



Lengoaia eta Sistema Informatikoak Saila

Bilboko Industria Ingeniaritza Teknikoko Unibertsitate Eskola

Lengoaiak, Konputazioa eta Sistema Adimendunak

Kudeaketaren eta Informazio Sistemen Informatikaren Ingeniaritzako Gradua

2. Maila

2015-16 ikasturtea

1. Gaia: Sarrera

JOSE GAINZARAIN IBARMIA

Azken eguneraketa: 2015-9-2

GAIEN AURKIBIDEA

1. Sarrera	1
1.1 Motibazioa	1
1.2 Egoeratan eta egoeren arteko trantsizioetan oinarritutako makinak	2
1.2.1 Adibidea: LED bat kontrolatzen duen makina	2
1.2.2 Adibidea: Hutsa ez den sekuentzia bateko osagai denak a al diren erabakitzen duen makina	2
1.2.3 Adibidea: Bukaeran zuriunea egongo dela jakinda, $\#$ sinboloarekin bukatzen diren eta 1-ez osatuta dauden bi kate elkartzan dituen makina.	3
1.2.4 Egoeratan oinarritutako makinetako memoria	4
1.2.5 Egoeratan oinarritutako makinetan sarrera eta irteerako datuak adierazteko era: Lengoaiak	5

IRUDIEN ZERRENDA

1.1	LED baten piztea eta itzaltzea kontrolatzen duen egoera-makina.	2
1.2	Hitza hutsa ez dela eta bakarrik <i>a-z</i> osatuta al dagoen erabakitzen duen makina.	3
1.3	Bukaeran zuriunea egongo dela jakinda, 1-ez osatutako eta # sinboloaz bukatutako bi kate elkartzen dituen makina.	4
1.4	Bukaeran zuriunea egongo dela jakinda, 1-ez osatutako eta # sinboloaz bukatutako bi kate elkartzen dituen <u>hiru egoerako</u> makina	5

1. SARRERA

“Lengoaiak, Konputazioa eta Sistema Adimendunak” irakasgaian konputagarritasunaren azterketa formala landuko da eta konputazioaren oinarriak aurkeztuko dira. Horretarako, alde batetik, 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 diren bide berrien eta indarra hartzen ari diren ikuspegi berrien nondik norakoa hobeto ulertzeko balioko digu.

1.1 Motibazioa

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

Informatikaren azken helburua gizakiek planteatzen dituzten 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 aztertzerako 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 burutzerakoan era desberdinak edo eredu desberdinak erabili dira. Garrantzitsuenak honako hauek dira:

- Turing-en Makina (egoeratan eta egoeren arteko trantsizioetan oinarritua).
- λ -kalkulua (funtzio matematikoetan oinarritua).
- Funtzio errekursiboak.
- 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)
- λ -kalkulua: Funtzio bidezko programazioa (Haskell, Scheme, ML, eta abar)
- Funtzio errekursiboak: Programazio algebrakoa (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 λ -kalkuluan oinarrituta dagoen Haskell lengoia erabiliko dugu. Horrela, aipatu diren konputazio-ereduetako bi landu ahal izango ditugu.

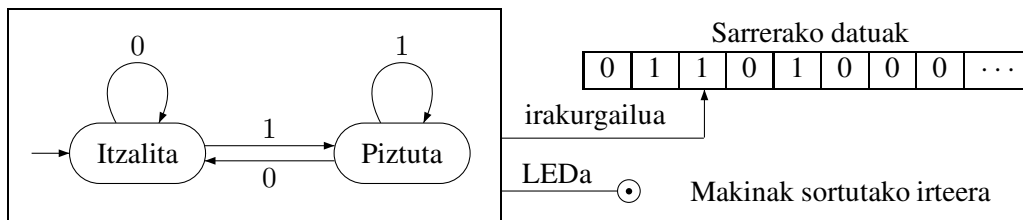
1.2 Egoeratan eta egoeren arteko trantsizioetan oinarritutako makinak

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 egoerataro 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 kalkulatu-prozesuan zehar edo bukaeran erakutsiko da irteera bezala.
- Informazioa gordetzeko balioko duen **memoria**.

1.2.1 Adibidea: LED bat kontrolatzen duen makina

Adibide honetan sarrera bezala zeroak eta batekoak jasoko dituen makina bat erakutsiko da (1.1 Irudia). Irteera argia igortzen duen diodo bat (LED bat)¹ 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.



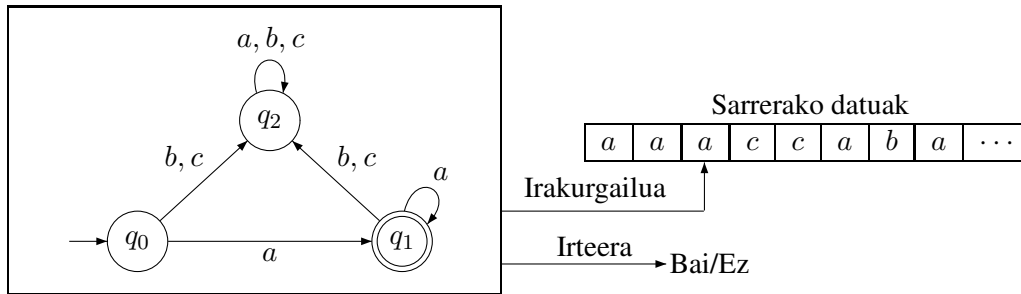
1.1 Irudia: LED baten piztea eta itzaltzea kontrolatzen duen egoera-makina.

1.2.2 Adibidea: Hutsa ez den sekuentzia bateko osagai denak a al diren erabakitzen duen makina

Adibide honetako makinak (1.2 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

¹ LED: Light-Emitting Diode

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 1.2 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 agertzen esan nahiko du, beraz 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 nahi 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.



1.2 Irudia: Hitza hutsa ez dela eta bakarrik a -z osatuta al dagoen erabakitzen duen makina.

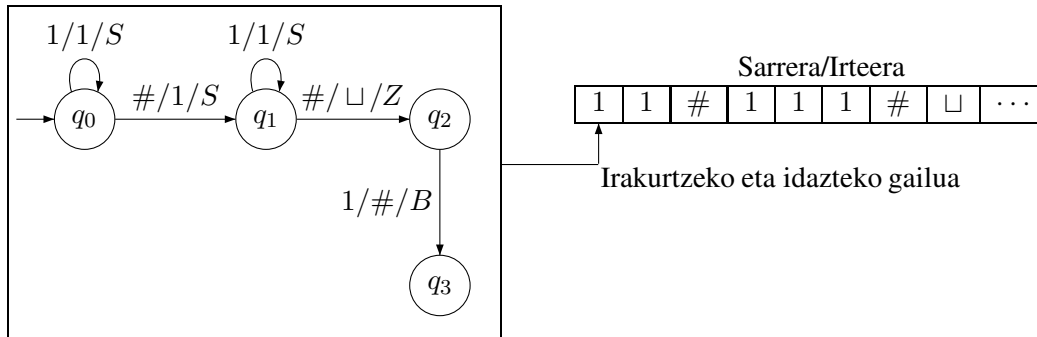
1.2.3 Adibidea: Bukaeran zuriunea egongo dela jakinda, # sinboloarekin bukatzen diren eta 1-ez osatuta dauden bi kate elkartzen dituen makina.

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. Makinaren kontrola lau² egoeren bidez antolatuko da: q_0 , q_1 , q_2 eta q_3 . 1.2.1 eta 1.2.2 adibideetako trantsizioetan (geziatan) sinbolo bakarria ageri da. Sinbolo horren bidez sinbolo hori irakurtzen bada, geziak adierazten duen egoerara joan behar dela jakinarazten da. Gainera 1.2.1 eta 1.2.2 adibideetan irakurgailua beti eskuinerantz mugitzen da. Baina adibide honetako makinako trantsizioetan $\alpha/\beta/\gamma$ ³ erako hirukoteak agertzen dira. Hirukote horietan α eta β Sarrera/Irteeran ager daitezkeen sinboloen multzokoak dira, hau da, $\{1, \#, \sqcup\}$ multzokoak. Baina γ 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 α sinboloa baldin badago, posizio horretan β idazteko eta irakurtzeko eta idazteko gailuak γ sinboloak adierazten duen mugimendua burutzeko esan nahi du. Gainera geziak apuntatzen

² 1.4 Irudian ikus daitezkeen bezala, hiru egoerako makina batekin ere egin daiteke.

³ $\alpha/\beta/\gamma$: alfa/beta/gamma

duen egoerara pasatu beharko da. Adibide honen bidez emaitzak idaztea eta irakurtzeko eta idazteko gailuaren mugimendua kontrolatzea posible dela erakusten da.



1.3 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 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 (□ sinboloa erabiliko da zuriunea adierazteko) eta irakurtzeko eta idazteko gailua ezkerreko posizio bat mugituko da, sekuentziako azkeneko 1-ekoan kokatuz. Gainera makina q_2 egoerara pasatuko da. q_2 egoeran berriro azkeneko 1-ekoa irakurriko da (hor kokatuta baikaude), bere lekuan # idatziko da eta makina q_3 egoerara pasatuko da, egin beharreko eragiketak bukatuz.

1.3 irudian agertzen den sarrera ematen badigute

1	1	#	1	1	1	#	□	...
---	---	---	---	---	---	---	---	-----

irteerak honako honek izan beharko luke:

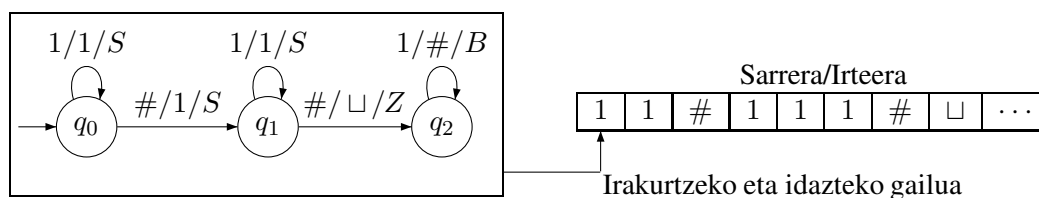
1	1	1	1	1	#	□	□	...
---	---	---	---	---	---	---	---	-----

Gauza bera egiten duen baina bakarrik hiru egoera dituen makina dugu 1.4 Irudian.

1.2.4 Egoeratan oinarritutako makinetako memoria

1.2.1 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.

1.2.2 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



1.4 Irudia: Bukaeran zuriunea egongo dela jakinda, 1-ez osatutako eta # sinboloaz bukatutako bi kate elkartzen dituen hiru egoerako makina

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 burutzen dute.

1.2.3 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.

1.2.5 Egoeratan oinarritutako makinetan sarrera eta irteerako datuak adierazteko era: Lengoiak

Sarrera eta irteerako datuak karaktere-kateak izango dira. Makina bakoitzaren kasuan karaktere-kateak osatzerakoan zein sinbolo erabili ahal izango diren zehaztuko da. Beraz makina hauetan kalkuluak burutzea sarrera bezala emandako karaktere-kateak prozesatu eta irteerako emaitza lortzea izango da, 1.2.1, 1.2.2 eta 1.2.3 Ataletako adibideetan ikusi den bezala. Konputazioa karaktere-kateen prozesamendua bezala planteatzea oso naturala da. Izan ere, ordenagailuak burutzen dituen lanak karaktere-kateen prozesaketa bezala uler daitezke. Adibidez, programa bat exekutatu nahi dugunuean, ordenagailuari programa eta sarrerako datuak ematen 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 burutzen ditu sarrerako datuak kontuan hartuz. Egoera-makinetan oinarritutako konputazio-ereduan lengoia bat karaktere-katez osatutako multzo bat izango da eta, aurrerago azalduko den bezala, makina bakoitzari lengoia bat egoikituko zaio.