



# Lengoiak, Konputazioa eta Sistema Adimendunak

Kudeaketaren eta Informazio Sistemen Informatikaren Ingeniaritzako Gradua

Bilboko Ingeniaritza Eskola (UPV/EHU)

2. maila

2019-2020 ikasturtea

## 1. gaia: Sarrera

JOSÉ GAINZARAIN IBARMIA

Lengoaia eta Sistema Informatikoak Saila

Azken eguneraketa: 2019 - 08 - 31



# Aurkibidea

<b>1</b>	<b>Sarrera</b>	<b>5</b>
<b>1.1</b>	<b>Aurkezpena</b>	<b>7</b>
<b>1.2</b>	<b>Motibazioa</b>	<b>9</b>
<b>1.3</b>	<b>Egoeratan eta egoeren arteko trantsizioetan oinarritutako makinak</b>	<b>11</b>
1.3.1	Adibideak . . . . .	11
1.3.1.1	LED bat kontrolatzen duen makina . . . . .	11
1.3.1.2	Hutsa ez den sekuentzia bateko osagai denak $a$ al diren erabakitzen duen makina . . . . .	12
1.3.1.3	Bukaeran zuriunea egongo dela jakinda, $\#$ sinboloarekin bukatzen diren eta 1-ez osatuta dauden bi kate elkartzen dituen makina. . . . .	12
1.3.2	Egoeratan oinarritutako makinetako memoria . . . . .	14
1.3.3	Egoeratan oinarritutako makinetan sarrera eta irteerako datuak adierazteko era: Lengoaiak . . . . .	15
1.3.4	Konputagarritasunaren azterketa egoeratan oinarritutako makinaren bidez . . .	16



**1. gaia**  
**Sarrera**



## 1.1.

# Aurkezpena

“Lengoaiak, Konputazioa eta Sistema Adimendunak” irakasgaian konputagarritasunaren azterketa formala landuko da eta konputazioaren oinarriak aurkeztuko dira. Horretarako, alde bate-tik, dagoeneko ondo finkatuta dauden emaitzak aurkeztuko dira eta, beste aldetik, irekita dauden arazoak eta konputagarritasunaren mugak zabaltzeko helburuarekin sortu diren ikuspegi berriak erakutsiko dira. Dagoeneko ondo finkatuta dauden emaitzak ezagutzeak konputagarritasunaren ikerketa nola egin den, zein tresna eta teknika erabili diren eta noraino iritsi den jakiteko balioko digu. Gainera hori dena ezagutzeak une honetan konputagarritasunaren arloan ikertzen ari di-ren bide berrien eta indarra hartzen ari diren ikuspegi berrien nondik norakoa hobeto ulertzeko balioko digu.





## 1.2.

# Motibazioa

Informatika (edo konputazioaren zientzia) informazioaren tratamendu automatikoaz arduratzen den jakintzaren (edo zientziaren) arloa da.

Informatikaren azken helburua gizakiek planteatzen eta egin ditzaketen kalkuluak era eraginkorrean egiteko gai diren sistema informatikoak garatzea da. Beraz, helburua sistema adimendunak garatzea izango litzateke.

Asko aurreratu izan arren, guztiz portaera adimenduna duten sistemak edukitzea ez da lortu oraindik. Aurkitu diren zailtasunen aurrean, helburu hori lortu ez izanaren arrazoiak ikuspuntu teorikotik aztertzea jo da, helburua zergatik ez den lortu jakin nahian: oraindik ez dugu teknologia egokia garatu ala gaindi ezin daitezkeen muga teorikoak daude? Beraz, konputagarritasunaren mugak non dauden jakin nahi da.

Konputazioaren mugen azterketa egitean era desberdinak edo eredu desberdinak erabili dira. Garrantzitsuenak honako hauek dira:

- Turing-en Makina (egoeratan eta egoeren arteko trantsizioetan oinarritua).
- $\lambda$ -kalkulua (funtzio matematikoetan oinarritua).
- Funtzio errekurtsiboak.
- Logika (batez ere Lehen Mailako Logikan eta ebazpenean oinarritua).
- Post-en Makina (fluxu-diagramatan oinarritua).

Konputazio-eredu bakoitzak programazio-eredu edo paradigma bat du berari lotuta:

- Turing-en Makina: Agindu bidezko programazioa (ADA, C, Java eta abar)
- $\lambda$ -kalkulua: Funtzio bidezko programazioa (Haskell, Scheme, ML eta abar)
- Funtzio errekurtsiboak: Programazio algebraikoa (Opal, Magma).
- Logika: Programazio logikoa (Prolog).
- Post-en Makina: Agindu bidezko programazioa (ADA, C, Java eta abar).

Irakasgai honetan konputazioaren alde teorikoa aztertzeko egoeratan eta egoeren arteko trantsizioetan oinarritutako eredua jarraituko da (Turing-en Makina) eta praktikak egiteko  $\lambda$ -kalkuluan oinarrituta dagoen Haskell lengoaia erabiliko dugu. Horrela, aipatu diren konputazio-ereduetatik bi landu ahal izango ditugu.

## 1.3.

# Egoeratan eta egoeren arteko trantsizioetan oinarritutako makinak

Egoeratan oinarritutako makina batek (egoera-makina batek) honako osagai hauek izaten ditu:

- Datuak jasotzeko balioko duen **sarrerako** gailua.
- Emaitzak aurkezteko balioko duen **irteerako** gailua.
- **Egoera-multzo** bat eta **trantsizio funtzioa**. Egoera batean gaudenean datu berri bat irakurtzen bada, zein egoeratarara igaro behar dugun adieraziko du trantsizio funtzioak. Datu bat irakurritakoan gerta daiteke egoera berean jarraitzea edo beste egoera batera igarotzea, trantsizio funtzioa jarraituz. Irteera bezala zerbait lortu edo aurkeztu behar bada, aurkeztu beharreko hori kalkulu-prozesuan zehar edo bukaeran erakutsiko da irteera bezala.
- Informazioa gordetzeko balioko duen **memoria**.

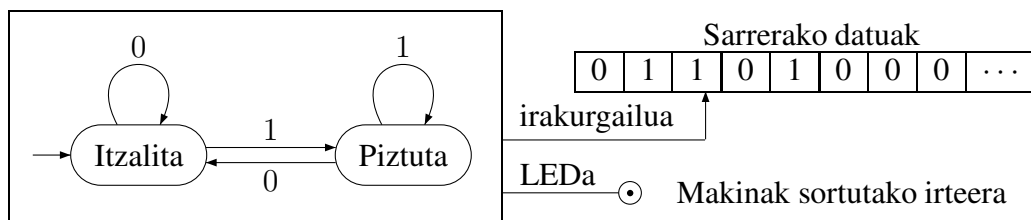
### 1.3.1 Adibideak

#### 1.3.1.1 LED bat kontrolatzen duen makina

Adibide honetan sarrera bezala zeroak eta batekoak jasoko dituen makina bat erakutsiko da (1.3.1 irudia). Irteera argia igortzen duen diodo bat (LED bat)<sup>1</sup> izango da. Batekoa jasotzen den bakoitzean LEDa piztu egin behar da eta zeroa jasotzen den bakoitzean LEDa itzali egin behar da. Makina “Itzalita” eta “Piztuta” izeneko bi egoeren bidez kontrolatuko da. “Itzalita” izeneko egoerara ezkerretik iristen den geziaren bidez, makina martxan jartzean egoera horretatik abiatuko garela adierazten da. Beraz, makina martxan jartzean LEDa itzalita egongo da. “Itzalita” izeneko egoeratik egoera berera iristen den geziaren bidez, zeroak datozen bitartean makina egoera horretantxe mantenduko dela adierazten da. Bateko bat irakurtzen bada, “Piztuta” egoerara igaroko da eta LEDa piztu egingo da. Makina “Piztuta” egoeran dagoela bateko bat iristen bada, egoera horretan jarraituko du. Aldiz, zero bat iristen bada, “Itzalita” egoerara igaroko da eta LEDa itzali egingo da.

---

<sup>1</sup>LED: Light-Emitting Diode



**1.3.1 irudia.** LED baten piztea eta itzaltzea kontrolatzen duen egoera-makina.

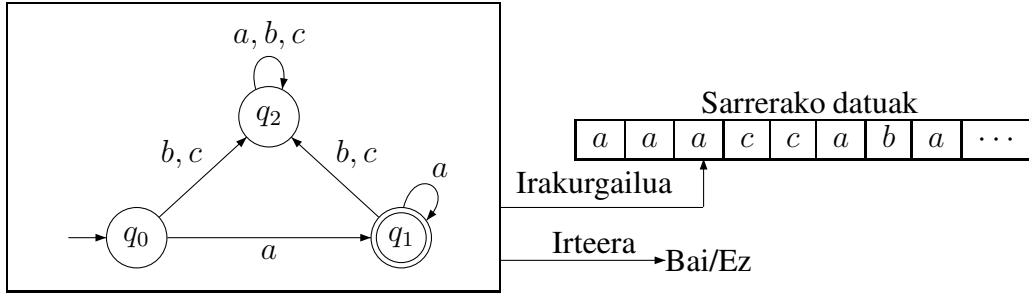
### 1.3.1.2 Hutsa ez den sekuentzia bateko osagai denak $a$ al diren erabakitzen duen makina

Adibide honetako makinak (1.3.2 irudia)  $a$ ,  $b$  eta  $c$  sinboloak erabiliz osatutako sekuentziak hartzen ditu sarrera bezala. Sekuentzia bat jasotakoan, hutsa ez den  $a$ -z osatutako sekuentzia al den erabakiko du makinak. Baiezko kasuan “Bai” eta ezezko kasuan “Ez” erantzun beharko du. Beraz irteera “Bai” edo “Ez” izango da. Makinaren kontrola hiru egoeren bidez antolatuta dago. Hasierako egoera  $q_0$  da. Hori 1.3.2 irudian  $q_0$  egoerara ezkerretik iristen den geziaren bidez adierazten da. Karaktere-kate bat edo sekuentzia bat irakurtzen hasterakoan  $q_0$  egoeran egongo gara. Lehenengo karakterea  $a$  baldin bada, makina  $q_1$  egoerara igaroko da. Lehenengo karakterea  $b$  edo  $c$  baldin bada, makina  $q_2$  egoerara igaroko da. Makina  $q_1$  egoeran dagoenean,  $a$  karakterea etortzen bada, makina egoera berean mantenduko da, hau da,  $q_1$  egoeran. Baina  $b$  edo  $c$  sinboloa etortzen bada,  $q_2$  egoerara igaroko da. Makina  $q_2$  egoeran baldin badago, etortzen den sinboloa etortzen dela ere, makina egoera berean mantenduko da,  $q_2$  egoeran. Karaktere-katea irakurtzea bukatutakoan makina  $q_1$  egoeran baldin badago, katea ez dela hutsa eta gainera  $b$  eta  $c$  sinboloak ez direla agertu esan nahiko du horrek eta, ondorioz,  $a$ -z osatutako katea da. Baina katea irakurtzea bukatu ondoren makina  $q_2$  egoeran baldin badago, katean gutxienez  $b$  bat edo  $c$  bat badagoela esan nahiko du horrek eta, beraz, katea ez dagoela bakarrik  $a$ -z osatuta. Karaktere-katea hutsa denean makina  $q_0$  egoeran geldituko da. Ondorioz, katea bukatutakoan makina  $q_1$  egoeran gelditu bada, irteera bezala “Bai” aurkeztu beharko da, baina katea bukatutakoan makina  $q_1$  ez den egoeraren batean gelditu bada, hau da,  $q_0$  edo  $q_2$  egoeran gelditu bada, irteera bezala “Ez” aurkeztu beharko da. Erantzuna baiezkoa izateko makinak  $q_1$  egoeran bukatu behar duenez, egoera hori zirkunferentzia bikoitzarekin adierazi da.

### 1.3.1.3 Bukaeran zuriunea egongo dela jakinda, $\#$ sinboloarekin bukatzen diren eta 1-ez osatuta dauden bi kate elkartzen dituen makina.

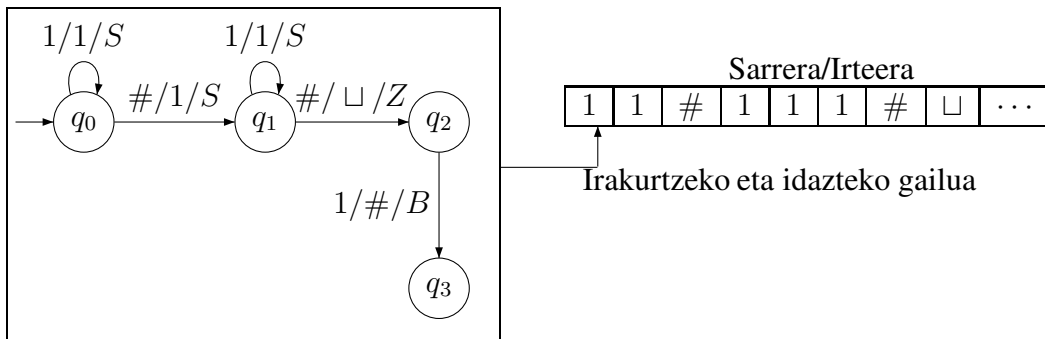
Adibide honetako makinak 1-ez osatutako bi sekuentzia jasoko ditu sarrera bezala. Sekuentzia biak  $\#$  sinboloarekin bukatuko dira eta gainera bigarren  $\#$  sinboloaren ostean zuriune bat egongo da. Zuriunea  $\sqcup$  sinboloaren bidez adieraziko dugu. Makina 1.3.3 irudian erakusten da. Irudi horretan ikus daitekeen bezala, makinaren kontrola lau<sup>2</sup> egoeren bidez antolatuko da:  $q_0$ ,  $q_1$ ,  $q_2$

<sup>2</sup>1.3.4 irudian ikus daitekeen bezala, hiru egoerako makina batekin ere egin daiteke.



**1.3.2 irudia.** Hitza hutsa ez dela eta bakarrik  $a$ - $z$  osatuta al dagoen erabakitzen duen makina.

eta  $q_3$ . 1.3.1.1 eta 1.3.1.2 adibideetako trantsizioetan (gezietan) sinbolo bakarra ageri da. Sinbolo horren bidez sinbolo hori irakurtzen bada, geziak adierazten duen egoerara joan behar dela jakinarazten da. Gainera 1.3.1.1 eta 1.3.1.2 adibideetan irakurgailua beti eskuinerantz mugitzen da. Baina adibide honetako makinako trantsizioetan  $\alpha/\beta/\gamma$ <sup>3</sup> erako hirukoteak agertzen dira. Hirukote horietan  $\alpha$  eta  $\beta$  Sarrera/Irteeran ager daitezkeen sinboloen multzokoak dira, hau da,  $\{1, \#, \sqcup\}$  multzokoak. Baina  $\gamma$  osagaia  $\{S, Z, B\}$  multzokoa da:  $S$  sinboloak “irakurtzeko eta idazteko gailua posizio bat eskuinera” mugitzeko adierazten du,  $Z$  sinboloak “irakurtzeko eta idazteko gailua posizio bat ezkerrera” mugitzeko adierazten du eta  $B$  sinboloak “irakurtzeko eta idazteko gailua posizio berean mantentzeko” adierazten du. Beraz  $\alpha/\beta/\gamma$  erako hirukote batek irakurtzeko eta idazteko gailua dagoen posizioan  $\alpha$  sinboloa baldin badago, posizio horretan  $\beta$  idazteko eta irakurtzeko eta idazteko gailuak  $\gamma$  sinboloak adierazten duen mugimendua egiteko esan nahi du. Gainera geziak apuntatzen duen egoerara igaro beharko da. Adibide honen bidez emaitzak idaztea eta irakurtzeko eta idazteko gailuaren mugimendua kontrolatzea posible dela erakusten da.



**1.3.3 irudia.** Bukaeran zuriunea egongo dela jakinda, 1-ez osatutako eta  $\#$  sinboloaz bukatutako bi kate elkartzen dituen makina.

Hasieran makina  $q_0$  egoeran egongo da eta irakurtzeko eta idazteko gailua Sarrera/Irteerako

<sup>3</sup> $\alpha/\beta/\gamma$ : alfa/beta/gamma

gailuko lehenengo posizioan egongo da. Sarrera beti egokia izango dela suposatuko dugu, hau da, 1-ez osatutako eta # sinboloaz bukatutako bi sekuentzia emango zaizkigu eta bukaeran zuriune bat egongo da. Horrela  $q_0$  egoeran gaudenean, 1 irakurtzen bada, leku berean 1 idatziko da eta irakurtzeko eta idazteko gailua eskuinera mugituko da posizio bat eta makina egoera berean mantenduko da,  $q_0$  egoeran. # sinboloa lehenengo aldiz irakurritakoan, hau da, 1-ez osatutako lehenengo sekuentzia bukatutakoan, posizio horretan 1 idatziko da (# sinboloa ezabatuz) eta, jarraian, irakurtzeko eta idazteko gailua eskuinera mugituko da posizio bat. Gainera makina egoeraz aldatuko da,  $q_1$  egoerara igaroz. Makina  $q_1$  egoerara iristen denean 1-ez osatutako bigarren sekuentzia irakurtzen hasiko da. Egoera horretan 1-eko bat irakurtzen bada, posizio berean 1-eko bat idatziko da eta irakurtzeko eta idazteko gailua eskuinera mugituko da posizio bat. Makina egoera berean mantenduko da,  $q_1$  egoeran. # sinboloa bigarren aldiz agertzen denean, sinbolo hori zuriuneaz ordezkatzeko da ( $\sqcup$  sinboloa erabiliko da zuriunea adierazteko) eta irakurtzeko eta idazteko gailua ezkerrera posizio bat mugituko da, sekuentziako azkeneko 1-ekoan kokatuz. Gainera makina  $q_2$  egoerara igaroko da.  $q_2$  egoeran berriro azkeneko 1-ekoa irakurriko da (hor kokatuta baikaude), bere lekuan # idatziko da eta makina  $q_3$  egoerara igaroko da, egin beharreko eragiketak bukatuz.

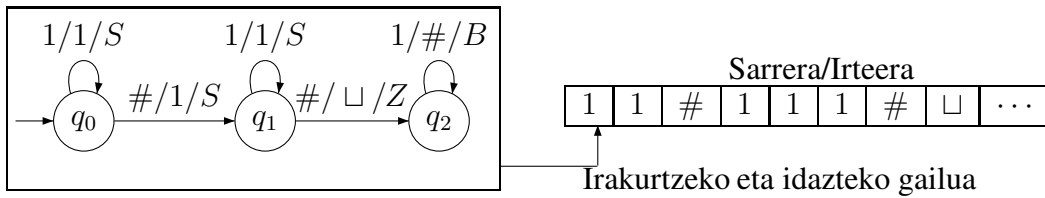
1.3.3 irudian agertzen den sarrera ematen badigute

1	1	#	1	1	1	#	$\sqcup$	...
---	---	---	---	---	---	---	----------	-----

irteerak honako honek izan beharko luke:

1	1	1	1	1	#	$\sqcup$	$\sqcup$	...
---	---	---	---	---	---	----------	----------	-----

Gauza bera egiten duen baina bakarrik hiru egoera dituen makina dugu 1.3.4 irudian.



**1.3.4 irudia.** Bukaeran zuriunea egongo dela jakinda, 1-ez osatutako eta # sinboloaz bukatutako bi kate elkartzen dituen hiru egoerako makina

## 1.3.2 Egoeratan oinarritutako makinetako memoria

1.3.1.1 atalean aurkeztutako adibidean, makina “Piztuta” egoeran dagoen bitartean LEDa piztuta mantenduko da eta makina “Itzalita” egoeran dagoen bitartean LEDa itzalita mantenduko da. Adibide horretan makinaren memoria egoeren bidez lortzen da, uneko egoeraren bidez irakurri den azkeneko sinboloa zein izan den gogora baitezake eta horrela LEDa piztuta edo itzalita mantendu behar al duen jakin dezake.

1.3.1.2 atalean aurkeztutako adibidean ere, makinaren memoria egoerak dira, egoeren bidez une bakoitzean ordura arte irakurritako sinbolo denak  $a$ -k izan al diren ala ez gogora baitezake. Horrela  $q_1$  egoeran baldin bagaude badakigu orain arte  $a$ -z osatutako sekuentzia irakurri dela, baina  $q_2$  egoeran baldin bagaude, badakigu  $b$  edo  $c$  gutxienez behin agertu dela. Gainera, sekuentzia osoa bukatutakoan  $q_1$  egoeran baldin bagaude “Bai” erantzun behar da eta  $q_1$  egoeran ez bagaude, “Ez” erantzun beharko da. Beraz, egoerek memoriaren lana egiten dute.

1.3.1.3 atalean azaldutako adibidean, egoerek memoria papera betetzen dute, izan ere beraien bidez 1-ez osatutako lehenengo sekuentzian ala bigarren sekuentzian al gauden eta # sinboloa agertzean, bere lehenengo ala bigarren agerpenean gauden jakin baitezakegu. Horrela  $q_0$  egoeran gaudenean badakigu 1-ez osatutako lehenengo sekuentzian gaudela eta  $q_1$  egoeran gaudenean badakigu 1-ez osatutako bigarren sekuentzian gaudela. Era berean,  $q_0$  egoeran gaudela # agertzen bada, badakigu # sinboloaren lehenengo agerpenaren aurrean gaudela, eta  $q_1$  egoeran gaudela # agertzen bada, badakigu # sinboloaren bigarren agerpenaren aurrean gaudela. Baina aurreko adibideetan ez bezala, adibide honetan Sarrera/Irteerako gailuak ere memoria papera du, emaitza bertan gordetzen baita.

Laburtuz, kalkuluak egiteko hain garrantzitsua den memoria egoeren eta Sarrera/Irteerako gailuaren artean banatuta egongo da. Memoriak kalkuluak egiteko ahalmena ematen du. Zenbat eta memoria gehiago, orduan eta kalkulu-ahalmen handiagoa (neurri batean behintzat). Beraz, erabil daitekeen memoriaren arabera kalkulu-ahalmen handiagoa edo txikiagoa duten makinak garatu ahal izango dira.

### 1.3.3 Egoeratan oinarritutako makinetan sarrera eta irteerako datuak adierazteko era: Lengoaiak

Sarrera eta irteerako datuak karaktere-kateak izango dira. Makina bakoitzaren kasuan karaktere-kateak osatzerakoan zein sinbolo erabili ahal izango diren zehaztuko da. Beraz, makina hauek kalkuluak egitea sarrera gisa emandako karaktere-kateak prozesatu eta irteerako emaitza lortzea izango da, 1.3.1.1, 1.3.1.2 eta 1.3.1.3 ataletako adibideetan ikusi den moduan. Konputazioa karaktere-kateen prozesamendu gisa planteatzea oso naturala da. Izan ere, ordenagailuak egiten dituen lanak karaktere-kateen prozesaketa moduan uler daitezke. Adibidez, programa bat exekutatu nahi dugunean, ordenagailuari programa eta sarrerako datuak emango dizkiogu sarrera bezala. Bai programa eta bai sarrerako datuak azken batean karaktere-kateak dira. Ordenagailuak programa eta sarrerako datuak hartu eta programaren bidez adierazitako aginduak exekututuko ditu sarrerako datuak kontuan hartuz. Egoera-makinetan oinarritutako konputazioereduan lengoaia bat karaktere-katez osatutako multzo bat izango da eta, aurrerago azalduko den bezala, makina bakoitzari lengoaia bat egokituko zaio.

### 1.3.4 Konputagarritasunaren azterketa egoeratan oinarritutako makinaren bidez

1.3.1.1, 1.3.1.2 eta 1.3.1.3 ataletako adibideetan erakutsi den bezala, egoeratan oinarritutako makinek ezaugarri desberdinak izan ditzakete:

- Era desberdinetako irteerak sortzeko aukera (argizko seinaleak, “Bai”/“Ez” erako erantzunak eta kalkuluren baten bidez lortutako karaktere-kateak).
- Sarrerako datuak behin bakarrik irakurri ahal izatea (irakurgailuari atzera egitea galaraziz) edo sarrerako datuak behin baino gehiagotan irakurtzeko aukera izatea (irakurgailuari atzera egitea ahalbidetuz).
- Era eta mota desberdinetako memoriaz baliatzeko aukera: memoria moduan bakarrik egoerak erabiliz edo egoerez gain memoria-gailu gehigarriak erabiliz (adibidez, Sarre-ra/Irteerako gailua memoria gisa erabiltzea ahalbidetuz, bertako datuak behin baino gehiagotan irakurtzea ahalbidetuz eta bertan emaitzak gordetzea ahalbidetuz).
- Memoria finitua (memoria egoerek bakarrik osatzen dutenean, egoera kopurua beti finitua izango baita) edo infinitua izan daitekeen memoria (egoerez gain beste gailu gehiagarri batzuk ere memoria gisa erabil daitezkeenean) izateko aukera.

Makinen beste ezaugarri garrantzitsuetako bat “Determinista” edo “Ez-determinista” izatea da. Makina deterministetan sarrerako datu bati dagokion kalkuluak bide bakarra jarraituko du. Bide bakar hori sarrerako datua eta makinaren egituraren arabera determinatuta edo finkatuta egongo da. Beraz, problema ebazteko estrategia bakarra edukiko da. Aldiz, makina ez-deterministetan, sarrerako datu bati dagokion kalkuluak egiteko bide edo aukera desberdinak planteatu daitezke. Beraz, problema ebazteko, oro har, estrategia bat baino gehiago edukiko da. Estrategia horietako batzuk okerrak izango dira emandako datuarentzat (ez baitute zuzenean emaitza egokia lortzeko balioko) baina planteatutako estrategietako batek ondo funtzionatzea (zuzenean emaitza egokia lortzeko balio izatea) nahikoa izango da kalkuluaren eta makinaren zuzentasuna bermatzeko. Problema bat ebazteko balio duen bertsio ez-deterministako estrategia desberdinak bertsio deterministako estrategia bakarra deskonposatuz lortuko dira gehienetan eta, horregatik, bertsio ez-deterministako strategiak desberdinak bertsio deterministako strategiak bakarraren azpiestrategiak bezala uler daitezke. Hori dela eta, gerta daiteke makina ez-deterministetan planteatu diren estrategia denak probatuz joatearen beharra edukitzea (azpies-trategiak baitira eta ez estrategia osoak). Aukeratuz goazen strategiak okerrak izatea gertatzen bada, oraindik aukeratu gabe gelditzen direnetako beste bat aukeratu beharko da. Kasu txarrean estrategia denak aukeratu beharko dira baina, hala ere, estrategia desberdinak planteatzeak, estrategia bakoitza makina desberdin baten bidez exekutatzeak aukera ematen du. horrela, exekuzio horiek paraleloan egin ahal izango dira eta, oro har, kalkuluak burutzeko denbora aurretuko da.

2. gaian lengoaiari buruzko oinarritzko kontzeptuak aurkeztuko dira, lengoia bat karaktere-kate edo hitz finituz osatutako multzo finitu edo infinitua izango dela kontuaz hartuz. Hitzen



gaineko eta lengoaien gaineko eragiketak definituko dira eta lengoaiak era formalean, hau da, notazio matematikoa erabiliz, nola definitu azalduko da.

3. gaian egoeratan oinarritutako era desberdinetako makinak aurkeztuko dira. Makina horiek bakarrik “Bai”/“Ez” erako erantzunak emango dituzte eta memoria finitua izango dute (memoria bezala bakarrik egoerak erabiliko baitituzte). Automata finituak dira makina horiek hain zuzen ere. Aztertuko diren automata finitu batzuk aldi berean kalkulu bat baino gehiago egiteko aukera emango dute (determinismoa / ez-determinismoa), baina automata finituen kasuan ezaugarri horrek kalkulu ahalmena ez duela handituko frogatuko da. 3. gaia bukatzeko, makina horiek (automa finituak) ez direla nahikoa (kalkulu batzuk egiteko gai ez direla) erakutsiko da eta ahalmen handiagoko (infinetua izan daitekeen memoria gehigarria duten) beste makina batzuk definitzea beharrezkoa dela azalduko da. Memoria-ahalmen handiagoko makina erakusgarri bezala piladun automatak aurkeztuko dira. Piladun automatek infinitua izan daitekeen memoria gehigarria erabiltzen dute. Memoria gehigarri horrek pilen ezaugarriak izango ditu, hau da, datu berriak gailurrean bakarrik ipin daitezke eta bakarrik gailurrean dagoen datua eskura daiteke une bakoitzean.

4. gaian orokorrenak diren makinak, hau da, Turing-en makinak aurkeztuko dira. Makina horiek “Bai”/“Ez” erako erantzunez gain erantzun bezala karaktere-kateak lortuko dituzten kalkuluak ere egin ditzakete. Erantzun bezala lortutako karaktere-kate horiek Sarrera/Irteera gailuan gordeko dira. Sarrera/Irteerako gailuak infinitua izan daitekeen eta datuak eskuratzean edo gordetzean pila batek dituen murriztapenak ez dituen memoria bezala jokatu du. Gainera, gai honetan problema erabakigarria, erabakiezina, bereizgarria eta bereiztezinaaren kontzeptuak aurkeztuko dira. Problema erabakigarrien kasuan onargarriak diren datu denentzat kalkulatu ahal izango da emaitza. Erabakiezinak baina bereizgarriak diren problemen kasuan, onargarriak diren datu batzuentzat bakarrik kalkulatu ahal izango da emaitza. Problema bereiztezinetan, oro har, ezingo da lortu emaitzik. Sailkapen horretatik problemak konputagarri, erdikonputagarri eta konputaezin gisa sailkatzera iritsiko gara.