ funtzioa baino orokorragoa da. $bik\_gehiago$ funtzioa $bik\_gehiago\_lag$ funtzioaren bidez definitzerakoan bm parametroa zatitzaileak bilatzerakoan zeharkatu beharreko tartearen beheko muga bezala erabili behar da. Bestalde, bik eta bak parametroak zatitzaile bikoitiak eta bakoitiak zenbatuz joateko erabili behar dira.

2 Bukaerako errekurtsibitatea (0,300 puntu)

Har dezagun honako funtzio hau:

$$
\begin{array}{l}
txertatu :: Integer \text{ -> } [Integer] \text{ -> } [Integer] \\
txertatu\ \ x\ \ [\,]\ \ =\ \ x : [\,] \\
txertatu\ \ x\ \ (y : s) \\
\ \ \mid\ x \leq y\ \ \ = x : (y : s) \\
\ \ \mid\ otherwise\ \ \ = y : (txertatu\ x\ \ s)
\end{array}
$$

Osoa den x zenbaki bat eta zenbaki osozko zerrenda bat emanda, x balioa x baino handiagoa edo berdina den zerrendako lehenengo elementuaren aurrean (ezkerreko aldean) kokatuz lortzen den zerrenda itzuliko du $txertatu$ funtzioak.

Adibideak:

$$
\begin{array}{rcl}
txertatu\ \ 5\ \ [4,8,3,6] & = & [4,5,8,3,6] \\
txertatu\ \ 2\ \ [4,8,3,6] & = & [2,4,8,3,6] \\
txertatu\ \ 8\ \ [4,8,3,6] & = & [4,8,8,3,6] \\
txertatu\ \ 10\ \ [4,8,3,6] & = & [4,8,3,6,10]
\end{array}
$$

$txertatu$ funtzioak ez du bukaerako errekurtsibitaterik. Bukaerako errekurtsibitatea edukitzeko, honako bi funtzio hauek definitu behar dira:

- $txertatu$ funtzioak jasotzen dituen x zenbakiaz eta zerrendaz gain, emaitza bezala eraikiz joango den zerrenda gordez joateko erabiliko den bigarren zerrenda duen $txertatu\_lag$ funtzioa. Beraz, $txertatu\_lag$ funtzioak jarraian zehazten diren bi zerrendak elkartuz lortzen den zerrenda itzuli beharko du:
  - Alde batetik, datu bezala emandako bigarren zerrenda.
  - Beste aldetik, datu bezala emandako lehenengo zerrendan x balioa x baino handiagoa edo berdina den lehenengo elementuaren aurrean (ezkerreko aldean) kokatuz lortzen den zerrenda berria.

$$
\begin{array}{rcl}
txertatu\_lag\ \ 5\ \ [4,8,3,6]\ \ [\,] & = & [4,5,8,3,6] \\
txertatu\_lag\ \ 5\ \ [4,8,3,6]\ \ [12,10] & = & [12,10,4,5,8,3,6] \\
txertatu\_lag\ \ 10\ \ [4,8,3,6]\ \ [12,10] & = & [12,10,4,8,3,6,10]
\end{array}
$$

- $txertatu\_lag$ funtzioari egokiak diren parametroekin deituz $txertatu$ funtzioak egiten duen gauza bera egingo duen $txertatu\_be$ funtzioa.

$$
\begin{array}{rcl}
txertatu\_be\ \ 5\ \ [4,8,3,6] & = & [4,5,8,3,6] \\
txertatu\_be\ \ 2\ \ [4,8,3,6] & = & [2,4,8,3,6] \\
txertatu\_be\ \ 8\ \ [4,8,3,6] & = & [4,8,8,3,6] \\
txertatu\_be\ \ 10\ \ [4,8,3,6] & = & [4,8,3,6,10]
\end{array}
$$

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

3 Zerrenda-eraketa (1,000 puntu)

3.1. (0,100 puntu) Zenbaki osozko zerrenda bat emanda, zerrendan elementu negatiborik baldin badago $True$ eta bestela $False$ itzultzen duen $negatiborik$ izeneko funtzioa definitu Haskell lengoaia erabiliz.

$$negatiborik :: [Integer] \text{ -> } Bool$$
$$negatiborik \ \ \ldots$$

Adibideak:

$$negatiborik \ \ [7,2,-3,8,-9] \ = True$$
$$negatiborik \ \ [7,2,5,0] \ = False$$

Aukera bat aurredefinitutako $length$ funtzioa erabiltzea da.

3.2. (0,150 puntu) Zenbaki osozko zerrendez eratutako zerrenda bat emanda, zerrenda bakoitzean elementu negatiborik agertzen al den adierazten duen balio Boolearrezko zerrenda itzultzen duen $neg\_agerpenik$ izeneko funtzioa definitu Haskell lengoaia erabiliz.

$$neg\_agerpenik :: [[Integer]] \text{ -> } [Bool]$$
$$neg\_agerpenik \ \ \ldots$$

Adibideak:

$$neg\_agerpenik \ = \ [[7,2,8],[0,0],[\,],[4,-6,8],[-2,-2]] \ = \ [False, False, False, True, True]$$
$$neg\_agerpenik \ = \ [\,] \ = \ [\,]$$
$$neg\_agerpenik \ = \ [[2,8],[10],[-1],[4,6]] \ = \ [False, False, True, False]$$

Aukera bat aurreko ariketako $negatiborik$ funtzioa erabiltzea da.

3.3. (0,150 puntu) Zenbaki osozko zerrendez eratutako zerrenda bat emanda, elementu negatiborik ez duten zerrendak bakarrik mantenduz geratzen den zerrenda itzultzen duen neg_gabe izeneko funtzioa definitu Haskell lengoaia erabiliz.

$$neg\_gabe :: [[Integer]] \text{ -> } [[Integer]]$$
$$neg\_gabe \ \ \ldots$$

Adibideak:

$$neg\_gabe \ = \ [[7,2,8],[0,0],[\,],[4,-6,8],[-2,-2]] \ = \ [[7,2,8],[0,0],[\,]]$$
$$neg\_gabe \ = \ [\,] \ = \ [\,]$$
$$neg\_gabe \ = \ [[2,8],[10],[-1],[4,6]] \ = \ [[2,8],[10],[4,6]]$$
$$neg\_gabe \ = \ [[3,-9],[-20],[-1],[-2,-7,6]] \ = \ [\,]$$

Aukera bat 3.1 ariketako $negatiborik$ funtzioa erabiltzea da.

3.4. (0,100 puntu) Osoa den zenbaki bat emanda, zenbaki horren zatitzaileen zerrenda itzuliko duen $zatitzaileak$ funtzioa definitu Haskell lengoaia erabiliz.

$$zatitzaileak :: Integer \text{ -> } [Integer]$$
$$zatitzaileak \ \ \ldots$$

Adibideak:

$$\begin{aligned} zatitzaileak\ \ 8 = &\ \ [1,2,4,8] \\ zatitzaileak\ \ 7 = &\ \ [1,7] \\ zatitzaileak\ \ 0 = &\ \ [\,] \end{aligned}$$

3.5. (0,100 puntu) Osoak diren zenbaki positibo denen zatitzaileen zerrendez osatutako zerrenda infinitua aurkeztuz joango de zat_denak funtzioa definitu Haskell lengoaia erabiliz.

$$\begin{aligned} & zat\_denak :: [[Integer]] \\ & zat\_denak\ \ \ldots \end{aligned}$$

Adibideak:

$$zat\_denak \ \ = \ \ [[1],[1,2],[1,3],[1,2,4],[1,5],[1,2,3,6],\ldots]$$

Aukera bat 3.4 ariketako $zatitzaileak$ funtzioa erabiltzea da.

3.6. (0,100 puntu) Lehenengo osagai bezala osoa eta positiboa den zenbaki bat (zenbakien ohiko ordena jarraituz) eta bigarren osagai bezala lehenengo osagaiaren zatitzaile-zerrendaz osatutako bikoteez eratutako zerrenda infinitua kalkulatuz joango den zat_bikote funtzioa definitu Haskell lengoaia erabiliz.

$$\begin{aligned} & zat\_bikote :: [(Integer, Integer)] \\ & zat\_bikote\ \ \ldots \end{aligned}$$

Adibidea:

$$zat\_bikote \ \ = \ \ [(1,[1]),(2,[1,2]),(3,[1,3]),(4,[1,2,4]),(5,[1,5]),(6,[1,2,3,6]),\ldots]$$

Aukera bat aurreko ariketako zat_denak funtzioa eta aurredefinitutako zip funtzioa erabiltzea da.

3.7. (0,200 puntu) Osoa den n zenbaki bat emanda, n baino zatitzaile gehiago dituzten zenbakiei dagozkien zat_bikote zerrendako bikoteen zerrenda infinitua aurkeztuz joango den $zat\_gehiago$ funtzioa definitu Haskell lengoaia erabiliz.

$$\begin{aligned} & zat\_gehiago :: Integer \text{ -> } [(Integer, Integer)] \\ & zat\_gehiago\ \ \ldots \end{aligned}$$

Adibideak:

$$zat\_gehiago\ \ 3 \ \ = \ \ [(6,[1,2,3,6]),(8,[1,2,4,8]),(10,[1,2,5,10]),(12,[1,2,3,4,6,12]),\ldots]$$

Kasu horietan zatitzaile-kopurua 3 baino handiagoa da.

Aukera bat aurreko ariketako zat_bikote funtzioa eta aurredefinitutako $length$ funtzioa erabiltzea da.

3.8. (0,100 puntu) Osoak diren n eta kop bi zenbaki emanda, n baino zatitzaile gehiago dituzten lehenengo kop zenbakiei dagozkien zat_bikote zerrendako bikoteez osatutako zerrenda aurkeztuko duen $gehiago\_finitua$ funtzioa definitu Haskell lengoaia erabiliz.

$$\begin{aligned} & gehiago\_finitua :: Integer \text{ -> } Integer \text{ -> } [(Integer, Integer)] \\ & gehiago\_finitua\ \ \ldots \end{aligned}$$

Adibideak:

$$\begin{aligned} gehiago\_finitua\ \ 3\ \ 5 \ \ = & \ \ [(6,[1,2,3,6]),(8,[1,2,4,8]),(10,[1,2,5,10]),(12,[1,2,3,4,6,12]), \\ & \ \ (14,[1,2,7,14])] \end{aligned}$$

3 zatitzaile baino gehiago dituzten lehenengo 5 zenbakiei dagozkien bikoteak.

Aukera bat aurreko ariketako $zat\_gehiago$ funtzioa eta aurredefinitutako $genericTake$ funtzioa erabiltzea da.