eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

Lengoaiak, Konputazioa eta Sistema Adimendunak

Kudeaketaren eta Informazio Sistemen Informatikaren Ingeniaritzako Gradua

Bilboko Ingeniaritza Eskola (UPV/EHU)

2. maila

2019-2020 ikasturtea

3. gaia: Automata finituak eta lengoaia erregularrak

JOSÉ GAINTZARAIN IBARMIA

Lengoaia eta Sistema Informatikoak Saila

Azken eguneraketa: 2019 - 08 - 30

Aurkibidea

3. gaia

Automata finituak eta lengoaia erregularrak

3.1.

Sarrera

Gai honetan automata finituak azalduko ditugu. "Automata finitu" era laburrean esateko eta idazteko AF inizialak erabiliko ditugu. Automata finituak egoeratan oinarritutako makinak dira. Lehenengo gaian ikusi genituen adibideetan, makinen memoria-ahalmena egoeren eta Sarrera/Irteerako gailuen bidez finkatzen zela esan genuen. Lehenengo bi adibideetan (1.2.1 eta 1.2.2 atalak, 1.1 eta 1.2 irudiak) memoria egoeren bidez inplementatzen genuen eta hirugarrengoan (1.2.3 atala, 1.3 edo 1.4 irudia) egoeren eta Sarrera/Irteerako gailuaren bidez inplementatzen genuen memoria. Izan ere, hirugarren adibide horretan emaitza gordez joateko Sarrera/Irteerako gailua erabiltzen genuen. Gai honetan landuko ditugun automata finituetan, egoerak erabiliko dira memoria gisa. Egoera-kopurua finitua izango denez, memoria ahalmena ere mugatua izango da eta ondorioz kalkuluak burutzeko gaitasuna ere mugatua izango da.

Automata finitu deterministak (AFD), automata finitu ez-deterministak (AFED) eta ε trantsizioak dituzten automata finitu ez-deterministak (ε-AFED) bereiziko ditugu eta hiru automata mota horiek baliokideak direla ere frogatuko dugu. Gainera, beraien bidez defini daitezkeen lengoaiak, hau da, lengoaia erregularrak ere aurkeztuko ditugu.

Automata finitu bakoitza lengoaia bati lotuta doa. Automata finitu bati hitz bat ematen zaionean, automatak hitz hori berari lotutako lengoaiakoa al den ala ez erabakiko du eta "Bai" edo "Ez" erantzungo du.

Jarraian automata finituak era zehatzagoan definituko dira.

3.2.

Automata Finitu Deterministak (AFDak)

3.2.1 AFDen definizioa

3.2.1.1 AFDen definizio formala

Automata finitu determinista bat boskote bat da (bost osagai dituen egitura bat da):

$$(Q, A, \delta, q_0, Y)$$

Bost osagai horien definizioa honako hau da:

- Q egoerez osatutako multzo finitua da: $q_0, q_1, q_2, q_3, \ldots$
- A alfabetoa da
- δ trantsizio-funtzio determinista da. Q multzoko egoera bat eta A multzoko sinbolo bat emanda, egoera horretan gaudenean sinbolo hori irakurtzen badugu zein egoeratara joango garen adierazten du δ trantsizio-funtzioak. δ funtzioaren mota honako hau da:

$$\delta : Q \times A \rightarrow Q$$

- q_0 hasierako egoera da. Hasierako egoera beti bakarra izango da.
- Y osagaia “Bai” erantzungo duten egoerez osatutako multzoa da. Y multzoa Q multzoaren azpimultzoa izango da, hau da, $Y \subseteq Q$, edo beste era batera esanda, $Y \in 2^Q$, hor 2^Q elementua Q multzoaren azpimultzo guztiez osatutako multzoa izanda. Y multzoa **onarpen egoeren multzoa** da eta Y multzoko elementuak **onarpen egoerak** dira. AFD batek berari dagokion lengoaiako hitz bat irakurtzen duenean, Y multzoko egoera batean bukatuko du. Aldiz, bere lengoaiakoa ez den hitz bat irakurtzen duenean, Y multzokoa ez den egoera batean bukatuko du.

AFD baten **funtzionamendua** honako hau da: hasieran AFDa q_0 egoeran egongo da eta sarrera gisa A^* multzoko hitz bat jasoko du. Hitz hori osatzen duten sinboloak banan-banan irakurriko ditu. Sinbolo bakoitza irakurtzean zein egoeratara pasatu erabakiko da δ trantsizio-funtzioa erabiliz. Beraz, q_i egoera batean gaudela A alfabetokoa den α sinboloa irakurtzen badugu eta $\delta(q_i, \alpha) = q_j$ baldin bada, q_j egoerara igaroko gara. Hitzak beti finituak izango direnez, hitz bat irakurtzeko prozesua ere finitua izango da. Hitza irakurtzea bukatutakoan Q multzokoa den egoeraren batean egongo gara. Egoera hori Y multzokoa ere badenean, automatak "Bai" erantzungo du, hitza automatari lotutako lengoaiakoa dela adieraziz. Baina hitza bukatzean Y-koa ez den egoeraren batean gelditu bagara, automatak "Ez" erantzungo du, hitza automatari lotutako lengoaiakoa ez dela adieraziz.

Gai honetan ikusiko ditugun AFDetan Sarrera/Irteerako gailuan ez da ezer idatziko, bakarrik irakurri egingo da. Gainera irakurgailuak Sarrera/Irteerako gailutik sinbolo bat irakurtzen duen bakoitzean, irakurgailua eskuinera mugituko da posizio bat.

q_j Q multzoko egoera bat izanda eta w A^* multzoko hitz bat izanda, (q_j, w) bikoteari **konfigurazio determinista** deituko diogu. Konfigurazio deterministak $Q \times A^*$ motakoak dira. (q_j, w) erako konfigurazio determinista baten bidez q_j egoeran gaudela eta oraindik w hitza irakurtzeko daukagula adierazten da. Adibidez, A alfabetoa $\{a, b, c\}$ baldin bada, oraingo konfigurazio determinista $(q_3, caaba)$ baldin bada eta trantsizio-funtzioak $\delta(q_3, c) = q_6$ betetzen dela adierazten badu, $caaba$ hitzeko lehenengo sinboloa irakurritakoan q_6 egoerara igaroko gara eta konfigurazio determinista berria $(q_6, aaba)$ izango da. Konfigurazio determinista berrian ikus daitekeen bezala, irakurtzeko gelditzen den hitza laburragoa da orain (irakurri den sinboloa desagertu egin baita).

Notazioa arintzeko, *konfigurazio determinista* idatzi beharrean *konfigurazioa* idatziko dugu automata finitu deterministei buruzko atal honetan zehar.

AFD batek burututako **konputazioa** edo kalkulu-prozesua δ trantsizio-funtzioan oinarritzen den δ^* funtzioaren bidez definitzen da. δ^* funtzioaren mota honako hau da:

$$\delta^* : Q \times A^* \to Q$$

Beraz, q_j egoera bat eta w hitz bat emanda (hau da, konfigurazio bat emanda), δ^* funtzioak egoera bat itzuliko du (q_j egoeratik abiatu eta w hitzeko sinbolo denak irakurri ondoren lortuko den egoera).

δ^* funtzioaren definizio formala honako hau da:

- $\delta^*(q_j, \varepsilon) = q_j$
- $\delta^*(q_j, \alpha w) = \delta^*(\delta(q_j, \alpha), w)$

Definizio horren bidez honako hau adierazten da: irakurtzeko gelditzen den hitza hutsa baldin bada, hau da, ε, orduan egoera berean jarraituko dugu. Irakurtzeko gelditzen den hitza hutsa ez bada, eta bere lehenengo sinboloa A-koa den α sinboloa baldin bada, δ trantsizio-funtzioa aplikatuko zaie oraingo q_j egoerari eta α sinboloari eta egoera berri batera igaroko gara. Egoera berrira igaro ondoren w-eko gainontzeko sinboloak irakurtzen jarraitu beharko da:

$$\delta^*(\underbrace{\delta(q_j,\alpha)}_{\substack{\text{egoera}\\ \text{berria}}},w)$$

Adibidez, demagun alfabetoa $A = \{a, b, c\}$ dela, oraingo konfigurazioa (q_3, aac) dela eta $\delta(q_3, a) = q_5$, $\delta(q_5, a) = q_2$ eta $\delta(q_2, c) = q_6$ direla. $\delta^*(q_3, aac)$ konputazioa urratsez urrats honela egingo litzateke:

- aac hitza hutsa ez denez, $\delta^*(q_3, aac) = \delta^*(\delta(q_3, a), ac)$.
  $\delta(q_3, a) = q_5$ denez, $\delta^*(\delta(q_3, a), ac)$ espresioa $\delta^*(q_5, ac)$ bezala geratuko da.
- ac hitza hutsa ez denez, $\delta^*(q_5, ac) = \delta^*(\delta(q_5, a), c)$.
  $\delta(q_5, a) = q_2$ dela kontuan hartuz, $\delta^*(\delta(q_5, a), c)$ espresiotik $\delta^*(q_2, c)$ espresioa lortuko da.
- c hitza hutsa ez denez, $\delta^*(q_2, c) = \delta^*(q_2, c\varepsilon) = \delta^*(\delta(q_2, c), \varepsilon)$.
  $\delta(q_2, c) = q_6$ denez, $\delta^*(\delta(q_2, c), \varepsilon)$ espresiotik $\delta^*(q_6, \varepsilon)$ espresioa lortuko da.
- ε hitz hutsa geratu zaigunez, $\delta^*(q_6, \varepsilon)$ espresioaren balioa q_6 da.

Beraz, $\delta^*(q_3, aac) = q_6$.

Automata hauek **deterministak** direla esaten da, egoera bakoitzetik sinbolo bakoitzarekin zein egoeratara joan behar den garbi zehaztua dutelako δ funtzioaren bidez. Egoera bakoitzean sinbolo bakoitzeko mugimendu-aukera bakarra da. Automata finitu **ez-deterministetan** egoera batzuetan sinbolo batzuekin zer egin behar den ez da adierazten eta kasu batzuetan sinbolo batekin egoera batetik egoera batera baino gehiagotara joateko aukera dago.

$D = (Q, A, \delta, q_0, Y)$ **AFD bati lotutako** $L(D)$ **lengoaia** D automatarengandik “Bai” erantzuna jasotzen duten hitzez osatutako lengoaia da. Beste era batera esanda, q_0-tik abiatuta Y multzokoa den egoera batean lagatzen gaituzten A^*-ko hitzez osatutako lengoaia da $L(D)$. Era formalean definizioa honako hau izango litzateke:

$$L(D) = \{w \mid w \in A^* \wedge \delta^*(q_0, w) \in Y\}$$

3.2.1.2 AFDen adibideak

3.2.1.2.1 Hutsak ez diren eta bakarrik a sinboloaren errepikapenez eratuta dauden hitzez osatutako lengoaiari dagokion AFDa

Adibide honetako AFDa (3.2.1 irudia begiratu) 1. gaiko bigarren adibideko (1.2.2 atala, 1.2 irudia) makina da. 1. gaian makina hori era informalean aurkeztu genuen baina orain era formalean aurkeztuko dugu gai honetan (3. gaian) emandako AFDen definizioa kontuan hartuz.

AFD honek a, b eta c sinboloak eduki ditzakeen hitz bat jasoko du sarrera gisa. AFDak, emandako hitza hutsa ez den eta bakarrik a sinboloaren errepikapenez eratuta dagoen hitza al den erabakiko du. Horrela bada, "Bai" erantzuna itzuliko du eta bestela "Ez" erantzuna itzuliko du. Lengoaiaren definizio formala honako hau da:

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

Gogora dezagun lengoaiak funtzioen bidez ere adieraz daitezkeela. Adibide honetako lengoaiaren kasuan aukera bat honako funtzio hau da:

$$f : A^* \to 2$$
$$f(w) = \begin{cases} 1 & \text{baldin eta } |w|_a = |w| \wedge |w| \geq 1 \\ 0 & \text{bestela} \end{cases}$$

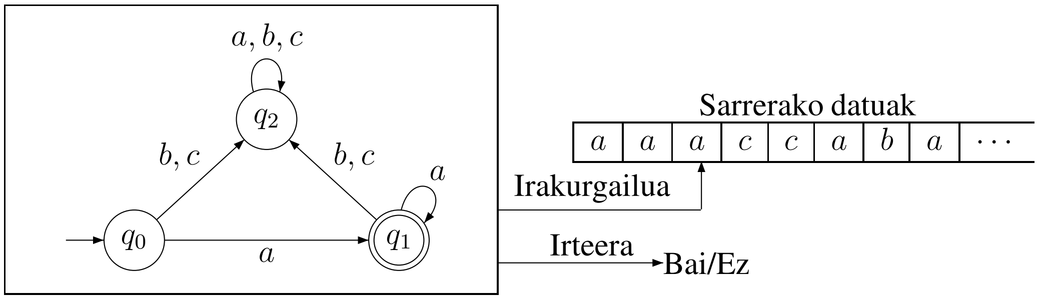
f funtzioaren definizio horretan, $2 = \{0, 1\}$ da, 0 balioa $True$ balioaren ordezkoa da eta 1 balioa $False$ balioaren ordezkoa da.

Beraz, irteera "Bai" edo "Ez" izango da. AFDa hiru egoerez osatuta dago. q_0 hasierako egoera da. Hitz bat edo karaktere-kate bat irakurtzen hasten garenean q_0 egoeran gaude beraz. Lehenengo karakterea (edo sinboloa) a baldin bada, AFDa q_1 egoerara igaroko da. Lehenengo karakterea b edo c baldin bada, AFDa q_2 egoerara igaroko da. AFDa q_1 egoeran dagoenean, a sinboloa etortzen bada, AFDa egoera berean mantenduko da, hau da, q_1 egoeran. Bestalde, q_1 egoeran egonda, b edo c etortzen bada, q_2 egoerara igaroko da. AFDa q_2 egoeran baldin badago, etortzen dena etortzen dela ere, egoera berean mantenduko da. Karaktere-kate osoa irakurritakoan AFDa q_1 egoeran baldin badago, horrek esan nahiko du karaktere-katea ez dela hutsa eta a sinboloaren errepikapenez osatuta dagoela. Aldiz, karaktere-kate osoa irakurritakoan AFDa q_2 egoeran baldin badago, horrek esan nahiko du karaktere-katea ez dela hutsa eta ez dagoela a sinboloaren errepikapenez bakarrik osatuta. Azkenik, karaktere-kate osoa irakurritakoan AFDa q_0 egoeran baldin badago, horrek esan nahiko du karaktere-katea hutsa dela. Beraz, q_1 egoeran bukatzen bada, jasotako hitza AFD honi dagokion lengoaiakoa da eta irteerako erantzuna "Bai" izango da, baina q_1 ez den egoeraren batean bukatzen bada, erantzuna "Ez" izango da, jasotako hitza AFD honi dagokion lengoaiako ez dela adieraziz. AFDari dagokion irudian q_1 egoera zirkulu bikoitzarekin agertzen da erantzuna baiezkoa izateko AFDak egoera honetan bukatu beharko duela adierazteko hain zuzen ere. Aipatzekoa da hitz bat jaso edo irakurritakoan q_0 egoeran bukatzeko aukera bakarra hitz hori hutsa izatea dela, hau da, emandako hitza hutsa baldin bada, orduan AFDak q_0 egoeran bukatuko du eta bestela q_1 egoeran edo q_2 egoeran bukatuko du.

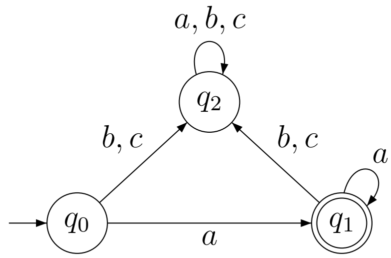
Gai honetan ikusiko ditugun AFD denek beti "Bai" edo "Ez" erantzuten dutenez eta sarrerako gailutik jasotako karaktere-katean ez dutenez inoiz aldaketarik egiten, AFDak diseinatzean egoerak eta egoeren arteko trantsizioak adierazteko balio duten geziak bakarrik ipiniko ditugu. Beraz, Sarrera/Irteerako gailuak ez ditugu ipiniko. Honela, 3.2.1 irudiko AFDaren kasuan, 3.2.2 irudian ikus daitekeena bakarrik ipiniko genuke.

Jarraian AFDaren osagaiak definituko dira boskote eran:

$$(Q, A, \delta, q_0, Y)$$



3.2.1 irudia. Hutsak ez diren eta bakarrik a sinboloaren errepikapenez eratuta dauden hitzei dagokien AFDa.



3.2.2 irudia. 3.2.1 irudiko AFDaren grafiko laburtua.

Osagai bakoitza honela geratuko da:

- $Q = \{q_0, q_1, q_2\}$
- $A = \{a, b, c\}$
- $\delta : Q \times A \rightarrow Q$ jarraian datorren taularen bidez definitutako trantsizio-funtzioa da:

δ	a	b	c
q_0	q_1	q_2	q_2
q_1	q_1	q_2	q_2
q_2	q_2	q_2	q_2

  Taula horren bidez $\delta(q_0, a)$-ren balioa q_1 dela, $\delta(q_0, b)$-ren balioa q_2 dela, eta abar adierazten da.

- q_0 hasierako egoera da. Hasierako egoera beti bakarra izango da.
- $Y = \{q_1\}$. Onarpen egoeren multzoa.

Orain (q_0, aaa) konfigurazioari dagokion konputazioa garatuko dugu urratsez urrats:

1. urratsa: $\delta^*(q_0, aaa) = \delta^*(\delta(q_0, a), aa) = \delta^*(q_1, aa)$

2. urratsa: $\delta^*(q_1, aa) = \delta^*(\delta(q_1, a), a) = \delta^*(q_1, a)$

3. urratsa: $\delta^*(q_1, a) = \delta^*(q_1, a\varepsilon) = \delta^*(\delta(q_1, a), \varepsilon) = \delta^*(q_1, \varepsilon)$

4. urratsa: $\delta^*(q_1, \varepsilon) = q_1$

Konputazioa Y multzokoa den q_1 egoeran amaitu denez, hitza lengoaiakoa da (ez da hutsa eta a sinboloaren errepikapenez eratutako dago).

AFDen konputazioak konfigurazioz osatutako sekuentziak bezala edo adar bakarreko zuhaitz gisa ere adierazi ohi dira:

$$\begin{array}{c} (q_0, aaa) \\ | \\ (q_1, aa) \\ | \\ (q_1, a) \\ | \\ (q_1, \varepsilon) \end{array}$$

Jarraian $(q_0, abca)$ konfigurazioari dagokion konputazioa garatuko dugu urratsez urrats:

1. urratsa: $\delta^*(q_0, abca) = \delta^*(\delta(q_0, a), bca) = \delta^*(q_1, bca)$

2. urratsa: $\delta^*(q_1, bca) = \delta^*(\delta(q_1, b), ca) = \delta^*(q_2, ca)$

3. urratsa: $\delta^*(q_2, ca) = \delta^*(\delta(q_2, c), a) = \delta^*(q_2, a)$

4. urratsa: $\delta^*(q_2, a) = \delta^*(q_2, a\varepsilon) = \delta^*(\delta(q_2, a), \varepsilon) = \delta^*(q_2, \varepsilon)$

5. urratsa: $\delta^*(q_2, \varepsilon) = q_2$

Konputazioa Y multzokoa ez den q_2 egoeran amaitu denez, hitza ez da AFD honi dagokion lengoaiakoa.

Konputazio hori konfigurazioz osatutako sekuentzia gisa ere adieraz daiteke:

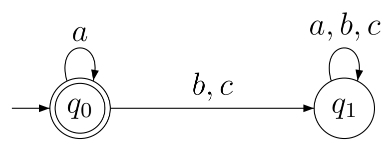
$$\begin{array}{c} (q_0, abca) \\ | \\ (q_1, bca) \\ | \\ (q_2, ca) \\ | \\ (q_2, a) \\ | \\ (q_2, \varepsilon) \end{array}$$

3.2.1.2.2 b-rik eta c-rik ez duten hitzez osatutako lengoaiari dagokion AFDa

Adibide honetako AFDak (3.2.3 irudian begiratu) a, b eta c sinboloak izan ditzakeen hitz bat jasoko du sarrera bezala. Emandako hitza b-rik eta c-rik ez duen hitz bat al den erabaki beharko du AFDak. Horrela, b-rik eta c-rik ez badu “Bai” erantzun beharko du eta bestela “Ez” erantzun beharko du. Lengoaiaren definizio formala honako hau da:

$$\{w \mid w \in A^* \wedge |w|_a = |w|\}$$

Adibide honetako AFDak bi egoera ditu. Hasierako egoera q_0 da eta ondorioz, hitz bat irakurtzen hasterakoan q_0 egoeran egongo da. Lehenengo karakterea b edo c baldin bada, AFDa q_1 egoerara igaroko da. AFDa q_1 egoeran baldin badago, etortzen dena etortzen dela ere egoera horretan bertan jarraituko du. Karaktere-katea irakurtzea bukatutakoan AFDa q_0 egoeran baldin badago, badakigu karaktere-kateak ez duela b-rik eta c-rik. Bestalde, karaktere-katea irakurtzea bukatutakoan AFDa q_1 egoeran baldin badago, badakigu karaktere-katean gutxienez b bat edo c bat badagoela. Beraz, q_0 egoeran bukatuz gero, hitza AFD honi dagokion lengoaiakoa izango da eta “Bai” erantzuna itzuliko da, baina q_1 egoeran bukatuz gero, erantzuna “Ez” izango da, irakurritako hitza ez baita AFD honi dagokion lengoaiakoa izango. q_0 egoerak zirkulu bikoitza du erantzuna baiezkoa izateko hor bukatu behar dela adierazteko. Hitz hutsaren kasuan AFDak q_0 egoeran bukatuko du eta ondorioz hitz hutsarentzat ere erantzuna baiezkoa izango da.



3.2.3 irudia. b-rik eta c-rik ez duten hitzez osatutako lengoaiari dagokion AFDa.

Jarraian AFDa boskote gisa definituko da.

$$(Q, A, \delta, q_0, Y)$$

Bost osagaiak honako hauek dira:

- $Q = \{q_0, q_1\}$
- $A = \{a, b, c\}$
- $\delta : Q \times A \to Q$ jarraian datorren taularen bidez definitutako trantsizio-funtzioa da:

δ	a	b	c
q_0	q_0	q_1	q_1
q_1	q_1	q_1	q_1

Taularen bidez $\delta(q_0, a)$-ren balioa q_0 dela, $\delta(q_0, b)$-ren balioa q_1 dela, eta abar adierazten da.

- q_0 hasierako egoera da. Hasierako egoera beti bakarra da.
- $Y = \{q_0\}$.

Orain (q_0, aaa) konfigurazioari dagokion konputazioa garatuko dugu urratsez urrats:

1. urratsa: $\delta^*(q_0, aaa) = \delta^*(\delta(q_0, a), aa) = \delta^*(q_0, aa)$

2. urratsa: $\delta^*(q_0, aa) = \delta^*(\delta(q_0, a), a) = \delta^*(q_0, a)$

3. urratsa: $\delta^*(q_0, a) = \delta^*(q_0, a\varepsilon) = \delta^*(\delta(q_0, a), \varepsilon) = \delta^*(q_0, \varepsilon)$

4. urratsa: $\delta^*(q_0, \varepsilon) = q_0$

Konputazioa Y multzokoa den q_0 egoeran amaitu denez, hitza lengoaiakoa da (ez du ez b-rik eta ez c-rik).

AFDen konputazioak konfigurazioz osatutako sekuentziak gisa edo adar bakarreko zuhaitz gisa ere adieraz daitezke:

$$\begin{array}{c} (q_0, aaa) \\ | \\ (q_0, aa) \\ | \\ (q_0, a) \\ | \\ (q_0, \varepsilon) \end{array}$$

Jarraian $(q_0, abca)$ konfigurazioari dagokion konputazioa garatuko dugu urratsez urrats:

1. urratsa: $\delta^*(q_0, abca) = \delta^*(\delta(q_0, a), bca) = \delta^*(q_0, bca)$

2. urratsa: $\delta^*(q_0, bca) = \delta^*(\delta(q_0, b), ca) = \delta^*(q_1, ca)$

3. urratsa: $\delta^*(q_1, ca) = \delta^*(\delta(q_1, c), a) = \delta^*(q_1, a)$

4. urratsa: $\delta^*(q_1, a) = \delta^*(q_1, a\varepsilon) = \delta^*(\delta(q_1, a), \varepsilon) = \delta^*(q_1, \varepsilon)$

5. urratsa: $\delta^*(q_1, \varepsilon) = q_1$

Konputazioa Y multzokoa ez den q_1 egoeran amaitu denez, hitza ez da AFD honi dagokion lengoaiakoa.

Konputazio horri dagokion konfigurazio-sekuentzia honako hau da:

$$\begin{array}{c}(q_0, abca)\\ |\\ (q_0, bca)\\ |\\ (q_1, ca)\\ |\\ (q_1, a)\\ |\\ (q_1, \varepsilon)\end{array}$$

3.2.1.2.3 c-rik ez duten eta a eta b-ren agerpenak orden horretan tartekatuta dituzten hitzez osatutako lengoaiari dagokion AFDa

Adibide honetako AFDak (3.2.4 irudia begiratu) c-rik ez duten eta a eta b orden horretan tartetakuta dituzten $A = \{a, b, c\}$ alfabetoaren gainean definitutako hitzentzat "Bai" erantzungo du. Beraz AFD honi dagokion lengoaiako hitzetan ezin dira bi a jarraian agertu eta ezta bi b ere. Hitz hutsa lengoaiakoa da eta hutsak ez diren hitzen kasuan lehenengo sinboloak a izan behar du. Hala ere, azkeneko sinboloa a edo b izan daiteke. Beraz, ε, $abab$ eta aba hitzentzat AFDak "Bai" erantzungo luke, baina $abca$, $bababa$, aaa, cc eta aab hitzentzat "Ez" erantzungo luke. Lengoaiaren definizio formala honako hau da:

$$\{w \mid w \in A^* \wedge \exists k(k \geq 0 \wedge (w = (ab)^k \vee w = ((ab)^k)a))\}$$

AFDak hiru egoera ditu, q_0, q_1 eta q_2. Hasierako egoera q_0 da eta bai q_0 egoeran bukatzen bada eta bai q_1 egoeran bukatzen bada, AFDaren erantzuna "Bai" izango da.

Jarraian AFDa boskote bezala definituko da.

$$(Q, A, \delta, q_0, Y)$$

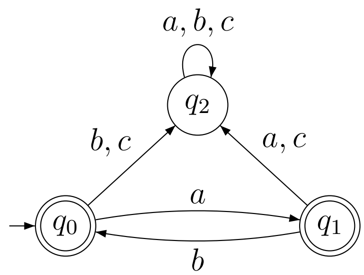
Bost osagaiak honako hauek dira:

- $Q = \{q_0, q_1, q_2\}$
- $A = \{a, b, c\}$
- $\delta : Q \times A \to Q$ jarraian datorren taularen bidez definitutako trantsizio-funtzioa da:

δ	a	b	c
q_0	q_1	q_2	q_2
q_1	q_2	q_0	q_2
q_2	q_2	q_2	q_2

  Taula horren bidez $\delta(q_0, a)$-ren balioa q_2 dela, $\delta(q_0, b)$-ren balioa q_1 dela, eta abar adierazten da.

- q_0 hasierako egoera da. Hasierako egoera beti bakarra da.



3.2.4 irudia. c-rik ez eta a eta b, orden horretan, tartekatuta dituzten hitzen lengoaiari dagokion AFDa.

- $Y = \{q_0, q_1\}$. Adibide honetan hasierako egoera Y multzoan ager daitekeela ikus dezakegu.

Orain $(q_0, abab)$ konfigurazioari dagokion konputazioa garatuko dugu urratsez urrats:

1. urratsa: $\delta^*(q_0, abab) = \delta^*(\delta(q_0, a), bab) = \delta^*(q_1, bab)$

2. urratsa: $\delta^*(q_1, bab) = \delta^*(\delta(q_1, b), ab) = \delta^*(q_0, ab)$

3. urratsa: $\delta^*(q_0, ab) = \delta^*(\delta(q_0, a), b) = \delta^*(q_1, b)$

4. urratsa: $\delta^*(q_1, b) = \delta^*(q_1, b\varepsilon) = \delta^*(\delta(q_1, b), \varepsilon) = \delta^*(q_0, \varepsilon)$

5. urratsa: $\delta^*(q_0, \varepsilon) = q_0$

Konputazioa Y multzokoa den q_0 egoeran amaitu denez, hitza AFD honi dagokion lengoaiakoa da.

Konputazio hori honela adieraz dezakegu grafikoki:

$$\begin{array}{c}(q_0, abab)\\ |\\ (q_1, bab)\\ |\\ (q_0, ab)\\ |\\ (q_1, b)\\ |\\ (q_0, \varepsilon)\end{array}$$

Jarraian (q_0, abc) konfigurazioari dagokion konputazioa garatuko dugu urratsez urrats:

1. urratsa: $\delta^*(q_0, abc) = \delta^*(\delta(q_0, a), bc) = \delta^*(q_1, bc)$

2. urratsa: $\delta^*(q_1, bc) = \delta^*(\delta(q_1, b), c) = \delta^*(q_0, c)$

3. urratsa: $\delta^*(q_0, c) = \delta^*(q_0, c\varepsilon) = \delta^*(\delta(q_0, c), \varepsilon) = \delta^*(q_2, \varepsilon)$

4. urratsa: $\delta^*(q_2, \varepsilon) = q_2$

Konputazioa Y multzokoa ez den q_2 egoeran amaitu denez, hitza ez da AFD honi dagokion lengoaiakoa.

Jarraian konputazio horri dagokion konfigurazio-sekuentzia dator:

$$\begin{array}{c} (q_0, abc) \\ | \\ (q_1 bc) \\ | \\ (q_0, c) \\ | \\ (q_2, \varepsilon) \end{array}$$

3.2.1.2.4 10-rekin bukatzen diren hitzez osatutako lengoaiari dagokion AFDa

Adibide honetako D AFDak (3.2.5 irudia begiratu) 10 katearekin bukatzen diren $A = \{0, 1\}$ alfabetoaren gainean definitutako hitzentzat "Bai" erantzungo du. Beraz, esate baterako 0011010 eta 0010 hitzen kasuan "Bai" erantzungo du, baina 000, 1101, ε eta 0 hitzen kasuan "Ez" erantzungo du. Lengoaiaren definizio formala honako hau da:

$$\{w \mid w \in A^* \wedge \exists u(u \in A^* \wedge w = u10)\}$$

Lengoaia hori honako funtzio honen bidez adierazten badugu:

$$g : A^* \to 2$$
$$g(w) = \begin{cases} 1 & \text{baldin eta } \exists u(u \in A^* \wedge w = u10) \\ 0 & \text{bestela} \end{cases}$$

g funtzioaren definizio horretan, $2 = \{0, 1\}$ da. Bestalde, $A = \{0, 1\}$ eta $2 = \{0, 1\}$ multzoak garbi bereizi behar dira. Izan ere, A multzoko 0 eta 1 balioak hitzak eratzeko erabiliko diren sinboloak dira baina 2 multzoko 0 eta 1 balioek *False* eta *True* balioak ordezkatzen dituzte hurrenez hurren.

D AFDak hiru egoera ditu: q_0, q_1 eta q_2. Hasierako egoera q_0 da eta q_2 egoeran bukatuz gero erantzuna "Bai" izango da.

Jarraian D AFDa boskote gisa definituko da.

$$(Q, A, \delta, q_0, Y)$$

D AFDaren bost osagaiak honako hauek dira:

- $Q = \{q_0, q_1, q_2\}$
- $A = \{0, 1\}$

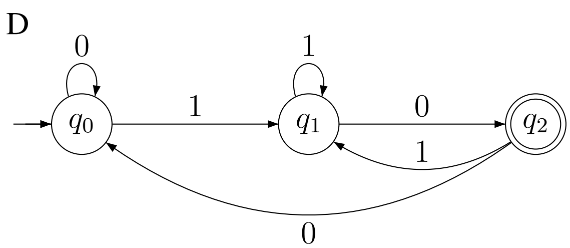
- $\delta : Q \times A \rightarrow Q$ jarraian datorren taularen bidez definitutako trantsizio-funtzioa da:

δ	0	1
q_0	q_0	q_1
q_1	q_2	q_1
q_2	q_0	q_1

  Taula horren bidez $\delta(q_0, 0)$-ren balioa q_0 dela, $\delta(q_0, 1)$-ren balioa q_1 dela, eta abar adierazten da.

- q_0 hasierako egoera da. Hasierako egoera beti bakarra da.

- $Y = \{q_2\}$.



3.2.5 irudia. 10 katearekin bukatzen diren hitzen lengoaiari dagokion AFD bat.

Hona hemen $(q_0, 0010)$ konfigurazioari dagokion urratsez urrats garatutako konputazioa:

1. urratsa: $\delta^*(q_0, 0010) = \delta^*(\delta(q_0, 0), 010) = \delta^*(q_0, 010)$

2. urratsa: $\delta^*(q_0, 010) = \delta^*(\delta(q_0, 0), 10) = \delta^*(q_0, 10)$

3. urratsa: $\delta^*(q_0, 10) = \delta^*(\delta(q_0, 1), 0) = \delta^*(q_1, 0)$

4. urratsa: $\delta^*(q_1, 0) = \delta^*(q_1, 0\varepsilon) = \delta^*(\delta(q_1, 0), \varepsilon) = \delta^*(q_2, \varepsilon)$

5. urratsa: $\delta^*(q_2, \varepsilon) = q_2$

Konputazioa Y multzokoa den q_2 egoeran amaitu denez, hitza AFD honi dagokion lengoaiakoa da.

Hona hemen konputazio hori adar bakarreko zuhaitz gisa ipinita:

$$\begin{array}{c} (q_0, 0010) \\ | \\ (q_0, 010) \\ | \\ (q_0, 10) \\ | \\ (q_1, 0) \\ | \\ (q_2, \varepsilon) \end{array}$$

Beste adibide bat izateko, $(q_0, 111)$ konfigurazioari dagokion urratsez urrats garatutako konputazioa honako hau da:

1. urratsa: $\delta^*(q_0, 111) = \delta^*(\delta(q_0, 1), 11) = \delta^*(q_1, 11)$

2. urratsa: $\delta^*(q_1, 11) = \delta^*(\delta(q_1, 1), 1) = \delta^*(q_1, 1)$

3. urratsa: $\delta^*(q_1, 1) = \delta^*(q_1, 1\varepsilon) = \delta^*(\delta(q_1, 1), \varepsilon) = \delta^*(q_1, \varepsilon)$

4. urratsa: $\delta^*(q_1, \varepsilon) = q_1$

Konputazioa Y multzokoa ez den q_1 egoeran amaitu denez, hitza ez da AFD honi dagokion lengoaiakoa.

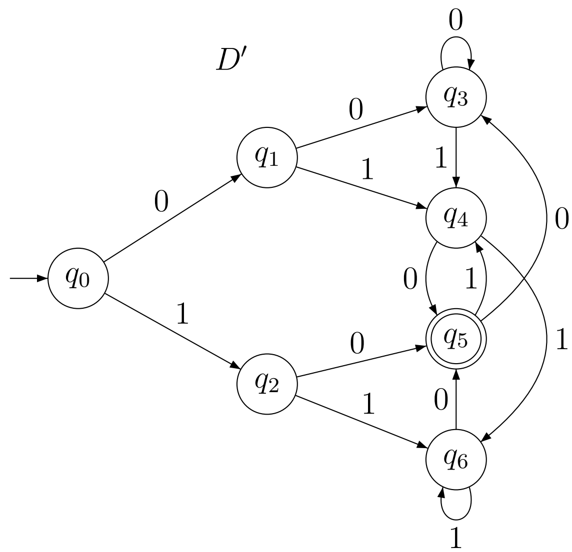
Konputazio hori jarraian erakusten den konfigurazio-sekuentzia gisa adieraz daiteke:

$$\begin{array}{c} (q_0, 111) \\ | \\ (q_1 11) \\ | \\ (q_1, 1) \\ | \\ (q_1, \varepsilon) \end{array}$$

Orokorrean, lengoaia batentzat AFD bat baino gehiago defini daitezke. Adibidez 3.2.6 irudian

$$\{w \mid w \in A^* \wedge \exists u(u \in A^* \wedge w = u10)\}$$

lengoaiako hitzei "Bai" eta lengoaia horretakoak ez diren hitzei "Ez" erantzuten dien beste AFD bat dugu: D' AFDa. AFD honek 3.2.5 irudiko D AFDak baino egoera gehiago ditu baina baliokideak dira, lengoaia bera dagokie biei.



3.2.6 irudia. 10 azpikatearekin amaitzen diren hitzen lengoaiarentzat beste AFD bat.

3.2.2 AFDen diseinua

AFD baten zeregina hitz bat AFD horren bidez definitutako lengoaiakoa al den erabakitzea da. Beste era batera esanda, AFD baten zeregina AFD horri sarrerako datutzat emango zaion hitz bakoitzak propietate jakin bat betetzen al duen erabakitzea da. Beraz, AFDa sinboloz sinbolo hitza irakurriz joango da eta irakurriz doan hitz horrek betetzen duen propietatea zein den "gogoratzeko" baliabide gisa egoerak erabiliko dira. Egoera bakoitzak propietate bat "gogoratzen" edo adierazten du. Une bakoitzean, une horretara arte irakurri den hitz zatiak (edo azpihitzak) zein propietate betetzen duen jakiteko AFDa zein egoeratan dagoen begiratu beharko da. AFDak hitza sinboloz sinbolo irakurriko du eta ezingo du atzeraka egin. Beraz, une bakoitzean AFDa zein egoeratan dagoen jakinda, une horretara arte irakurri den hitz zatiak zein propietate betetzen duen jakingo da eta hitza osatzen duten gainerako sinboloak irakurriz jarraitu beharko da. **Automaten diseinu metodikoan edo sistematikoan**, funtsezko oinarria hitzek bete ditzaketen propietate garrantzitsuak edo esanguratsuak "gogoratzeko" beharrezkoak diren egoerak identifikatzea da. Beharrezkoak diren egoerak zein izango diren erabaki ondoren, egoeren arteko trantsizioak finkatu beharko dira.

3.2.2.1 AFDen diseinua egoerak identifikatuz

Automata finituak sistematikoki diseinatzeko aukera bat automaten diseinu metodikoan edo sistematikoan funtsezkoa den oinarria zuzenean aplikatzen saiatzea da, hau da, urrats bakoitzaren ondoren (sinbolo bakoitza irakurri ondoren) ordura arte irakurritako hitz zatiak bete ditzakeen propietate garrantzitsuak edo esanguratsuak "gogoratzeko" beharrezkoak diren egoerak zuzenean identifikatzea.

Teknika hau hiru adibideren bidez erakutsiko da orain. Adibide denetan alfabetoa $A =$

$\{a, b, c\}$ da.

3.2.2.1.1 a kopuru bikoitia duten hitzen lengoaiari dagokion AFDa

Har dezagun $A = \{a, b, c\}$ alfabetoa. Alfabeto horren gainean definituta dauden eta a kopuru bikoitia duten hitzen lengoaiari dagokion AFD bat diseinatu nahi da. Lengoaia hori honela defini daiteke:

$$\{w|w \in A^* \wedge |w|_a \bmod 2 = 0\}$$

Hitz batek a kopuru bikoitia al duen erabakitzeko, nahikoa da une bakoitzean une horretara arte irakurritako a kopurua bikoitia ala bakoitia izan den gogoratzearekin. Ondorioz, bi egoera beharko dira. Hitza irakurtzen hasi aurretik a kopurua (zero) bikoitia da. Beraz, hasierako egoera a kopurua bikoitia dela "gogoratzeko" erabil daiteke. Aldiz, AFDa beste egoeran baldin badago, une horretara arte irakurritako a kopurua bakoitia dela esan nahiko du:

q_0: orain arte irakurritako a kopurua bikoitia da.

q_1: orain arte irakurritako a kopurua bakoitia da.

Beharrezkoak diren egoerak zein diren erabaki ondoren, egoeren arteko trantsizioak finkatu beharko dira. Horretarako, b edo c agertzeak a-ren agerpen kopurua bikoitia edo bakoitia izatean eraginik ez duela kontura gaitezke. Hori dela-eta, b edo c irakurtzen bada ez da egoeraz aldatu beharko. Aldiz, a sinboloaren beste agerpen batek egoera-aldaketa dakar berarekin. 3.2.7 irudian a kopuru bikoitia duten hitzez osatutako lengoairi dagokion AFD baten trantsizio-diagrama erakusten da.

Jarraian AFD horren osagaiak definituko dira. Adibide honetako AFDa honako boskote honen bidez defini daiteke:

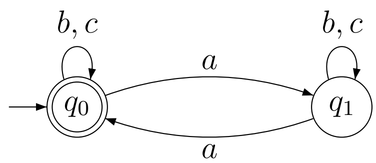
$$(Q, A, \delta, q_0, Y)$$

Boskote horretan:

- $Q = \{q_0, q_1\}$
- $A = \{a, b, c\}$
- $\delta : Q \times A \to Q$ honako taula honen bidez definitutako trantsizio-funtzioa da:

δ	a	b	c
q_0	q_1	q_0	q_0
q_1	q_0	q_1	q_1

- q_0 hasierako egoera da.
- $Y = \{q_0\}$.



3.2.7 irudia. a kopuru bikoitia duten hitzen lengoaiari dagokion AFD bat.

3.2.2.1.2 a kopuru bikoitia eta b kopuru bikoitia duten hitzen lengoaiari dagokion AFD bat

Har dezagun $A = \{a, b, c\}$ alfabetoa. a kopuru bikoitia eta, gainera, b kopurua ere bikoitia duten hitzez osatutako lengoaiarentzat AFD bat diseinatu nahi da. Lengoaia hori formalki honela defini daiteke:

$$\{w | w \in A^* \wedge (|w|_a \bmod 2 = 0) \wedge (|w|_b \bmod 2 = 0)\}$$

Hitz batek a kopuru bikoitia eta b kopuru bikoitia al dituen erabakitzeko, une bakoitzean une horretara arte irakurritako a kopurua bikoitia ala bakoitia al den eta une horretara arte irakurritako b kopurua bikoitia ala bakoitia al den gogoratzea da aukera bat. Ondorioz, lau egoera beharko dira lau konbinazio posible baitaude. Hitza irakurtzen hasi aurretik a kopurua (zero) bikoitia izango da eta b kopurua ere (zero) bikoitia izango da. Beraz, hasierako egoera a kopurua eta b kopurua, biak, bikoitiak direla "gogoratzeko" erabil daiteke. Beste hiru egoerek gelditzen diren beste hiru aukera edo konbinazioak gogoratzeko balioko dute:

- q_0: orain arte irakurritako a kopurua bikoitia da eta b kopurua ere bikoitia da.
- q_1: orain arte irakurritako a kopurua bakoitia da eta, aldiz, b kopurua bikoitia da.
- q_2: orain arte irakurritako a kopurua bikoitia da eta, aldiz, b kopurua bakoitia da.
- q_3: orain arte irakurritako a kopurua bakoitia da eta b kopurua ere bakoitia da.

Egoerak zein izango diren erabaki ondoren, trantsizioak finkatu behar dira. Hasteko, c agertzeak a kopurua bikoitia edo bakoitia izatean eraginik ez duela eta, era berean, b kopurua bikoitia edo bakoitia izatean eraginik ez duela kontura gaitezke. Beraz, c irakurritakoan ez da egoeraz aldatu behar. Bestalde, a edo b agertzeak a kopurua edo b kopurua bikotitik bakoitira edo bakoititik bikoitira igarotzea dakar. Hori dela-eta, a edo b agertzeak egoera-aldaketa dakar berarekin. 3.2.8 irudian a kopuru bikoitia eta b kopuru bikoitia duten hitzez osatutako lengoaiarentzat diseinatutako AFDari dagokion trantsizio-diagrama erakusten da. Diseinatutako AFDa formalki honako boskote honen bidez defini daiteke:

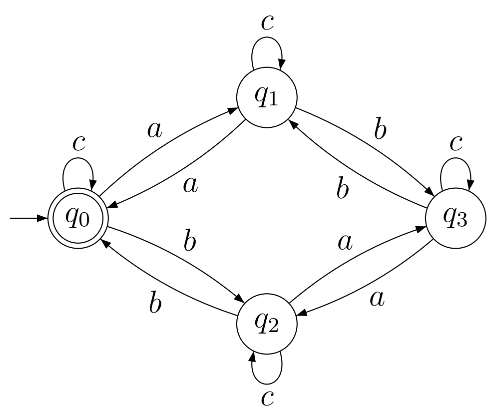
$$(Q, A, \delta, q_0, Y)$$

Boskote horren osagaien definizioa honako hau da:

- $Q = \{q_0, q_1, q_2, q_3\}$
- $A = \{a, b, c\}$
- $\delta : Q \times A \to Q$ honako taula honen bidez definitutako trantsizio-funtzioa da:

δ	a	b	c
q_0	q_1	q_2	q_0
q_1	q_0	q_3	q_1
q_2	q_3	q_0	q_2
q_3	q_2	q_1	q_3

- q_0 hasierako egoera da.
- $Y = \{q_0\}$.



3.2.8 irudia. a kopuru bikoitia eta b kopuru bikoitia duten hitzen legoaiari dagokion AFD bat.

3.2.2.1.3 aa katea bai baina c-rik ez duten hitzen lengoaiarentzat AFD bat

Har dezagun $A = \{a, b, c\}$ alfabetoa. aa katea bai baina c-rik ez duten hitzen lengoaiarentzat AFD bat diseinatu nahi da. Lengoaia hori formalki honela defini daiteke:

$$\begin{aligned}\{w | w \in A^* \wedge \quad & \exists u, v(u \in A^* \wedge v \in A^* \wedge w = uaav) \\ & \wedge \neg \exists u, v(u \in A^* \wedge v \in A^* \wedge w = ucv)\}\end{aligned}$$

Hitz batek aa katea gutxienez behin izatea eta c-ren agerpenik ez izatea betetzen al duen erabakitzeko, alde batetik c-rik agertu al den gogoratu beharko da —kasu horretan behin betiko erantzuna ezezkoa izango da— eta bestetik, c ez bada oraindik agertu, aa katea irakurri al den

—baldintza hori betez— edo aa oraindik ez dela irakurri baina irakurritako azkeneko sinboloa a izan al den edo aa oraindik ez dela irakurri eta irakurritako azkeneko sinboloa ere a ez dela izan gogoratu beharko da. Beraz, lau egoera beharko dira:

q_0: orain arte c ez da agertu, aa ere ez da agertu eta irakurritako azkeneko sinboloa ez da a izan.

q_1: orain arte c ez da agertu, aa ere ez da agertu baina irakurritako azkeneko sinboloa a izan da. Oraintxe irakurri den a hori aa bateko lehenengo a izan daiteke.

q_2: orain arte c ez da agertu baina aa agertu da.

q_3: c agertu da. Berdin da aa agertu al den ala ez.

Egoerak zein izango diren erabaki ondoren, trantsizioak zehaztu beharko dira. AFDa q_0, q_1 eta q_2 egoeretan dagoenean c agertzen bada, q_3 egoerara igaro beharko da. AFDa q_0 eta q_1 egoeretan dagoela b agertzen bada, ez dago aa baten atarian eta, ondorioz, hurrengo egoera q_0 izango da. AFDa q_0 egoeran dagoela a irakurtzen bada, aa baten hasieran egon daiteke eta hori gogoratzeko, hau da, a bat duela gogoratzeko q_1 egoerara igaroko litzateke. AFDa q_1 egoeran dagoenean horrek a bat irakurri dela esan nahiko du, eta jarraian beste a bat irakurtzen bada, q_2 egoerara igaroko da aa katea osatu delako. AFDa q_2 egoeran baldin badago, bai a eta bai b irakurrita ere, egoera berean mantenduko da, aa agertu delako lehenago eta oraindik c-rik ez delako agertu. Aipatutako irizpideak kontuan hartuz eraiki den AFDaren trantsizio-diagrama 3.2.9 irudian erakusten da.

Diseinatutako AFDa formalki honako boskote honen bidez adieraz daiteke:

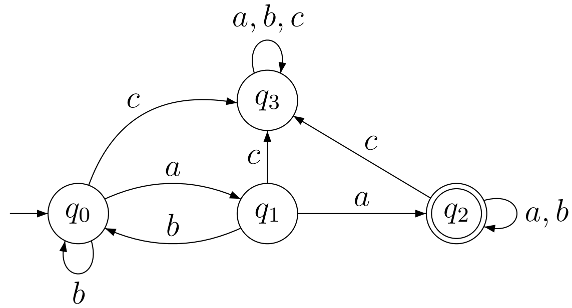
$$(Q, A, \delta, q_0, Y)$$

Boskote horretan:

- $Q = \{q_0, q_1, q_2, q_3\}$
- $A = \{a, b, c\}$
- $\delta : Q \times A \to Q$ honako taula honen bidez definitutako trantsizio-funtzioa da:

δ	a	b	c
q_0	q_1	q_0	q_3
q_1	q_2	q_0	q_3
q_2	q_2	q_2	q_3
q_3	q_3	q_3	q_3

- q_0 hasierako egoera da.
- $Y = \{q_2\}$.



3.2.9 irudia. aa katea duten baina c-rik ez duten hitzen lengoaiarentzat diseinatutako AFDa.

3.2.2.1.4 $a^j b^k$ egitura duten hitzen lengoaiarentzat diseinatutako AFD bat

Har dezagun $A = \{a, b, c\}$ alfabetoa. $j \geq 0$ eta $k \geq 0$ izanda, $a^j b^k$ egitura duten hitzez osatutako lengoaiarentzat AFD bat diseinatu nahi da. Lengoaia hori formalki honela defini daiteke:

$$\{w \mid w \in A^* \wedge \exists j, k(j \in I\!N \wedge k \in I\!N \wedge w = a^j b^k)\}$$

Hitz batek, $j \geq 0$ eta $k \geq 0$ izanda, $a^j b^k$ egitura al duen erabakitzeko une bakoitzean gogoratu beharreko propietate esanguratsutzat honako hauek har ditzakegu:

- Lehenengo, a sinboloaren errepikapenez eratutako blokean al gauden (bloke hori hutsa izan daiteke).
- Bigarren lekuan, a-ren errepikapenez eratutako blokea bukatu eta b sinboloaren errepikapenez eratutako blokean al gauden (bloke hori ere hutsa izan daiteke).
- Hirugarren lekuan, c-rik agertu al den.
- Laugarren kasuan, oraindik c-rik agertu ez baina b sinboloaren errepikapenez eratutako blokea bukatu ondoren a-ren bat agertu al den, $a^j b^k$ egitura apurtuz.

Planteamendu horri jarraituz, lau egoera beharko dira:

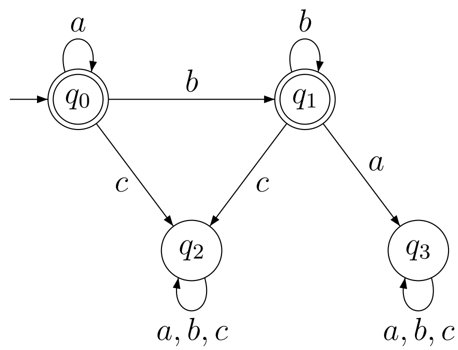
q_0: orain arte c-rik ez da agertu eta a sinboloaren agerpenez eratutako blokean gaude.

q_1: orain arte c-rik ez da agertu eta a sinboloaren agerpenez eratutako blokea bukatu da eta b sinboloaren agerpenez eratutako blokean gaude.

q_2: gutxienez c bat agertu da.

q_3: orain arte c-rik ez da agertu baina b sinboloaren agerpenez eratutako bloke ez-hutsaren ondoren a-ren bat agertu da.

Zein egoera beharko diren erabaki ondoren, trantsizioak zehaztu beharko dira. AFDa edozein egoeratan egonda ere, c sinboloa agertzen bada q_2 egoerara igaro beharko du AFDak. AFDa q_0 egoeran baldin badago eta a irakurtzen bada, q_0 egoeran mantendu beharko du baina b irakurtzen bada, q_1 egoerara igaro beharko du. Oraingo egoera q_1 baldin bada eta b agertzen bada, AFDak q_1 egoeran jarraitu beharko du baina a agertzen bada, AFDak q_3 egoerara igaro beharko du. AFDa q_2 egoeran baldin badago, edozein sinbolo irakurrita ere, egoera berean manteduko da. Azkenik, AFDa q_3 egoeran baldin badago eta a edo b agertzen bada, AFDak q_3 egoeran jarraituko du, baina c agertzen bada, q_2 egoerara igaroko da. Oraintxe aipatutako irizpideak kontuan hartuz eraiki den AFDaren trantsizio-diagrama 3.2.10 irudian dago.



3.2.10 irudia. $j \geq 0$ eta $k \geq 0$ izanda, $a^j b^k$ egitura duten hitzen lengoaiari dagokion AFD bat.

Diseinatutako AFDa formalki honako boskote honen bidez defini daiteke:

$$(Q, A, \delta, q_0, Y)$$

Boskote horretan:

- $Q = \{q_0, q_1, q_2, q_3\}$
- $A = \{a, b, c\}$
- $\delta : Q \times A \to Q$ honako taula honen bidez definitutako trantsizio-funtzioa da:

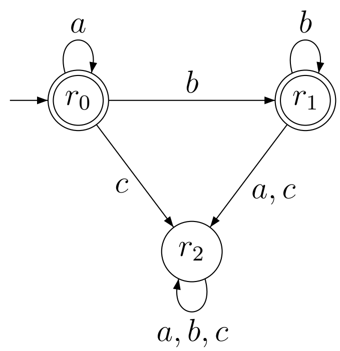
δ	a	b	c
q_0	q_0	q_1	q_2
q_1	q_3	q_1	q_2
q_2	q_2	q_2	q_2
q_3	q_3	q_3	q_3

- q_0 hasierako egoera da.
- $Y = \{q_0, q_1\}$.

Eraiki den AFDa hobeto aztertuz, q_2 eta q_3 egoerek propietate bera adierazten dutela kontsidera dezakegu: irakurritako hitzak ez du $a^j b^k$ egitura, hor $j \geq 0$ eta $k \geq 0$ izanda. Beraz, bi egoera desberdin eduki beharrean egoera bakar bat nahikoa izango litzateke:

- r_0: orain arte c-rik ez da agertu eta a sinboloaren agerpenez eratutako blokean gaude.
- r_1: orain arte c-rik ez da agertu eta a sinboloaren agerpenez eratutako blokea bukatu da eta b sinboloaren agerpenez eratutako blokean gaude.
- r_2: gutxienez c bat agertu da edo b sinboloaren errepikapenez eratutako bloke ez-hutsaren ondoren a-ren bat agertu da.

3.2.11 irudian bigarren planteamendu horri jarraituz eraikitako AFDari dagokion trantsizio-diagrama ikus daiteke.



3.2.11 irudia. $j \geq 0$ eta $k \geq 0$ izanda, $a^j b^k$ egitura duten hitzen lengoaiari dagokion beste AFD bat.

Diseinatutako AFDa formalki honako boskote honen bidez defini daiteke:

$$(R, A, \delta, q_0, Y)$$

Boskote horretako osagaiak honako hauek dira:

- $R = \{r_0, r_1, r_2\}$
- $A = \{a, b, c\}$
- $\delta : R \times A \to R$ honako taula honen bidez definitutako trantsizio-funtzioa da:

δ	a	b	c
r_0	r_0	r_1	r_2
r_1	r_2	r_1	r_2
r_2	r_2	r_2	r_2

- r_0 hasierako egoera da.

- $Y = \{r_0, r_1\}$.

Adibide honek erakusten duen bezala, AFD bat eraikitzean gerta daiteke berez batera ipini daitezkeen propietateak bakoitza bere aldetik, hau da, bereizita kontsideratzea. Hala ere, diseinatutako AFDa diseina daitekeen AFD txikiena ez izateak ez gaitu kezkatu behar. Hasieran ez da beharrezkoa egoera kopurua ahal den txikiena izateaz gehiegi arduratzea.

3.2.2.2 AFDen diseinua egoera-multzoak identifikatuz eta gero multzo bakoitzari dagozkion egoerak zehaztuz

Lengoaia batekoak diren hitzek betetzen duten baldintza oso erraza ez denean eta, ondorioz, gogoratu beharreko propietate esanguratsuak zein diren identifikatzea eta beharko diren egoerak zein diren identifikatzea erraza ez denean, aukera bat abstrakzio-maila handitzea eta orokorragoak diren propietateak identifikatzea da. Gero, bigarren urrats batean edo behar bada, urrats kopuru handiago batean propietate orokor horiek xehatuz joan beharko da. Kasu gehienetan, hasierako propietate orokor bakoitzari egoera multzo bat egokituko zaio xehatze-urratsen ondoren.

Jarraian AFDak diseinatzeko teknika hau erakusten duten bi adibide emango dira.

3.2.2.2.1 aaa katea gutxienez behin eta b kopuru bikoitia duten hitzen lengoaiarentzat AFD baten diseinua

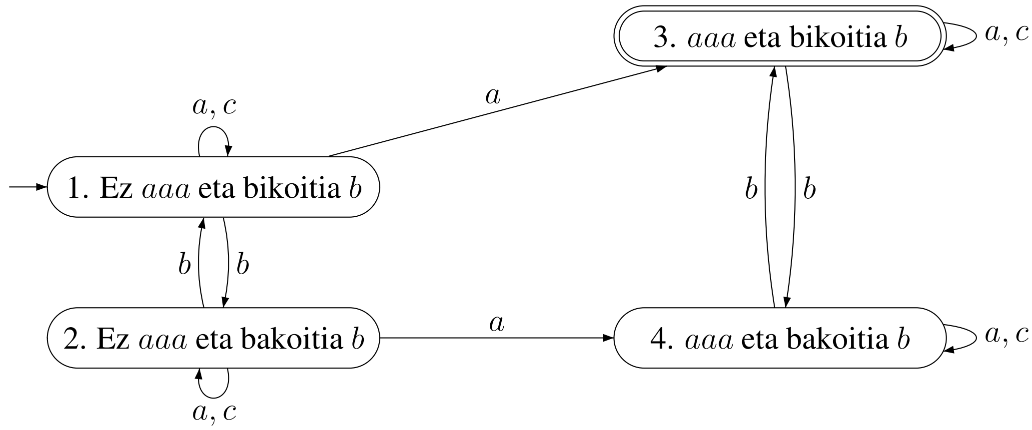
Har ditzagun $A = \{a, b, c\}$ alfabetoa eta aaa katea gutxienez behin eta b kopuru bikoitia duten hitzez eratutako lengoaia. Lengoaia horrentzat AFD bat diseinatuko dugu orain. Lengoaia hori formalki honela defini daiteke:

$$\{w | w \in A^* \wedge \exists u, v (u \in A^* \wedge v \in A^* \wedge w = uaaav) \wedge (|w|_b \bmod 2 = 0)\}$$

Lehenengo urrats bezala lau kasu bereiz ditzakegu:

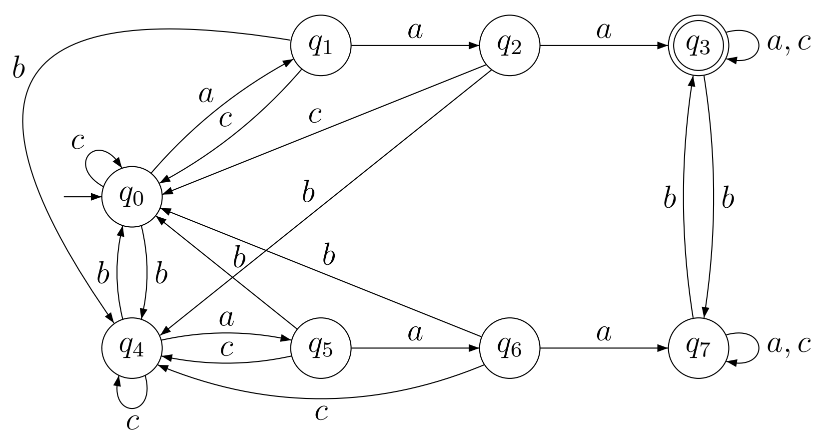
1. Oraindik aaa ez da agertu eta b kopurua bikoitia da.
2. Oraindik aaa ez da agertu eta b kopurua bakoitia da.
3. aaa agertu da eta b kopurua bikoitia da.
4. aaa agertu da eta b kopurua bakoitia da.

3.2.12 irudian lehenengo urrats horren ondorio bezala identifikatutako blokeak eta bloke (edo egoera-multzo) horien arteko trantsizioak erakusten dituen eskema dugu. Eskema horretan anbiguotasun batzuk ere badaude. Hain zuzen ere, lehenengo blokean gaudenean a irakurriz bloke horretan bertan gera gaitezke edo hirugarren blokera joan gaitezke. Izan ere, lehenengo blokean gaudela a agertzen bada, a hori a sinboloaren errepikapenez eratutako kate bateko lehenengo osagaia edo bigarrena edo hirugarrena izan daiteke. Bigarren urrats bezala, bloke horietako bakoitza xehatu egin beharko da. Bai lehenengo blokean eta bai bigarren blokean, jarraian dauden zero a, a bat eta jarraian dauden bi a daudela gogoratzeko hiru egoera beharko



3.2.12 irudia. aaa katea gutxienez behin eta b kopuru bikoitia duten hitzen lengoaiarentzat AFD bat eraikitzeko eskema.

dira. Hirugarren eta laugarren blokeetan egoera bat nahikoa da. Beharrezkoak diren egoera denak identifikatu ondoren, egoera horien arteko trantsizioak zehaztu beharko dira. 3.2.13 irudian adibide honetan eraiki den AFDaren trantsizio-diagrama dugu. Lehenengo blokea xehatuz q_0, q_1 eta q_2 egoerak sortu dira; bigarren blokea xehatuz q_4, q_5 eta q_6 egoerak sortu dira; hirugarren eta laugarren blokeetatik q_3 eta q_7 egoerak sortu dira hurrenez hurren.



3.2.13 irudia. aaa katea gutxienez behin eta b kopuru bikoitia duten hitzen lengoaiarentzat AFD bat.

Diseinatutako AFDa formalki honako boskote honen bidez defini daiteke:

$$(Q, A, \delta, q_0, Y)$$

Boskote horretan:

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7\}$
- $A = \{a, b, c\}$
- $\delta : Q \times A \to Q$ honako taula honen bidez definitutako trantsizio-funtzioa da:

δ	a	b	c
q_0	q_1	q_4	q_0
q_1	q_2	q_4	q_0
q_2	q_3	q_4	q_0
q_3	q_3	q_7	q_3
q_4	q_5	q_0	q_4
q_5	q_6	q_0	q_4
q_6	q_7	q_0	q_4
q_7	q_7	q_3	q_7

- q_0 hasierako egoera da.
- $Y = \{q_3\}$.

3.2.2.2.2 aa baduten baina bb ez duten hitzen lengoaiarentzat AFD baten diseinua

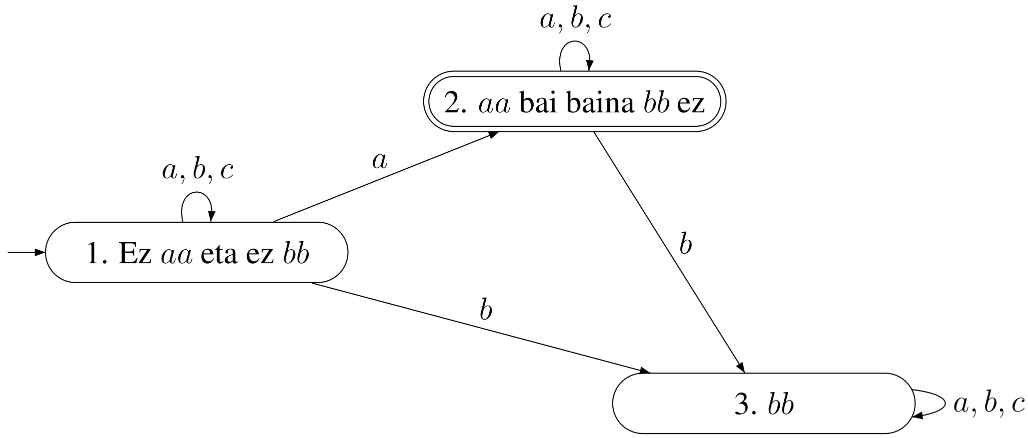
Har dezagun $A = \{a, b, c\}$ alfabetoa. Gutxienez aa katearen agerpen bat eta bb katearen agerpenik ez duten hitzez eratutako lengoaiarentzat AFD bat diseinatu nahi da. Lengoaia hori formalki honela defini daiteke:

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

Lehenengo urratsean honako hiru kasu hauek kontsidera daitezke:

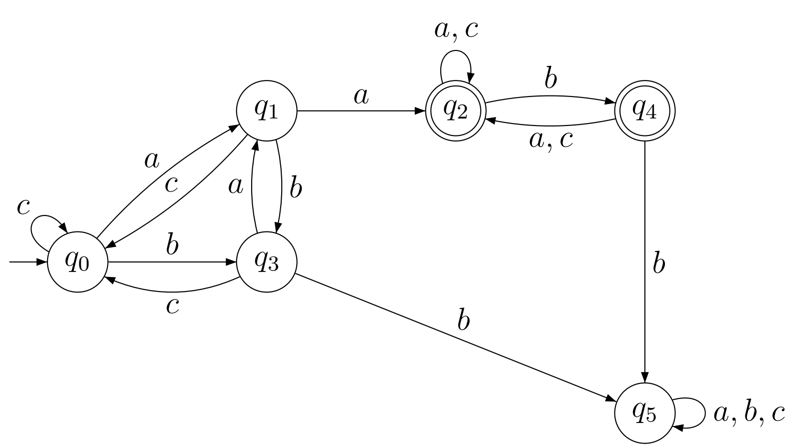
1. Oraindik ez da agertu ez aa eta ez bb.
2. aa agertu da baina bb ez.
3. bb agertu da eta berdin da aa agertu al den ala ez.

3.2.14 irudian lehenengo urrats horretan identifikatutako bloke edo egoera multzoen arteko trantsizioak erakusten dituen eskema dugu. Eskema horretan anbiguotasun batzuk ere badaude. Hain zuzen ere, lehenengo blokean gaudenean b irakurriz bloke horretan bertan gera gaitezke edo hirugarren blokera joan gaitezke. Izan ere, lehenengo blokean gaudela b agertzen bada, b hori b sinboloaren errepikapenez eratutako kate bateko lehenengo osagaia edo bigarrena izan daiteke. Eskema eraiki ondoren, bigarren urrats bezala bloke bakoitza xehatu egin beharko da. Lehenengo blokean hiru egoera beharko dira une honetan jarraian dauden zero a eta zero b ditugula edo a bat dugula edo b bat dugula gogoratzeko edo adierazteko. Bigarren blokean bi egoera beharko dira une honetan jarraian dauden zero b ditugula edo b bat dugula gogoratzeko. Hirugarren blokean egoera bakarra beharko da bb dugula gogoratzeko. Beharrezkoak diren



3.2.14 irudia. aa bai baina bb-rik ez duten hitzez eratutako lengoaiari dagokion AFD baten eskema.

egoerak identifikatu ondoren, trantsizioak zehaztu beharko dira. Lortutako AFDari dagokion trantsizio-diagrama 3.2.15 irudian ikus daiteke. Eskemako lehenengo blokea xehatuz q_0, q_1 eta q_3 egoerak sortu dira; bigarren blokea xehatuz q_2 eta q_4 egoerak sortu dira; hirugarren blokea q_5 egoeraren bidez gauzatu da.



3.2.15 irudia. aa bai baina bb-rik ez duten hitzez eratutako lengoaiari dagokion AFD bat.

Diseinatutako AFDa formalki honako boskote honen bidez defini daiteke:

$$(Q, A, \delta, q_0, Y)$$

Boskote horretako osagaiak honako hauek dira:

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$
- $A = \{a, b, c\}$
- $\delta : Q \times A \rightarrow Q$ honako taula honen bidez definitutako trantsizio-funtzioa da:

δ	a	b	c
q_0	q_1	q_3	q_0
q_1	q_2	q_3	q_0
q_2	q_2	q_4	q_2
q_3	q_1	q_5	q_0
q_4	q_2	q_5	q_2
q_5	q_5	q_5	q_5

- q_0 hasierako egoera da.
- $Y = \{q_2, q_4\}$.

3.2.2.3 AFDen diseinua propietateen ukapenentzat

Kasu askotan, lengoaia bateko hitzek propietate baten ukapena bete behar izateak lengoaia horrentzat AFD bat diseinatzea zailagoa egiten du. Kasu horietan, estrategia posible bat lengoaia osagarriari dagokion AFD bat eraikitzea da, hau da, propietatearen ukapena bete beharrean propietatea bera betetzen duten hitzen lengoaiari dagokion AFDa eraikitzea. AFD hori eraiki ondoren, zirkulu bikoitzik ez duten egoerei zirkulu bikoitza ipini eta zirkulu bikoitza duten egoerei zirkulu bikoitza kenduz lortzen den AFDa propietatearen ukapena betetzen duten hitzen lengoaiari dagokion AFD bat izango da.

Orain teknika hau bi adibide garatuz azalduko da.

3.2.2.3.1 aaa ez duten hitzen lengoaiarentzat AFD baten diseinua

Har dezagun $A = \{a, b, c\}$ alfabetoa. aaa katea **ez** duten hitzez osatutako L lengoaiarentzat AFD bat diseinatu nahi da. L lengoaia hori formalki honela defini daiteke:

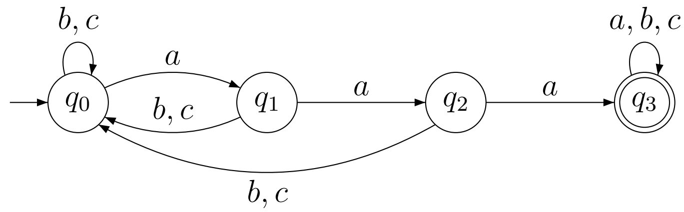
$$L = \{w \mid w \in A^* \wedge \neg\exists u, v(u \in A^* \wedge v \in A^* \wedge w = uaaav)\}$$

L lengoaia horretako hitzek propietate baten ukapena bete behar dute. Propietatea aaa katea edukitzea da. $\overline{L}$ lengoaia osagarria aaa katea duten hitzez eratutakoa izango da:

$$\overline{L} = \{w \mid w \in A^* \wedge \exists u, v(u \in A^* \wedge v \in A^* \wedge w = uaaav)\}$$

$\overline{L}$ lengoaian ez dago ukapenik. Hasteko, $\overline{L}$ lengoaiarentzat AFD bat diseinatuko genuke. 3.2.16 irudian $\overline{L}$ lengoaiarentzat diseinatutako AFD baten trantsizio-diagrama dugu.

AFD hori formalki honako boskote honen bidez adieraz daiteke:



3.2.16 irudia. aaa duten hitzen lengoaiarentzat AFD bat.

$$(Q, A, \delta, q_0, Y)$$

Boskote horretako osagai bakoitza honela definituta dago:

- $Q = \{q_0, q_1, q_2, q_3\}$
- $A = \{a, b, c\}$
- $\delta : Q \times A \to Q$ honako taula honen bidez definitutako trantsizio-funtzioa da:

δ	a	b	c
q_0	q_1	q_0	q_0
q_1	q_2	q_0	q_0
q_2	q_3	q_0	q_0
q_3	q_3	q_3	q_3

- q_0 hasierako egoera da.
- $Y = \{q_3\}$.

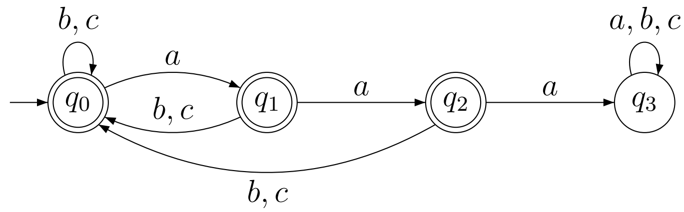
L lengoaiarentzat AFD bat diseinatzeko, nahikoa da $\overline{L}$ lengoaiarentzat diseinatu den AFDa hartu eta zirkulu bikoitza ez duten egoerei zirkulu bikoitza ipintzearekin eta zirkulu bikoitza duten egoerei zirkulu bikoitza kentzearekin. 3.2.17 irudian metodo hau jarraituz L lengoaiarentzat lortu den AFDari dagokion tratsizio-diagrama dugu.

L lengoaiarentzat eraiki den AFD hori honako boskote hau bezala adieraz daiteke:

$$(Q, A, \delta, q_0, Y)$$

Boskote horretako osagaiak honako hauek dira:

- $Q = \{q_0, q_1, q_2, q_3\}$
- $A = \{a, b, c\}$



3.2.17 irudia. aaa ez duten hitzen lengoaiarentzat AFD bat.

- $\delta : Q \times A \rightarrow Q$ honako taula honen bidez definitutako trantsizio-funtzioa da:

δ	a	b	c
q_0	q_1	q_0	q_0
q_1	q_2	q_0	q_0
q_2	q_3	q_0	q_0
q_3	q_3	q_3	q_3

- q_0 hasierako egoera da.
- $Y = \{q_0, q_1, q_2\}$.

3.2.2.3.2 $a^j b^k$ egitura ez duten hitzen lengoaiarentzat AFD baten diseinua

Har dezagun $A = \{a, b, c\}$ alfabetoa. $j \geq 0$ eta $k \geq 0$ izanda, $a^j b^k$ egitura **ez** duten hitzez eratutako L lengoaiarentzat AFD bat diseinatu nahi da. L lengoaia hori formalki honela defini daiteke:

$$L = \{w \mid w \in A^* \land \neg\exists j, k(j \in I\!N \land k \in I\!N \land w = a^j b^k)\}$$

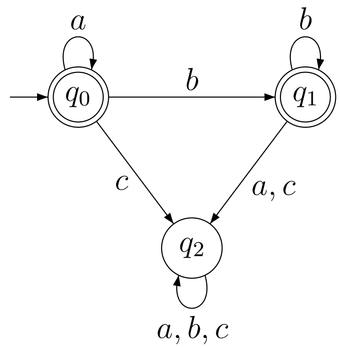
L lengoaia horretako hitzek propietate baten ukapena bete behar dute. Propietatea $a^j b^k$ egitura ez edukitzea da, hor $j \geq 0$ eta $k \geq 0$ izanda. $\overline{L}$ lengoaia osagarria $a^j b^k$ egitura duten hitzez eratutako lengoaia da, hor $j \geq 0$ eta $k \geq 0$ izanda:

$$\overline{L} = \{w \mid w \in A^* \land \exists j, k(j \in I\!N \land k \in I\!N \land w = a^j b^k)\}$$

$\overline{L}$ lengoaian ez dago ukapenik. L lengoaiarentzat AFD bat eraikitzeko, hasteko $\overline{L}$ lengoaiarentzat AFD bat eraikiko dugu. 3.2.18 irudian $\overline{L}$ lengoaiarentzat eraikitako AFD baten trantsizio-diagrama erakusten da (3.2.11 irudiko trantsizio-diagrama bera).

$\overline{L}$ lengoaiarentzat eraikitako AFDa boskoten baten bidez formalizatu daiteke:

$$(Q, A, \delta, q_0, Y)$$



3.2.18 irudia. $j \geq 0$ eta $k \geq 0$ izanda, $a^j b^k$ egitura duten hitzen lengoaiarentzat AFD bat.

Boskote horretako osagai bakoitza honela definituta dago:

- $Q = \{q_0, q_1, q_2\}$
- $A = \{a, b, c\}$
- $\delta : Q \times A \rightarrow Q$ honako taula honen bidez definitutako trantsizio-funtzioa da:

δ	a	b	c
q_0	q_0	q_1	q_2
q_1	q_2	q_1	q_2
q_2	q_2	q_2	q_2

- q_0 hasierako egoera da.
- $Y = \{q_0, q_1\}$.

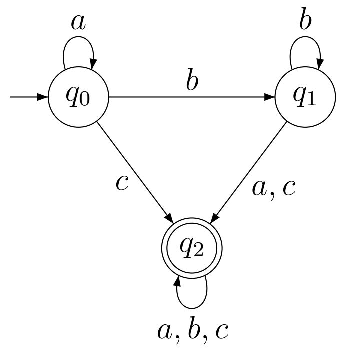
L lengoaiarentzat AFD bat eraikitzeko, nahikoa da $\overline{L}$ lengoaiarentzat eraiki den AFDa hartu eta zirkulu bikoitza duten egoerei zirkulu bikoitza kentzearekin eta zirkulu bikoitza ez duten egoerei zirkulu bikoitza ipintzearekin. 3.2.19 irdian, metodo hori erabiliz L lengoaiarentzat lortu den AFDaren trantsizio-diagrama ikus daiteke.

L lengoaiarentzat diseinatutako AFDa honako boskote honen bidez formaliza daiteke:

$$(Q, A, \delta, q_0, Y)$$

Boskote horretako elementu bakoitza honela definituta dago:

- $Q = \{q_0, q_1, q_2\}$
- $A = \{a, b, c\}$
- $\delta : Q \times A \rightarrow Q$ honako taula honen bidez definitutako trantsizio-funtzioa da:



3.2.19 irudia. $j \geq 0$ eta $k \geq 0$ izanda, $a^j b^k$ egitura ez duten hitzen lengoaiarentzat AFD bat.

δ	a	b	c
q_0	q_0	q_1	q_2
q_1	q_2	q_1	q_2
q_2	q_2	q_2	q_2

- q_0 hasierako egoera da.
- $Y = \{q_2\}$.

3.2.2.4 AFDen diseinua propietateen konjuntzioarentzat (Lengoaien arteko ebakidura)

L lengoaia bateko hitzek bete beharreko propietatea $\psi_1 \wedge \psi_2$ erako konjuntzio baten bidez adieraz daitekeenean, ψ_1 eta ψ_2 bi propietate izanda, L lengoaia hori ψ_1 propietatea betetzen duten hitzez osatutako L_1 lengoaiaren eta ψ_2 propietatea betetzen duten hitzez osatutako L_2 lengoaiaren arteko ebakidura gisa defini daiteke. Beraz, $L = L_1 \cap L_2$ izango da. Gainera, L-ri dagokion AFD bat era sistematikoan edo metodikoan diseina daiteke L_1 lengoaiari dagokion AFD bat eta L_2 lengoaiari dagokion beste AFD bat erabiliz. L_1 lengoaiari dagokion $D_1 = (Q, A, \delta_1, q_0, Y_1)$ AFDa eta L_2 lengoaiari dagokion $D_2 = (R, A, \delta_2, r_0, Y_2)$ AFDa hartuz, $D = (Q \times R, A, \delta_1 \times \delta_2, (q_0, r_0), Y_1 \times Y_2)$ AFDa $L_1 \cap L_2$ lengoaiari dagokion AFD bat izango da. D AFD horretan, $Q \times R$ egoera-multzoa Q eta R multzoen arteko biderketa kartesiarra da; beraz, lehenengo osagaitzat Q-ko elementu bat eta bigarren osagaitzat R-ko elementu bat ipiniz osa daitezkeen bikote guztiez eratutako multzoa. Bestalde, $Y_1 \times Y_2$ multzoa Y_1 eta Y_2 multzoen arteko biderketa kartesiarra da, hau da, lehenengo osagaitzat Y_1-eko elementu bat eta bigarren osagaitzat Y_2-ko elementu bat ipiniz osa daitezkeen bikote guztiez eratutako multzoa. Azkenik, $\delta_1 \times \delta_2$ funtzioa $(Q \times R) \times A \to Q \times R$ motakoa izango da eta $(q_i, r_j) \in Q \times R$ erako edozein bikote eta $\alpha \in A$ edozein sinbolo hartuz, $\delta_1 \times \delta_2((q_i, r_j), \alpha) = (\delta_1(q_i, \alpha), \delta_2(r_j, \alpha))$ beteko da.

Hiru adibide garatuko dira teknika hau erakusteko. Lehenengo adibideko lengoaiako hitzek bi propietate bete behar dituzte eta, ondorioz, lengoaia hori bi lengoairen arteko ebakidura bezala plantea daiteke. Bigarrengo adibidean ere hitzek bi baldintza bete behar dituzte baina adibide horretako azpimarragarriena honako hau da: konjuntzioaren teknika aplikatuz lortutako AFDan egoera batzuk hasierako egoeratik iristezinak dira eta, horregatik, ezaba daitezke. Hirugarren adibideko lengoaiako hitzek hiru propietate bete behar dituzte eta lengoaia horri dagokion AFDa diseinatzeko konjuntzioaren teknika bi aldiz aplikatuko da.

3.2.2.4.1 aaa katea duten eta b sinboloa kopuru bikoitian duten hitzez eratutako lengoaiarentzat AFD bat

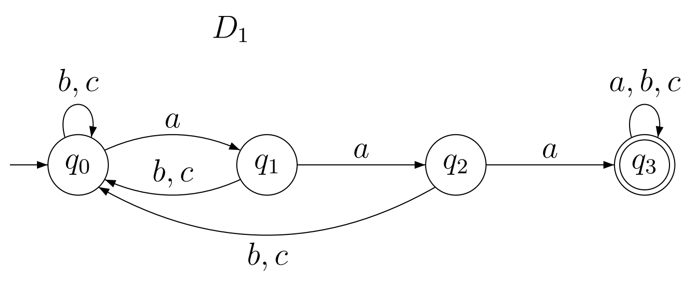
Har dezagun $A = \{a, b, c\}$ alfabetoa. aaa katea gutxienez behin eta, gainera, b sinboloa kopuru bikoitian duten hitzez eratutako L lengoaiarentzat AFD bat diseinatu nahi da. L lengoaia hori formalki honela defini daiteke:

$$L = \{w \mid w \in A^* \land \exists u, v(u \in A^* \land v \in A^* \land w = uaaav) \land ((|w|_b \bmod 2) = 0)\}$$

L lengoaiako hitzek bi propietateren konjuntzioa den baldintza bat bete behar dute: gutxienez aaa bat edukitzea eta b kopuru bikoitia edukitzea. L lengoaiarentzat AFD bat eraikitzeko, aukera bat propietate horietako bakoitza kontuan hartuz bi AFD eraikitzea da. Beste era batera esanda, honako L_1 eta L_2 lengoaiak kontsidera ditzakegu:

$$L_1 = \{w \mid w \in A^* \land \exists u, v(u \in A^* \land v \in A^* \land w = uaaav)\}$$
$$L_2 = \{w \mid w \in A^* \land ((|w|_b \bmod 2) = 0)\}$$

L lengoaia $L_1 \cap L_2$ da. L_1 lengoaiako hitzek betetzen duten propietatea aaa edukitzea da eta L_2 lengoaiako hitzek betetzen duten propietatea b kopuru bikoitia edukitzea da. Hasteko, D_1 AFDa eta D_2 AFDa eraikiko ditugu, bata L_1 lengoaiarentzat eta bestea L_2 lengoaiarentzat, hurrenez hurren. 3.2.20 irudian D_1 AFDari dagokion grafikoa erakusten da eta 3.2.21 irudian D_2 AFDari dagokion grafikoa erakusten da.

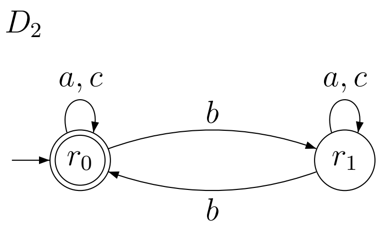


3.2.20 irudia. aaa katea duten hitzez osatutako L_1 lengoaiarentzat AFDa.

D_1 AFDa boskote bezala honela defini dezakegu:

$$(Q, A, \delta_1, q_0, Y_1)$$

Boskote horretan:



3.2.21 irudia. b kopuru bikoitia duten hitzez osatutako L_2 lengoaiarentzat AFDa.

- $Q = \{q_0, q_1, q_2, q_3\}$
- $A = \{a, b, c\}$
- $\delta_1 : Q \times A \to Q$ honako taula honen bidez definitutako trantsizio-funtzioa da:

δ	a	b	c
q_0	q_1	q_0	q_0
q_1	q_2	q_0	q_0
q_2	q_3	q_0	q_0
q_3	q_3	q_3	q_3

- q_0 hasierako egoera da.
- $Y_1 = \{q_3\}$.

Bestalde, D_2 boskote bezala honela formaliza dezakegu:

$$(R, A, \delta_2, q_0, Y_2)$$

Boskote horretako osagaien definizioak honako hauek dira:

- $R = \{r_0, r_1\}$
- $A = \{a, b, c\}$
- $\delta_2 : R \times A \to R$ honako taula honen bidez definitutako trantsizio-funtzioa da:

δ_2	a	b	c
r_0	r_0	r_1	r_0
r_1	r_1	r_0	r_1

- r_0 hasierako egoera da.
- $Y_2 = \{r_0\}$.

L lengoaiarentzat D AFD bat eraikitzeko, hau da, aaa katea izateaz gain b kopuru bikoitia duten hitzez osatutako L lengoaiarentzat AFD bat eraikitzeko, aukera bat D_1 eta D_2 erabiltzea da. Horretarako, D_1 AFDko egoera bat eta D_2 AFDko egoera bat hartuz osa daitezkeen bikote denak hartu beharko dira kontuan. D_1 AFDko egoerak $Q = \{q_0, q_1, q_2, q_3\}$ baldin badira eta D_2 AFDko egoerak $R = \{r_0, r_1\}$ baldin badira, D AFDari dagokion S egoera multzoa $Q \times R$ izango da. Beraz, Q eta R multzoen arteko biderketa kartesiarrak finkatuko du D AFD berriaren egoera multzoa:

$$S = Q \times R = \{s_0 = (q_0, r_0), s_1 = (q_0, r_1), s_2 = (q_1, r_0), s_3 = (q_1, r_1), s_4 = (q_2, r_0), s_5 = (q_2, r_1), s_6 = (q_3, r_0), s_7 = (q_3, r_1)\}$$

D AFD berriko hasierako egoera D_1 eta D_2 AFDetako hasierako egoerez osatutako bikotea izango da: (q_0, r_0). Bestalde, D AFDan onarpen egoerak D_1 AFDan eta D_2 AFDan onarpen egoerak diren egoerez osatutako bikoteak izango dira. Beste era batera esanda, $Y = Y_1 \times Y_2$ izango da. Beraz, D AFDan onarpen egoera bakarra egongo da: (q_3, r_0).

D AFDko trantsizioei dagokienez, (q_i, r_j) egoeratik (q_k, r_ℓ) egoerara A alfabetokoa den α sinboloarekin trantsizio bat egongo da baldin eta soilik baldin q_i egoeratik q_k egoerara α sinboloarekin D_1 AFDan trantsizio bat baldin badago eta r_j egoeratik r_ℓ egoerara α sinboloarekin D_2 AFDan trantsizio bat baldin badago. Era formalagoan adieraziz, $\delta((q_i, r_j), \alpha) = \delta_1 \times \delta_2((q_i, r_j), \alpha) = (\delta_1(q_i, \alpha), \delta_2(r_j, \alpha))$.

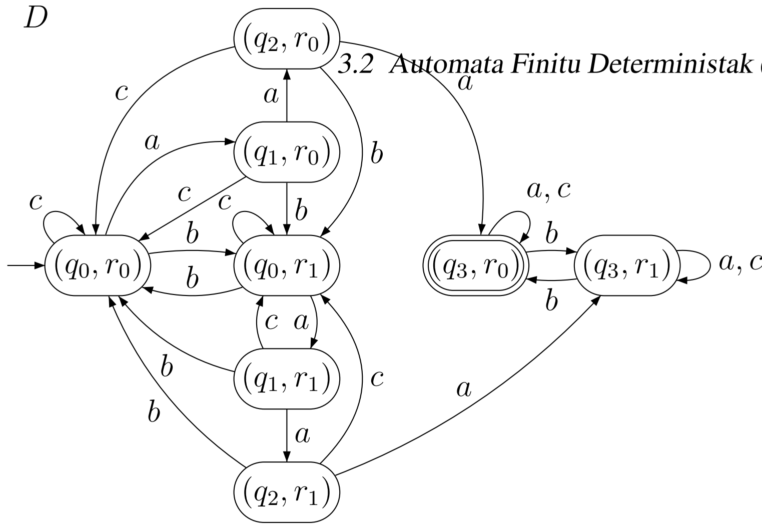
D AFDari dagokion δ trantsizio-funtzioa eraikitzeko, D_1 eta D_2 AFDen δ_1 eta δ_2 trantsizio-funtzioak erabiltzea da onena:

δ_1	a	b	c
q_0	q_1	q_0	q_0
q_1	q_2	q_0	q_0
q_2	q_3	q_0	q_0
q_3	q_3	q_3	q_3

δ_2	a	b	c
r_0	r_0	r_1	r_0
r_1	r_1	r_0	r_1

δ	a	b	c
(q_0, r_0)	(q_1, r_0)	(q_0, r_1)	(q_0, r_0)
(q_0, r_1)	(q_1, r_1)	(q_0, r_0)	(q_0, r_1)
(q_1, r_0)	(q_2, r_0)	(q_0, r_1)	(q_0, r_0)
(q_1, r_1)	(q_2, r_1)	(q_0, r_0)	(q_0, r_1)
(q_2, r_0)	(q_3, r_0)	(q_0, r_1)	(q_0, r_0)
(q_2, r_1)	(q_3, r_1)	(q_0, r_0)	(q_0, r_1)
(q_3, r_0)	(q_3, r_0)	(q_3, r_1)	(q_3, r_0)
(q_3, r_1)	(q_3, r_1)	(q_3, r_0)	(q_3, r_1)

3.2.22 irudian aaa katea duten eta b kopuru bikoitia duten hitzez osatutako L lengoaiarentzat eraiki den D AFDari dagokion trantsizio-diagrama erakusten da. D AFDa D_1 eta D_2 AFDak erabiliz eta propietateen konjuntzioa dugunerako dagoen metodoa aplikatuz lortu da. D AFDko egoeretan r_1 osagaia agertzen denean, b kopurua bakoitia da eta r_0 osagaia agertzen denean b kopurua bikoitia da. Era berean, lehenengo osagaitzat q_3 duten egoeretan badakigu aaa agertu dela dagoeneko. Aldiz, lehenengo osagaitzat q_0, q_1 edo q_2 duten egoeretan badakigu oraindik aaa ez dela agertu. Hori dela-eta, onarpen egoera bakarra (q_3, r_0) izango da. Izan ere, egoera horretan gaudenean aaa agertu da dagoeneko eta, gainera, b kopurua bikoitia da. Eraikitako D AFDa honako boskote honen bidez formalizatuko genuke: $(Q \times R, A, \delta_1 \times \delta_2, (q_0, r_0), Y_1 \times Y_2)$.



3.2.22 irudia. aaa duten eta b kopuru bikoitia duten hitzez osatutako L lengoaiarentzat AFD bat.

Beharrezkoa izanez gero, egoerak berrizenda daitezke egoerek ohiko erako etiketa izan dezaten. Horrela, 3.2.23 irudian D AFDarentzat 3.2.22 irudian erakutsi den trantsizio-diagrama bera dugu baina egoerak berrizendatuta agertzen dira.

3.2.2.4.2 aaa duten eta a sinboloaz hasten diren hitzen lengoaiarentzat AFD bat

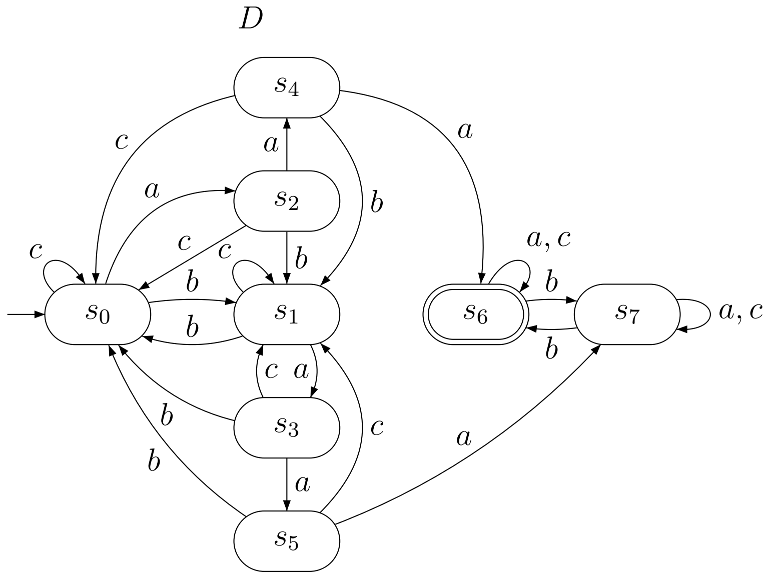
Har dezagun $A = \{a, b, c\}$ alfabetoa. aaa azpikatea duten eta, gainera, a sinboloaz hasten diren hitzez osatutako lengoaiarentzat AFD bat eraiki nahi da. L legoaia formalki honela defini daiteke:

$$L = \{w \mid w \in A^* \wedge \exists u, v(u \in A^* \wedge v \in A^* \wedge w = uaaav) \wedge \exists x(x \in A^* \wedge w = ax)\}$$

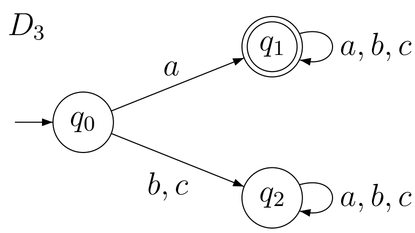
L lengoaiako hitzek bi propietaren konjuntzioa den baldintza bat bete behar dute: aaa azpikatea edukitzea eta a sinboloaz hastea. L-rentzat AFD bat eraikitzeko aukera bat, hasteko L_1 eta L_3 lengoaientzat AFD bana eraiki eta, gero, konjuntzioaren teknika erabiliz L-rentzat AFD bat lortzea da. L_1 eta L_3 lengoaiak honela defini daitezke:

$$L_1 = \{w \mid w \in A^* \wedge \exists u, v(u \in A^* \wedge v \in A^* \wedge w = uaaav)\}$$
$$L_3 = \{w \mid w \in A^* \wedge \exists x(x \in A^* \wedge w = ax)\}$$

L lengoaia $L_1 \cap L_3$ da. L_1 lengoaiako hitzek aaa azpikatea dute eta L_3 lengoaiako hitzak a sinboloaz hasten dira. Aurreko adibidean (ikus 3.2.2.4.1 atala) L_1-entzat $(Q, A, \delta_1, q_0, Y_1)$ AFD



3.2.23 irudia. aaa duten eta b kopuru bikoitia duten hitzez osatutako L lengoaiarentzat AFD bat ohiko erara etiketatuta.



3.2.24 irudia. a sinboloaz hasten diren hitzen L_3 lengoaiarentzat AFDa.

bat diseinatu da (ikus 3.2.20 irudia). Orain L_3 lengoaiarentzat D_3 AFDa diseinatuko dugu. Hain zuzen ere, 3.2.24 irudian erakusten da D_3-ren adierazpen grafikoa.

D_3 AFDa honako boskote honen bidez formaliza daiteke:

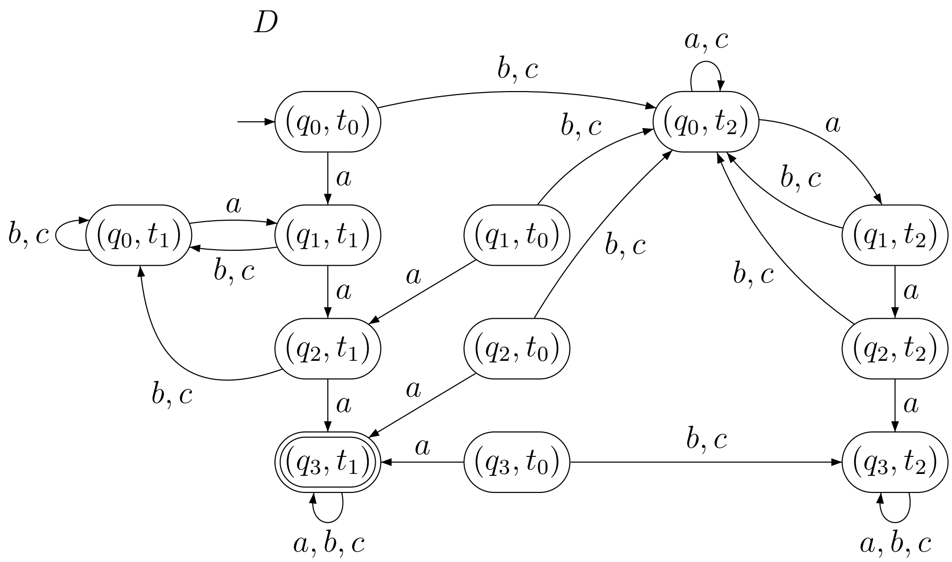
$$(T, A, \delta_3, t_0, Y_3)$$

Boskote horretako osagaiak honela defini daitezke:

- $T = \{t_0, t_1, t_2\}$
- $A = \{a, b, c\}$
- $\delta_3 : T \times A \to T$ honako trantsizio-taula honen bidez adierazitako trantsizio-funtzioa da:

δ_3	a	b	c
t_0	t_1	t_2	t_2
t_1	t_1	t_1	t_1
t_2	t_2	t_2	t_2

- t_0 hasierako egoera da.
- $Y_3 = \{t_1\}$.



3.2.25 irudia. aaa duten eta a sinboloaz hasten diren hitzez osatutako L lengoaiarentzat AFD bat.

$L = L_1 \cap L_3$ lengoaiarentzat, hau da, aaa azpikatea duten eta a sinboloaz hasten diren hitzez osatutako lengoaiarentzat $D = (S, A, \delta, s_0, Y)$ AFD bat eraikitzeko, lehenengo osagaitzat D_1-eko egoera bat eta bigarren osagaitzat D_3-ko egoera bat duten bikote denak hartu behar dira kontuan. D_1-en egoera-multzoa $Q = \{q_0, q_1, q_2, q_3\}$ baldin bada eta D_3-ren egoera-multzoa $T = \{t_0, t_1, t_2\}$ baldin bada, D-ri dagokion S egoera multzoa $Q \times T$ izango da. Beraz, Q eta T-ren arteko biderketa kartesiarrak finkatuko ditu D AFD berriko egoerak:

$$S = Q \times T = \{s_0 = (q_0, t_0), s_1 = (q_0, t_1), s_2 = (q_0, t_2), s_3 = (q_1, t_0), s_4 = (q_1, t_1), s_5 = (q_1, r_2), s_6 = (q_2, t_0), s_7 = (q_2, t_1), s_8 = (q_2, t_2), s_9 = (q_3, t_0), s_{10} = (q_3, t_1), s_{11} = (q_3, t_2)\}$$

AFD berriko s_0 hasierako egoera, abiapuntutzat hartu ditugun bi AFDetako hasierako egoerez eratutako bikotea izango da, hau da, (q_0, t_0) egoera. Onarpen egoeren Y multzoa abiapuntuko AFDetako onarpen egoerez osatutako bikoteez eratuta egongo da. Beraz, $Y = Y_1 \times Y_3$. Adibide honetan onarpen egoera bakarra (q_3, t_1) izango da.

Trantsizioei dagokienez, D AFDan (q_i, t_j) egoera batetik A-koa den α sinbolo batekin (q_k, t_ℓ) beste egoera batera trantsizioa egongo da baldin eta soilik baldin q_i egoeratik α sinboloarekin q_k egoerara trantsizioa baldin badago D_1-en eta t_j egoeratik α sinboloarekin t_ℓ egoerara trantsizioa baldin badago D_3-n. Era formalean adieraziz, $\delta((q_i, t_j), \alpha) = \delta_1 \times \delta_3((q_i, t_j), \alpha) = (\delta_1(q_i, \alpha), \delta_3(t_j, \alpha))$.

D-ri dagokion δ trantsizio-funtzioa kalkulatzeko, δ_1 eta δ_3 trantsizio-funtzioei dagozkien trantsizio-taulak erabiliz $\delta = \delta_1 \times \delta_3$ trantsizio-funtzioari dagokion trantsizio-taula lortzea da:

δ_1	a	b	c
q_0	q_1	q_0	q_0
q_1	q_2	q_0	q_0
q_2	q_3	q_0	q_0
q_3	q_3	q_3	q_3

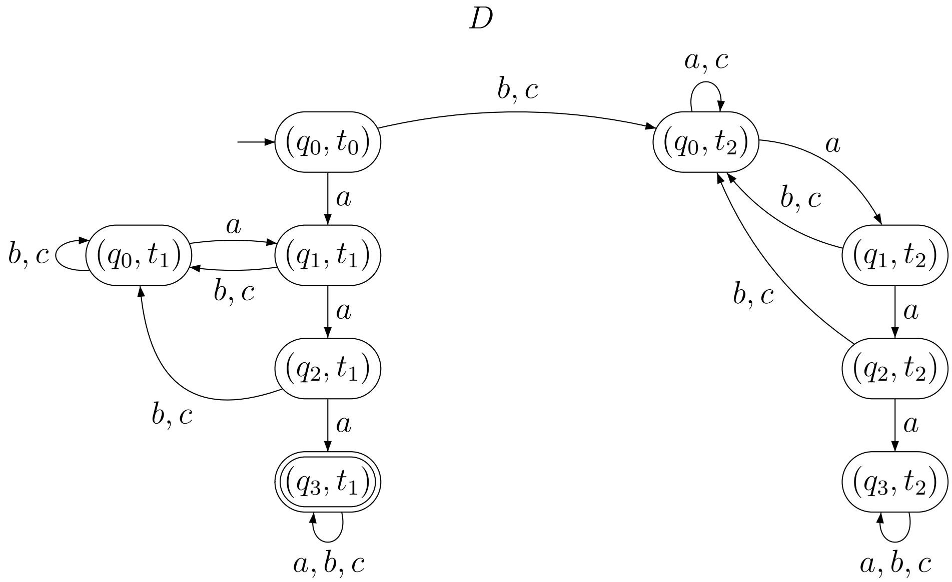
δ_3	a	b	c
t_0	t_1	t_2	t_2
t_1	t_1	t_1	t_1
t_2	t_2	t_2	t_2

δ	a	b	c
(q_0, t_0)	(q_1, t_1)	(q_0, t_2)	(q_0, t_2)
(q_0, t_1)	(q_1, t_1)	(q_0, t_1)	(q_0, t_1)
(q_0, t_2)	(q_1, t_2)	(q_0, t_2)	(q_0, t_2)
(q_1, t_0)	(q_2, t_1)	(q_0, t_2)	(q_0, t_2)
(q_1, t_1)	(q_2, t_1)	(q_0, t_1)	(q_0, t_1)
(q_1, t_2)	(q_2, t_2)	(q_0, t_2)	(q_0, t_2)
(q_2, t_0)	(q_3, t_1)	(q_0, t_2)	(q_0, t_2)
(q_2, t_1)	(q_3, t_1)	(q_0, t_1)	(q_0, t_1)
(q_2, t_2)	(q_3, t_2)	(q_0, t_2)	(q_0, t_2)
(q_3, t_0)	(q_3, t_1)	(q_3, t_2)	(q_3, t_2)
(q_3, t_1)	(q_3, t_1)	(q_3, t_1)	(q_3, t_1)
(q_3, t_2)	(q_3, t_2)	(q_3, t_2)	(q_3, t_2)

3.2.25 irudian D_1 eta D_3 abiapuntutzat hartuz eraiki den D AFDari dagokion trantsizio-diagrama erakusten da. D AFD hori aaa duten eta a sinboloaz hasten diren hitzez eratutako lengoaiarentzat definitutako AFD bat da. Egoerei dagokienez, t_1 duten den kasuetan badakigu hitza a sinboloarekin hasi dela eta t_0 edo t_2 agertzen denean ez dela a sinboloarekin hasi. Era berean, q_3 osagaia duten egoeretan badakigu aaa agertu dela eta q_0, q_1 edo q_2 duten egoeretan oraindik aaa ez dela agertu. Ondorioz, onarpen egoera bakarra (q_3, t_1) izango da, egoera horretan gaudenean aaa badaukagulako eta hitza a sinboloaz hasi delako.

Eraiki den AFDan (q_1, t_0), (q_2, t_0) eta (q_3, t_0) iristezinak dira (q_0, t_0) egoeratik eta, ondorioz, ken daitezke. 3.2.26 irudian erakusten da egoera horiek kenduz lortzen den AFDaren trantsizio-diagrama. Bestalde, trantsioen taula honela geldituko litzateke:

δ	a	b	c
(q_0, t_0)	(q_1, t_1)	(q_0, t_2)	(q_0, t_2)
(q_0, t_1)	(q_1, t_1)	(q_0, t_1)	(q_0, t_1)
(q_0, t_2)	(q_1, t_2)	(q_0, t_2)	(q_0, t_2)
(q_1, t_1)	(q_2, t_1)	(q_0, t_1)	(q_0, t_1)
(q_1, t_2)	(q_2, t_2)	(q_0, t_2)	(q_0, t_2)
(q_2, t_1)	(q_3, t_1)	(q_0, t_1)	(q_0, t_1)
(q_2, t_2)	(q_3, t_2)	(q_0, t_2)	(q_0, t_2)
(q_3, t_1)	(q_3, t_1)	(q_3, t_1)	(q_3, t_1)
(q_3, t_2)	(q_3, t_2)	(q_3, t_2)	(q_3, t_2)



3.2.26 irudia. aaa duten eta a sinboloaz hasten diren hitzez osatutako L lengoaiarentzat iristezinak diren egoerarik gabeko AFD bat.

3.2.2.4.3 Gutxienez a bat b bat eta c dituzten hitzen lengoaiarentzat AFD bat

Har dezagun $A = \{a, b, c\}$ alfabetoa. Gutxienez a bat, gutxienez b bat eta gutxienez c bat dituzten hitzez eratutako L lengoaiarentzat AFD bat eraiki nahi da. L lengoaia formalki honela defini daiteke:

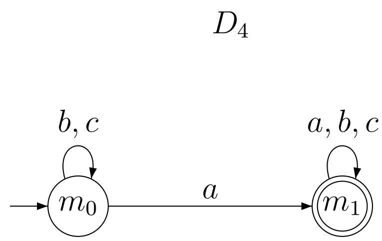
$$\begin{aligned} L = \{w \mid w \in A^* \quad & \wedge \exists u, v(u \in A^* \wedge v \in A^* \wedge w = uav) \\ & \wedge \exists u, v(u \in A^* \wedge v \in A^* \wedge w = ubv) \\ & \wedge \exists u, v(u \in A^* \wedge v \in A^* \wedge w = ucv)\} \end{aligned}$$

L lengoaiako hitzek hiru propietate bete behar dituzte: gutxienez a bat edukitzea, gutxienez b bat edukitzea eta gutxienez c bat edukitzea. L-rentzat AFD bat lortzeko, hasteko, propietate horietako bakoitzarentzat AFD bat diseinatzea eta, gero, konjuntzioaren teknika erabiltzea da aukera bat. Horretarako L_4, L_5 eta L_6 lengoaiak hartuko ditugu kontuan:

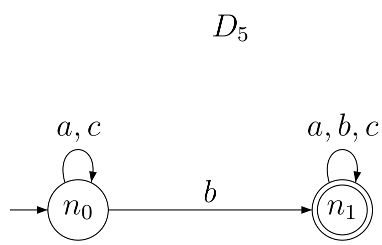
$$\begin{aligned} L_4 &= \{w \mid w \in A^* \wedge \exists u, v(u \in A^* \wedge v \in A^* \wedge w = uav)\} \\ L_5 &= \{w \mid w \in A^* \wedge \exists u, v(u \in A^* \wedge v \in A^* \wedge w = ubv)\} \\ L_6 &= \{w \mid w \in A^* \wedge \exists u, v(u \in A^* \wedge v \in A^* \wedge w = ucv)\} \end{aligned}$$

L lengoaia $L_4 \cap L_5 \cap L_6$ da. L_4 lengoaiako hitzek gutxienez a bat dute, L_5 lengoaiako hitzek gutxienez b bat dute eta L_6 lengoaiako hitzek gutxienez c bat dute. Hasteko, L_4 lengoaiarentzat D_4 AFD bat eraikiko dugu, L_5 lengoaiarentzat D_5 AFD bat eraikiko dugu eta L_6 lengoaiarentzat D_6 AFD bat eraikiko dugu. 3.2.27, 3.2.28 eta 3.2.29 irudietan D_4, D_5 eta D_6 AFDen adierazpen grafikoak erakusten dira.

D_4 AFDa honako boskote bezala formaliza daiteke:



3.2.27 irudia. Gutxienez a bat duten hitzen L_4 lengoaiarentzat AFD bat.



3.2.28 irudia. Gutxienez b bat duten hitzen L_5 lengoaiarentzat AFD bat.

$$(M, A, \delta_4, m_0, Y_4)$$

Boskote horretako osagai bakoitza honela defini daiteke:

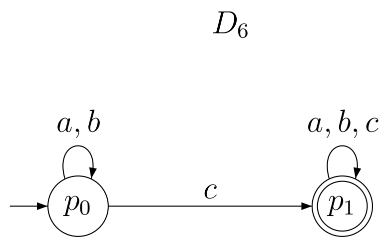
- $M = \{m_0, m_1\}$
- $A = \{a, b, c\}$
- $\delta_4 : M \times A \rightarrow M$ honako taula honen bidez definitutako trantsizio-funtzioa da:

δ_4	a	b	c
m_0	m_1	m_0	m_0
m_1	m_1	m_1	m_1

- m_0 hasierako egoera da.
- $Y_4 = \{m_1\}$.

D_5 AFDa honako boskote bezala formaliza daiteke:

$$(N, A, \delta_5, n_0, Y_5)$$



3.2.29 irudia. Gutxienez c bat duten hitzen L_6 lengoaiarentzat AFD bat.

Boskote horretako osagai bakoitza honela defini daiteke:

- $N = \{n_0, n_1\}$
- $A = \{a, b, c\}$
- $\delta_5 : N \times A \to N$ honako taula honen bidez definitutako trantsizio-funtzioa da:

δ_5	a	b	c
n_0	n_0	n_1	n_0
n_1	n_1	n_1	n_1

- n_0 hasierako egoera da.
- $Y_5 = \{n_1\}$.

D_6 AFDa honako boskote bezala formaliza daiteke:

$$(P, A, \delta_6, p_0, Y_6)$$

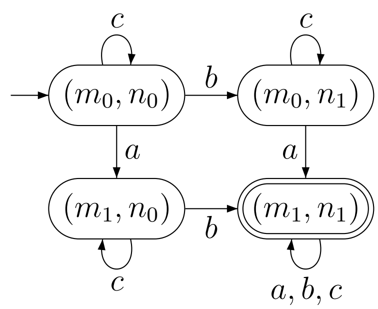
Boskote horretako osagai bakoitza honela defini daiteke:

- $P = \{p_0, p_1\}$
- $A = \{a, b, c\}$
- $\delta_6 : P \times A \to P$ honako taula honen bidez definitutako trantsizio-funtzioa da:

δ_6	a	b	c
p_0	p_0	p_0	p_1
p_1	p_1	p_1	p_1

- p_0 hasierako egoera da.

D_{45}



3.2.30 irudia. Gutxienez a bat eta b bat dituzten hitzen L_{45} lengoaiari dagokion AFD bat.

- $Y_6 = \{p_1\}$.

$L_{456} = L_4 \cap L_5 \cap L_6$ lengoaiarentzat, hau da, gutxienez a bat, b bat eta c bat dituzten hitzen lengoaiarentza $D_{456} = (S, A, \delta_{456}, s_0, Y_{456})$ AFD bat eraikitzeko D_4, D_5 eta D_6 AFDetako egoerez osatutako hirukote denak hartu behar dira kontuan. $L_4 \cap L_5 \cap L_6 = (L_4 \cap L_5) \cap L_6$ betetzen denez, aukera errazena, hasteko, D_4 eta D_5 AFDetatik abiatuz eta konjuntzioaren teknika erabiliz $L_{45} = L_4 \cap L_5$ lengoaiari dagokion D_{45} AFDa eraikitzea eta, gero, D_{45} eta D_6 AFDetatik abiatuz eta konjuntzioaren teknika erabiliz $L_{456} = (L_4 \cap L_5) \cap L_6$ lengoaiari dagokion D_{456} AFDa eraikitzea da.

D_4-ren egoeren multzoa $M = \{m_0, m_1\}$ baldin bada eta D_5-en egoera multzoa $N = \{n_0, n_1\}$, baldin bada, D_{45} AFDari dagokion X egoera multzoa $M \times N$ izango da. Beraz, D_{45} AFDaren egoera multzoa M eta N-ren arteko biderketa kartesiarrak finkatzen du:

$$X = M \times N = \{x_0 = (m_0, n_0), x_1 = (m_0, n_1), x_2 = (m_1, n_0), x_3 = (m_1, n_1)\}$$

AFD berriaren x_0 hasierako egoera, abiapuntutzat hartutako AFDetako hasierako egoerez eratutako bikotea izango da, hau da, (m_0, n_0). Onarpen egoeren Y_{45} multzoa, abiapuntuko AFDetako onarpen egoerez osatutako bikoteez eratuta egongo da. Beraz, $Y_{45} = Y_4 \times Y_5$ izango da. Kasu honetan, onarpen egoera bakarra $x_3 = (m_1, n_1)$ izango litzateke.

Trantsizioei dagokienez, D_{45} AFDan (m_i, n_j) egoera batetik (m_k, n_ℓ) beste egoera batera A alfabetokoa den α sinbolo batekin trantsizioa egongo da baldin eta soilik baldin α sinboloarekin D_4 AFDan m_i egoeratik m_k egoerara trantsizio bat baldin badago eta α sinboloarekin D_5 AFDan n_j egoeratik n_ℓ egoerara trantsizio bat baldin badago. Formalki, $\delta_{45}((m_i, n_j), \alpha) = \delta_4 \times \delta_5((m_i, n_j), \alpha) = (\delta_4(m_i, \alpha), \delta_5(n_j, \alpha))$.

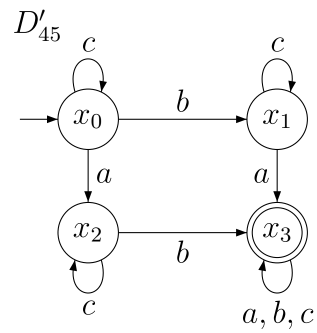
D_{45} AFDaren δ_{45} trantsizio-funtzioa kalkulatzeko, δ_4 eta δ_5 trantsizio funtzioei dagozkien taulak erabiliz δ_{45} trantsizio-funtzioari dagokion taula eraiki dezakegu:

δ_4	a	b	c
m_0	m_1	m_0	m_0
m_1	m_1	m_1	m_1

δ_5	a	b	c
n_0	n_0	n_1	n_0
n_1	n_1	n_1	n_1

δ_{45}	a	b	c
(m_0, n_0)	(m_1, n_0)	(m_0, n_1)	(m_0, n_0)
(m_0, n_1)	(m_1, n_1)	(m_0, n_1)	(m_0, n_1)
(m_1, n_0)	(m_1, n_0)	(m_1, n_1)	(m_1, n_0)
(m_1, n_1)	(m_1, n_1)	(m_1, n_1)	(m_1, n_1)

3.2.30 irudian, gutxienez a bat eta b bat dituzten hitzen lengoaiari dagokion eta D_4 eta D_5 AFDak abiapuntutzat hartuz eraiki den D_{45} AFDaren trantsizio-diagrama erakusten da. Lehenengo osagaitzat m_1 duten egoeretan badakigu dagoeneko a bat irakurri dela eta bigarren osagaitzat n_1 duten egoeretan badakigu dagoeneko b irakurri dela. Era berean, lehenengo osagaitzat m_0 duten egoeretan badakigu oraindik ez dela a-rik irakurri eta bigarren osagaitzat n_0 duten egoeretan badakigu oraindik ez dela b-rik irakurri.



3.2.31 irudia. Gutxienez a bat eta b bat dituzten hitzen L_{45} lengoaiari dagokion AFD bat ohiko eran etiketatuta.

Beharrezkoa izanez gero, egoerak ohiko eran etiketatuta egon daitezen egoera horiek berrizenda ditzakegu. 3.2.31 irudian D'_{45} AFDaren trantsizio-diagrama erakusten da. Diagrama hori D_{45} AFDari dagokion 3.2.30 irudiko trantsizio-diagrama bera da baina egoerak berrizendatu egin dira.

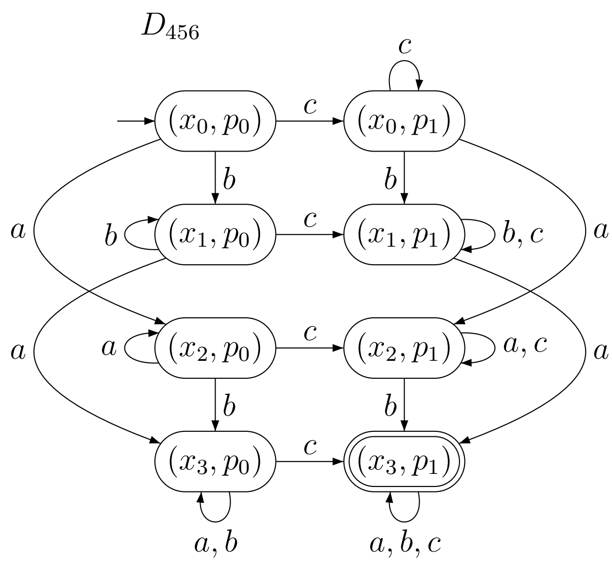
Orain D'_{45} eta D_6 AFDak erabiliz, $(L_4 \cap L_5) \cap L_6$ lengoairi dagokion $D_{456} = (S, A, \delta_{456}, s_0, Y_{456})$ AFD bat eraikiko da konjuntzioaren teknika erabiliz. Boskote horretako osagaiak honela definituta daude:

- $S = X \times P = \{(x_0, p_0), (x_0, p_1), (x_1, p_0), (x_1, p_1), (x_2, p_0), (x_2, p_1), (x_3, p_0), (x_3, p_1)\}$
- $A = \{a, b, c\}$
- $\delta_{456} : S \times A \to S$ honako taula honen bidez definitutako trantsizio-funtzioa da:

δ_{456}	a	b	c
(x_0, p_0)	(x_2, p_0)	(x_1, p_0)	(x_0, p_1)
(x_0, p_1)	(x_2, p_1)	(x_1, p_1)	(x_0, p_1)
(x_1, p_0)	(x_3, p_0)	(x_1, p_0)	(x_1, p_1)
(x_1, p_1)	(x_3, p_1)	(x_1, p_1)	(x_1, p_1)
(x_2, p_0)	(x_2, p_0)	(x_3, p_0)	(x_2, p_1)
(x_2, p_1)	(x_2, p_1)	(x_3, p_1)	(x_2, p_1)
(x_3, p_0)	(x_3, p_0)	(x_3, p_0)	(x_3, p_1)
(x_3, p_1)	(x_3, p_1)	(x_3, p_1)	(x_3, p_1)

- $s_0 = (x_0, p_0)$ hasierako egoera da.
- $Y_{456} = Y_{45} \times Y_6 = \{(x_3, p_1)\}$.

3.2.32 irudian D_{456} AFDari dagokion trantsizio-diagrama erakusten da.



3.2.32 irudia. Gutxienez a bat, b bat eta c bat dituzten hitzez osatutako L_{456} lengoaiari dagokion AFD bat.

3.2.2.5 AFDen diseinua propietateen disjuntzioarentzat (Lengoaien bildura)

L lengoaia bateko hitzek bete beharreko propietatea $\psi_1 \vee \psi_2$ erako disjuntzio baten bidez adieraz daitekeenean, ψ_1 eta ψ_2 bi propietate izanda, L lengoaia hori ψ_1 propietatea betetzen duten hitzez osatutako L_1 lengoaiaren eta ψ_2 propietatea betetzen duten hitzez osatutako L_2 lengoaiaren arteko bildura gisa defini daiteke. Beraz, $L = L_1 \cup L_2$ izango da. Gainera, L-ri dagokion AFD bat era sistematikoan edo metodikoan diseina daiteke L_1 lengoaiari dagokion AFD bat eta L_2 lengoaiari dagokion beste AFD bat erabiliz. L_1 lengoaiari dagokion

$D_1 = (Q, A, \delta_1, q_0, Y_1)$ AFDa eta L_2 lengoaiari dagokion $D_2 = (R, A, \delta_2, r_0, Y_2)$ AFDa hartuz, $D = (Q \times R, A, \delta_1 \times \delta_2, (q_0, r_0), Y_1 \times R \cup Q \times Y_2)$ AFDa $L_1 \cup L_2$ lengoaiari dagokion AFD bat izango da. D AFD horretan, $Q \times R$ egoera-multzoa Q eta R multzoen arteko biderketa kartesiarra da; beraz, lehenengo osagaitzat Q-ko elementu bat eta bigarren osagaitzat R-ko elementu bat ipiniz osa daitezkeen bikote guztiez eratutako multzoa. Bestalde, $Y_1 \times R \cup Q \times Y_2$ multzoa Y_1 eta R multzoen arteko biderketa kartesiarraren eta Q eta Y_2 multzoen arteko biderketa kartesiarraren arteko bildura da, hau da, lehenengo osagaitzat Y_1-eko elementu bat eta bigarren osagaitzat Y_2-ko elementu bat ipiniz osa daitezkeen bikote guztiez eratutako multzoa da. Azkenik, $\delta_1 \times \delta_2$ funtzioa $(Q \times R) \times A \to Q \times R$ motakoa izango da eta $(q_i, r_j) \in Q \times R$ erako edozein bikote eta $\alpha \in A$ edozein sinbolo hartuz, $\delta_1 \times \delta_2((q_i, r_j), \alpha) = (\delta_1(q_i, \alpha), \delta_2(r_j, \alpha))$ beteko da.

Hiru adibide garatuko dira teknika hau erakusteko. Konjuntzioa aztertu denean garatutako adibideetan agertu diren propietateak erabiliko ditugu hemen ere. Hala ere, propietate horien konjuntzioa kontsideratu beharrean orain disjuntzioa kontsideratuko dugu. Lehenengo adibideko lengoaiako hitzek bi propietate bete behar dituzte eta, ondorioz, lengoaia hori bi lengoairen arteko ebakidura gisa plantea daiteke. Bigarrengo adibidean ere hitzek bi baldintza bete behar dituzte baina adibide horretako azpimarragarriena honako hau da: konjuntzioaren teknika aplikatuz lortutako AFDan egoera batzuk hasierako egoeratik iristezinak dira eta, horregatik, ezaba daitezke. Hirugarren adibideko lengoaiako hitzek hiru propietate bete behar dituzte eta lengoaia horri dagokion AFDa diseinatzeko konjuntzioaren teknika bi aldiz aplikatuko da.

3.2.2.5.1 aaa katea duten edo b sinboloa kopuru bikoitian duten hitzez eratutako lengoaiarentzat AFD bat

Har dezagun $A = \{a, b, c\}$ alfabetoa. aaa katea gutxienez behin edo b sinboloa kopuru bikoitian duten hitzez eratutako L' lengoaiarentzat AFD bat diseinatu nahi da. L' lengoaiako hitzek bi propietate horietako bat edo biak bete ahal izango dituzte baina bat betetzea nahikoa da L' lengoaiakoa izateko. L' lengoaia hori formalki honela defini daiteke:

$$L' = \{w \mid w \in A^* \wedge (\exists u, v(u \in A^* \wedge v \in A^* \wedge w = uaaav) \vee ((|w|_b \bmod 2) = 0))\}$$

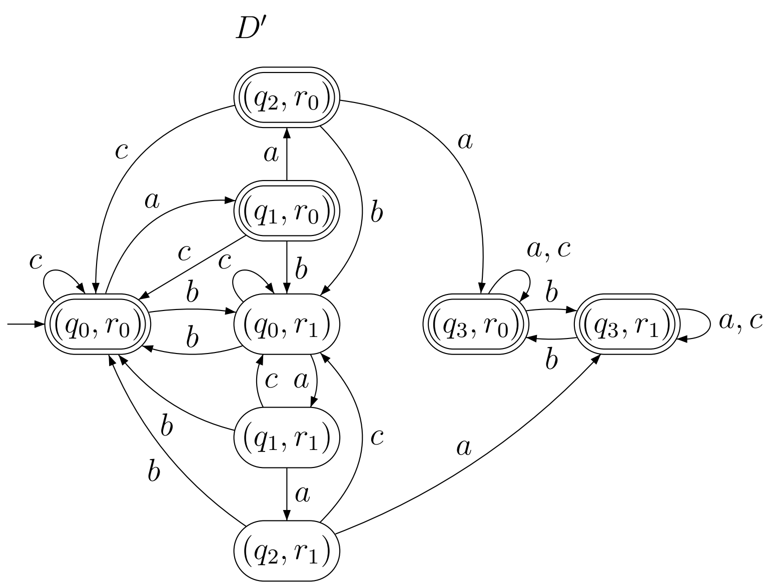
L' lengoaiako hitzek bi propietateen disjuntzioa den baldintza bat bete behar dute: gutxienez aaa bat edukitzea edo b kopuru bikoitia edukitzea. L' lengoaiarentzat AFD bat eraikitzeko, aukera bat propietate horietako bakoitza kontuan hartuz bi AFD eraikitzea da. Beste era batera esanda, konjuntzioari buruzko ataleko lehenengo adibidean (3.2.2.4.1 atalean) definitu diren L_1 eta L_2 lengoaiak kontsidera ditzakegu:

$$L_1 = \{w \mid w \in A^* \wedge \exists u, v(u \in A^* \wedge v \in A^* \wedge w = uaaav)\}$$
$$L_2 = \{w \mid w \in A^* \wedge ((|w|_b \bmod 2) = 0)\}$$

L' lengoaia $L_1 \cup L_2$ da. L_1 lengoaiako hitzek betetzen duten propietatea aaa edukitzea da eta L_2 lengoaiako hitzek betetzen duten propietatea b kopuru bikoitia edukitzea da. L' lengoaiari dagokion D' AFD bat eraikitzeko aukera bat, aurreko atalean propietateen konjuntzioa bete behar duten hitzez eratutako lengoaientzat AFD bat eraikitzeko aurkeztu den ideari jarraitzea da. Beraz, D_1 AFD bat eta D_2 AFD bat eraikiko genituzke hasteko, bata L_1 lengoaiarentzat eta bestea L_2 lengoaiarentzat, hurrenez hurren. 3.2.20 irudian D_1 AFDari dagokion grafikoa erakusten da eta 3.2.21 irudian D_2 AFDari dagokion grafikoa erakusten da. D_1 eta D_2 AFD horien boskote bidezko definizio formalak 3.2.2.4.1 atalean eman dira. Beraz, aurreko ataleko AFDak erabiltzen ari gara.

Lortuko den D' AFDak 3.2.2.4.1 atalean lortu den AFDak dituen osagai berak ditu, onarpen egoeren multzoa izan ezik. Onarpen egoeren multzoa $\{(q_0, r_0), (q_1, r_0), (q_2, r_0), (q_3, r_0), (q_3, r_1)\}$ izango da.

3.2.33 irudian aaa katea duten edo b kopuru bikoitia duten hitzez osatutako L' lengoaiarentzat eraiki den D' AFDari dagokion trantsizio-diagrama erakusten da. D' AFDa D_1 eta D_2 AFDak erabiliz eta propietateen disjuntzioa dugunerako dagoen metodoa aplikatuz lortu da. D' AFDko egoeretan r_1 osagaia agertzen denean, b kopurua bakoitia da eta r_0 osagaia agertzen denean b kopurua bikoitia da. Era berean, q_3 osagai bezala duten egoeretan badakigu aaa agertu dela dagoeneko. Aldiz, osagai bezala q_0, q_1 edo q_2 duten egoeretan badakigu oraindik aaa ez dela agertu. Hori dela-eta, onarpen egoerak (q_0, r_0), (q_1, r_0), (q_2, r_0), (q_3, r_0) eta (q_3, r_1) izango dira. Izan ere, egoera horietan gaudenean aaa agertu da dagoeneko edo b kopurua bikoitia da.



3.2.33 irudia. aaa duten edo b kopuru bikoitia duten hitzez osatutako L' lengoaiarentzat AFD bat.

3.2.2.5.2 aaa duten edo a-rekin hasten diren hitzen lengoaiari dagokion AFD baten diseinua

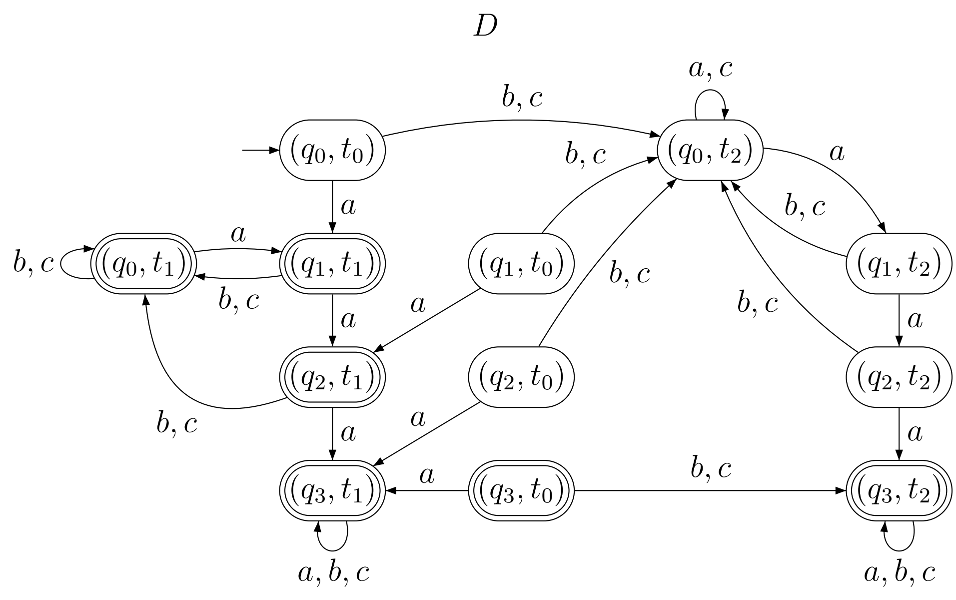
Har dezagun $A = \{a, b, c\}$ alfabetoa. aaa azpikatea duten edo a sinboloarekin hasten diren hitzen L lengoaiari dagokion AFD bat diseinatu nahi da. L lengoaia honela defini daiteke era formalean:

$$L = \{w \mid w \in A^* \wedge (\exists u, v(u \in A^* \wedge v \in A^* \wedge w = uaaav) \vee \exists x(x \in A^* \wedge w = ax))\}$$

Lengoaia horretako hitzek bi propietateren disjuntzioa den baldintza bat bete behar dute: aaa edukitzea edo a-rekin hastea. Aukera bat, propietate bakoitzarentzat AFD bat diseinatzea da. Beraz, honela definituko genituzkeen L_1 eta L_3 lengoaiak hartuko genituzke:

$$L_1 = \{w \mid w \in A^* \wedge \exists u, v(u \in A^* \wedge v \in A^* \wedge w = uaaav)\}$$
$$L_3 = \{w \mid w \in A^* \wedge \exists x(x \in A^* \wedge w = ax)\}$$

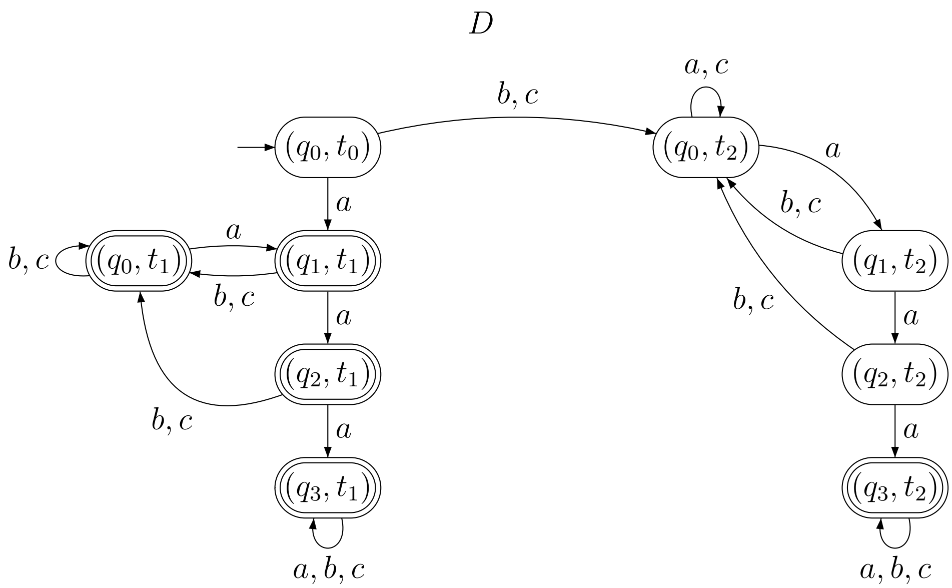
L lengoaia $L_1 \cup L_3$ da. L_1 lengoaiako hitzek aaa azpikatea dute eta L_3 lengoaiako hitzak a sinboloarekin hasten dira. Konjuntzioari buruzko atalean (3.2.2.4.1 eta 3.2.2.4.2 azpiataletan) L_1 lengoaiarentzat $(Q, A, \delta_1, q_0, Y_1)$ AFD bat (ikus 3.2.20 irudia) eta L_3 lengoaiarentzat $(T, A, \delta_3, t_0, Y_3)$ AFD bat diseinatu dira (ikus 3.2.20 eta 3.2.24 irudiak).



3.2.34 irudia. aaa duten edo a-rekin hasten diren hitzen L lengoaiarentzat AFD bat.

$L = L_1 \cup L_3$ lengoaiarentzat, hau da, aaa duten eta a-rekin hasten diren hitzen lengoaiarentzat $D = (S, A, \delta, s_0, Y)$ AFD bat eraikitzeko prozesua $L_1 \cap L_3$ lengoaiarentzat AFD bat eraikitzeko jarraitu den prozesu bera da (ikus 3.2.2.4.2 atala). $L_1 \cup L_3$ lengoaiarentzat eraikitako AFDa $L_1 \cap L_3$ lengoaiarentzat eraikitako AFDaren berdina izango da onarpen egoeren multzoa izan ezik.

D_1 eta D_3 AFDak erabiliz aaa duten edo a-rekin hasten diren hitzen lengoaiarentzat eraiki den D AFDari dagokion trantsizio-diagrama 3.2.34 irudian erakusten da. t_1 duten bikoteetan badakigu hitza a-rekin hasi dela eta, aldiz, t_0 edo t_2 duten bikoteetan badakigu hitza ez dela a-rekin hasi. Era berean, q_3 duten bikoteetan badakigu aaa agertu dela baina, bestalde, q_0, q_1 edo q_2 duten bikoteetan oraindik ez da agertu aaa.



3.2.35 irudia. aaa duten edo a-rekin hasten diren hitzen L lengoaiari dagokion AFD bat, iristezinak diren egoerarik gabe.

Eraiki den AFDan (q_1, t_0), (q_2, t_0) eta (q_3, t_0) egoerak iristezinak dira (q_0, t_0) egoeratik eta, horregatik, ezaba daitezke. 3.2.35 irudian erakusten da iristezinak diren egoerak ezabatu ondoren lortu den AFDari dagokion trantsizio-diagrama.

3.2.2.5.3 Gutxienez a bat edo b bat edo c bat duten hitzen lengoaiari dagokion AFD baten diseinua

Har dezagun $A = \{a, b, c\}$ alfabetoa. Gutxienez a bat edo gutxienez b bat edo gutxienez c bat duten hitzen L lengoaiari dagokion AFD bat diseinatu nahi da. L lengoaia formalki honela defini daiteke:

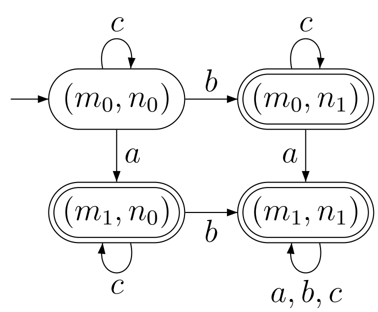
$$\begin{aligned} L = \{w \mid w \in A^* \quad & \wedge (\exists u, v(u \in A^* \wedge v \in A^* \wedge w = uav) \\ & \vee \exists u, v(u \in A^* \wedge v \in A^* \wedge w = ubv) \\ & \vee \exists u, v(u \in A^* \wedge v \in A^* \wedge w = ucv))\} \end{aligned}$$

Lengoaia horretako hitzek hiru propietateren disjuntzioa den baldintza bat bete behar dute: Gutxienez a bat edo gutxienez b bat edo gutxienez c bat edukitzea. Hasteko, propietate horietako bakoitzarentzat AFD bat eraiki dezakegu. Beste era batera esanda, honako L_4, L_5 eta L_6 lengoaiak kontsidera ditzakegu:

$$L_4 = \{w \mid w \in A^* \wedge \exists u, v(u \in A^* \wedge v \in A^* \wedge w = uav)\}$$
$$L_5 = \{w \mid w \in A^* \wedge \exists u, v(u \in A^* \wedge v \in A^* \wedge w = ubv)\}$$
$$L_6 = \{w \mid w \in A^* \wedge \exists u, v(u \in A^* \wedge v \in A^* \wedge w = ucv)\}$$

L lengoaia $L_4 \cup L_5 \cup L_6$ da. L_4 lengoaiako hitzek gutxienez a bat dute, L_5 lengoaiako hitzek gutxienez b bat dute eta L_6 lengoaiako hitzek gutxienez c bat dute. 3.2.27, 3.2.28 eta 3.2.29 irudietan 3.2.2.4.3 atalean diseinatu diren D_4, D_5 eta D_6 AFDei dagozkien grafikoak ditugu.

D



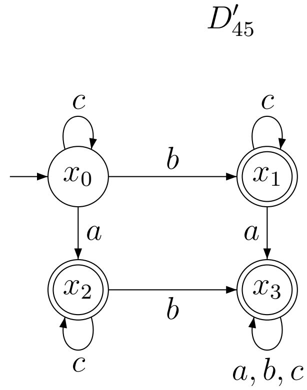
3.2.36 irudia. Gutxienez a bat edo gutxienez b bat duten hitzen L_{45} lengoaiari dagokion AFD bat.

L lengoaiarentzat, hau da, gutxienez a bat edo gutxienez b bat edo gutxienez c bat hitzen lengoaiarentzat $D_{456} = (S, A, \delta, s_0, Y)$ AFD bat eraikitzeko, D_4ko, D_5-eko eta D_6-ko egoerez osatutako hirukote guztiak hartu beharko dira kontuan. $L_4 \cup L_5 \cup L_6 = (L_4 \cup L_5) \cup L_6$ betetzen denez, bide errazena, hasteko, D_4 eta D_5 erabiliz $L_4 \cup L_5$ lengoaiarentzat D_{45} AFD bat eraitzea eta, gero, D_{45} eta D_6 erabiliz $(L_4 \cup L_5) \cup L_6$ lengoaiarentzat D_{456} AFD bat eraikitzea da.

$L_4 \cup L_5$ lengoaiari dagokion AFDa $L_4 \cap L_5$ lengoaiari dagokion AFDaren berdina da onarpen egoeren multzoa izan ezik. $L_4 \cup L_5$ lengoaiari dagokion onarpen egoeren multzoa $\{x_1 = (m_0, n_1), x_2 = (m_1, n_0), x_3 = (m_1, n_1)\}$ da. 3.2.36 irudian, D_4 eta D_5 AFDak erabiliz, gutxienez a bat edo gutxienez b bat duten hitzen lengoaiari dagokion D_{45} AFda dugu. m_1 duten egoeretan badakigu gutxienez a bat egon dela eta n_1 duten egoeretan badakigu gutxienez b bat egon dela. Bestalde, m_0 duten egoeretan badakigu oraindik a-rik ez dela agertu eta, n_0 duten egoeretan badakigu oraindik b-rik ez dela agertu.

Beharrezkoa izanez gero edo nahia izanez gero, egoerak berrizendatu daitezke ohiko eran etiketatuta ipintzeko. 3.2.36 irudian D_{45} AFDko egoerak berretiketatuz lortu den D'_{45} AFDaren trantsizio-diagrama dugu.

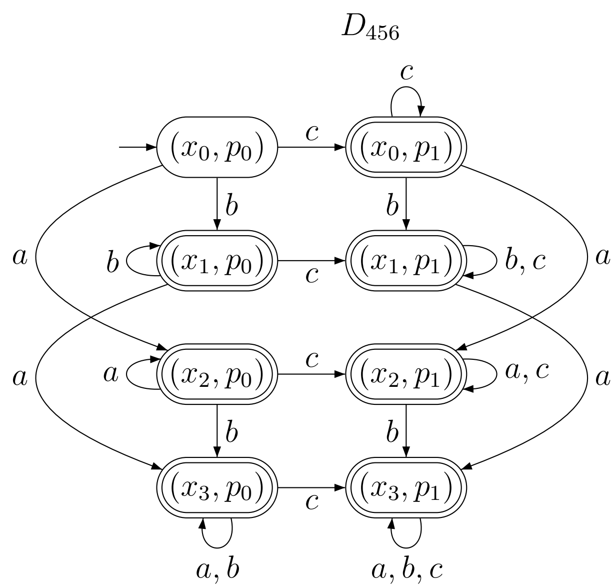
Orain D'_{45} eta D_6 erabiliz, $(L_4 \cup L_5) \cup L_6$ lengoaiari dagokion $(S, A, \delta_{456}, s_0, Y_{456})$ AFDa eraiki daiteke. AFD hori $(L_4 \cap L_5) \cap L_6$ lengoaiarentzat eraiki den AFDaren berdina izango



3.2.37 irudia. AFD con etiquetado habitual para el lenguaje L_{45} de las palabras que contienen al menos una a o al menos una b.

da onarpen egoeren multzoa izan ezik. Onarpen egoeren multzoan (x_0, p_0) ez eta beste bikote denak egongo dira.

3.2.38 irudian erakusten da eraikitako AFDari dagokion trantsizio-diagrama.



3.2.38 irudia. Gutxienez a bat edo gutxienez b edo gutxienez c bat duten hitzen L_{456} lengoaiari dagokion AFD bat.

3.2.3 AFDen minimizazioa

AFDen diseinua lantzerakoan, 10 azpikatearekin bukatzen diren hitzez osatutako lengoaiaren adibidean erakutsi den bezala, lengoaia bakoitzarentzat AFD desberdinak diseina daitezke (3.2.5 eta 3.2.6 irudiak begiratu). AFDen arteko desberdintasuna egoera-kopuruan egon ohi da.

Gai honetan AFD bat nola minimiza daitekeen azalduko da. AFD bat emanda, beharrezkoak ez diren egoerak ezabatu eta lortu ahal den AFD txikiena itzultzen duen metodo bat aplikatuko dugu.

3.2.3.1 Minimizatzeko metodoak

AFD bat minimizatzeko metodo bat baino gehiago daude baina metodo horietan denetan baliokideak diren egoerak identifikatu eta baliokideak diren egoera horietatik bakarra mantentzea da helburua. Hemen azalduko den metodoan egoerak multzotan sailkatuz joango gara. Multzo bakoitzean baliokideak izan daitezkeen egoerak egongo dira. Multzo bateko egoeren jokabideak aztertuz multzo horretako egoerak azpimultzotan banandu behar al diren ala ez erabakiko da.

3.2.3.2 Minimizazioaren adibideak

Metodoa adibideak garatuz azalduko da.

3.2.3.2.1 10 azpikatearekin bukatzen diren hitzez osatutako lengoaiari dagokion AFD baten minimizazioa

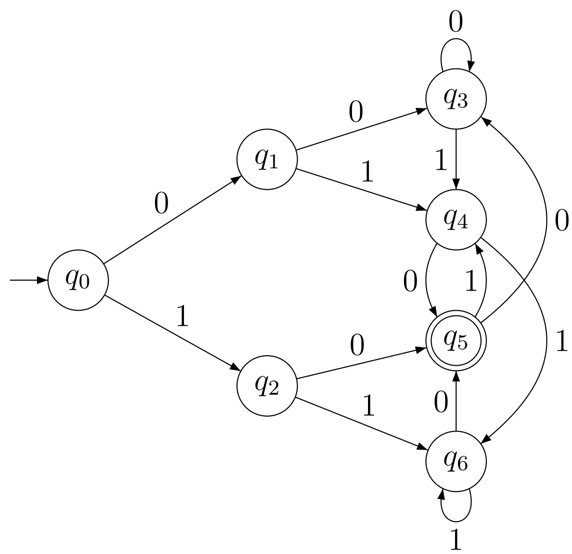
3.2.39 irudian $\{w \mid w \in A^* \land \exists u(u \in A^* \land w = u10)\}$ lengoaiari dagokion AFD bat ikus daiteke. Lengoaia hori $A = \{0, 1\}$ alfabetoaren gainean definituta dago.

AFD hori minimizatzeko jarraitu beharreko urratsak azalduko dira jarraian. Metodoa jokaera bera duten egoerak zein diren identifikatzean datza. Egoera batzuk beti jokaera bera badute, egoera horietako bat laga eta beste denak ezaba daitezke. Horrela, jokaera mota bakoitzeko egoera bat bakarrik geldituko da eta gelditzen diren egoera horiek beharrezkoak direla jakingo dugu.

Kontuan izan AFD bateko egoera bakoitzak memoria bezala jokatzen duela eta egoera bakoitza baldintzaren bat gogoratzeko erabiltzen dela. Baldintza bera gogoratzen duten egoera bat baino gehiago edukitzea alferrikakoa da. Horregatik metodo honek baldintza bera gogoratzen duten egoeretatik bat bakarrik mantenduko du.

Lehenengo zatiketa Lehenengo zatiketan bi multzo sortuko dira. Batean Y multzokoak direnak sartuko dira (onarpen egoerak) eta bestean Y multzokoak ez direnak (onarpen egoerak ez direnak). Multzo batean agertzen den azpiindize txikiena j baldin bada, multzoa $[q_j]$ bezala identifikatuko dugu.

$$[q_0] = \{q_0, q_1, q_2, q_3, q_4, q_6\}$$
$$[q_5] = \{q_5\}$$



3.2.39 irudia. 10 azpikatearekin bukatzen diren hitzez osatutako lengoaiari dagokion AFDa.

Bigarren zatiketa Gogoan izan δ trantsizio-funtzioari dagokion taula honako hau dela:

δ	0	1
q_0	q_1	q_2
q_1	q_3	q_4
q_2	q_5	q_6
q_3	q_3	q_4
q_4	q_5	q_6
q_5	q_3	q_4
q_6	q_5	q_6

Orain egoera bakoitzetik zein egoeratara joango garen adierazi beharrean, egoera bakoitzetik zein multzotara joango garen adieraziz taula hori eraldatu egingo dugu. Horrela, q_1 egoeran gaudenean 0 sinboloa irakurriz q_3 egoerara joango garenez, eta q_3 egoera $[q_0]$ multzoan dagoenez, q_3 ipini beharrean $[q_0]$ ipiniko dugu. Kasu guztietan aldaketa hori egin behar da:

δ	0	1
q_0	$[q_0]$	$[q_0]$
q_1	$[q_0]$	$[q_0]$
q_2	$[q_5]$	$[q_0]$
q_3	$[q_0]$	$[q_0]$
q_4	$[q_5]$	$[q_0]$
q_5	$[q_0]$	$[q_0]$
q_6	$[q_5]$	$[q_0]$

Jarraian $[q_0]$ eta $[q_5]$ multzoak zatitu behar al diren aztertuko da. $[q_5]$ multzoan elementu bakarra dagoenez, hor ezin da zatiketarik egin. Beraz $[q_0]$ multzoan zentratuko gara.

$[q_0]$ multzoko egoeren jokaera aztertuz, q_0, q_1 eta q_3 egoerek jokaera bera dutela ikus dezakegu: 0 sinboloa irakurriz gero $[q_0]$ multzoko egoera batera doaz eta 1 sinboloa irakurriz $[q_0]$ multzoko egoera batera doaz. Beste aldetik, q_2, q_4 eta q_6 egoerek beste jokaera bat dute: 0 sinboloa irakurriz gero $[q_5]$ multzoko egoera batera doaz eta 1 sinboloa irakurriz $[q_0]$ multzoko egoera batera doaz. Ondorioz, bi jokaera desberdin horiek kontuan hartuz $[q_0]$ multzoa bitan zatituko dugu. Guztira hiru multzo geldituko zaizkigu:

$$
\begin{aligned}
[q_0] &= \{q_0, q_1, q_3\} \\
[q_2] &= \{q_2, q_4, q_6\} \\
[q_5] &= \{q_5\}
\end{aligned}
$$

GARRANTZITSUA: q_5 egoerak $[q_0]$ multzoko egoeren jokaera bera du, baina hala ere ez dira multzo berean sartu behar. Metodo honetan multzoak zatituz joan behar dugu, baina multzo desberdietako egoerak ez dira inoiz nahastu behar, nahiz eta jokaera bera izan.

Ez dago hirugarren zatiketarik Bigarren zatiketa kontuan hartuz trantsizio-taula eguneratu egingo dugu:

δ	0	1
q_0	$[q_0]$	$[q_2]$
q_1	$[q_0]$	$[q_2]$
q_2	$[q_5]$	$[q_2]$
q_3	$[q_0]$	$[q_2]$
q_4	$[q_5]$	$[q_2]$
q_5	$[q_0]$	$[q_2]$
q_6	$[q_5]$	$[q_2]$

Jarraian $[q_0]$ eta $[q_2]$ multzoak zatitu beharrik ba al dagoen erabakiko dugu. Lehen esan den bezala, $[q_5]$ multzoan elementu bakarra dagoenez ezin da egin zatiketarik. Beraz, $[q_0]$ eta $[q_2]$ multzoetan zentratuko gara.

$[q_0]$ multzokoak diren q_0, q_1 eta q_3 egoerek jokaera bera dutela ikus dezakegu: 0 sinboloa irakurtzean $[q_0]$ multzoko egoera batera doaz eta 1 sinboloa irakurriz $[q_2]$ multzoko egoera batera doaz. Beraz $[q_0]$ multzoa ez da zatitu behar. $[q_2]$ multzoan ere antzekoa gertatzen da. Multzo horretakoak diren q_2, q_4 eta q_6 egoerek jokaera bera dute: 0 sinboloa irakurtzen bada, $[q_5]$ multzoko egoera batera doaz eta 1 sinboloa irakurtzen bada, $[q_2]$ multzoko egoera batera doaz. Beraz, $[q_2]$ multzoa ere ez da zatitu behar.

GARRANTZITSUA: q_5 egoerak $[q_0]$ multzoko egoeren jokaera bera du, baina hala ere ez dira multzo berean sartu behar. Metodo honetan multzoak zatituz joan behar dugu, baina multzo desberdietako egoerak ez dira inoiz nahastu behar, nahiz eta jokaera bera izan.

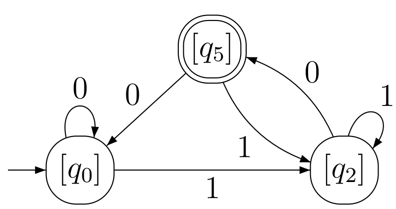
AFD txikiena Hiru egoera multzo daudela ikusi dugu metodoa aplikatuz:

$$[q_0] = \{q_0, q_1, q_3\}$$
$$[q_2] = \{q_2, q_4, q_6\}$$
$$[q_5] = \{q_5\}$$

Egoera multzo bakoitza egoera bat izango da AFD berrian. Azkeneko taula kontuan hartuz egoera berrien arteko trantsizioak kalkulatu beharko dira:

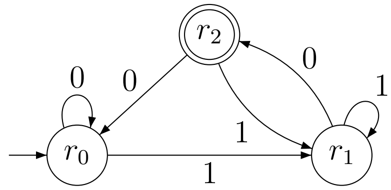
δ	0	1
q_0	$[q_0]$	$[q_2]$
q_2	$[q_5]$	$[q_2]$
q_5	$[q_0]$	$[q_2]$

Minimizatu beharreko AFDko q_0 hasierako egoera $[q_0]$ multzoan dagoenez, $[q_0]$ izango da AFD berriko hasierako egoera. Bestalde onarpen egoera $[q_5]$ izango da, minimizatu beharreko AFDan onarpen egoerak diren egoerez osatuta dagoelako. AFD berria 3.2.40 irudian ikus daiteke.



3.2.40 irudia. 3.2.39 irudiko AFDa minimizatuz lortu den AFD berria.

AFD berriko egoerak $0, 1, 2, 3, \ldots$ ordenean zenbatuta joan daitezen, berrizendatu egin ohi dira. Berrizendatu ondoren gelditu den AFD berria 3.2.41 irudian ikus daiteke.

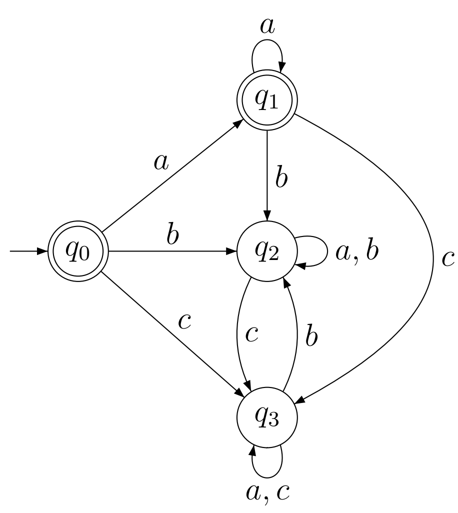


3.2.41 irudia. 3.2.40 irudiko AFDa berrizendatuz lortu den AFDa.

3.2.3.2.2 b-rik eta c-rik ez duten hitzez osatutako lengoaiari dagokion AFD baten minimizazioa

3.2.42 irudian $A = \{a, b, c\}$ alfabetoaren gainean definitutako $\{w \mid w \in A^* \wedge |w|_a = |w|\}$ lengoaiari dagokion AFD bat ikus dezakegu.

AFD hori minimizatzeko jarraitu beharreko urratsak azalduko dira jarraian. Lehenago esan den bezala, metodo honetan jokaera bera duten egoerak zein diren erabakitzen da eta jokaera mota bakoitzeko egoera bat bakarrik lagatzen da, beste egoerak ezabatuz. AFDetan egoerak memoria bezala erabiltzen direnez eta egoera bakoitzak baldintzaren bat gogoratzeko balio duenez, baldintza bera gogoratzen duten egoera bat baino gehiago edukitzea alferrikakoa da. Horregatik metodo honek baldintza bera gogoratzen duten egoeretatik bat bakarrik mantenduko du.



3.2.42 irudia. b-rik eta c-rik ez duten hitzez osatutako lengoaiari dagokion AFDa.

Lehenengo zatiketa Lehenengo zatiketan bi multzo sortuko dira. Batean Y multzokoak direnak sartuko dira (onarpen egoerak) eta bestean Y multzokoak ez direnak. Multzo batean agertzen den azpiindize txikiena j baldin bada, multzoa $[q_j]$ bezala identifikatuko dugu.

$$[q_0] = \{q_0, q_1\}$$
$$[q_2] = \{q_2, q_3\}$$

Ez dago bigarren zatiketarik δ trantsizio-funtzioari dagokion taula honako hau da:

δ	a	b	c
q_0	q_1	q_2	q_3
q_1	q_1	q_2	q_3
q_2	q_2	q_2	q_3
q_3	q_3	q_2	q_3

Orain egoera bakoitzetik zein egoeratara joango garen adierazi beharrean, egoera bakoitzetik zein multzotara joango garen adieraziz taula hori eraldatu egingo dugu. Horrela, q_1 egoeran gaudenean a sinboloa irakurriz q_1 egoerara joango garenez, eta q_1 egoera $[q_0]$ multzoan dagoenez, q_1 ipini beharrean $[q_0]$ ipiniko dugu. Kasu guztietan aldaketa hori egin behar da:

δ	a	b	c
q_0	$[q_0]$	$[q_2]$	$[q_2]$
q_1	$[q_0]$	$[q_2]$	$[q_2]$
q_2	$[q_2]$	$[q_2]$	$[q_2]$
q_3	$[q_2]$	$[q_2]$	$[q_2]$

Jarraian $[q_0]$ eta $[q_2]$ multzoak zatitu behar al diren erabakiko da.

$[q_0]$ multzokoak diren q_0 eta q_1 egoerek jokaera bera dutela ikus dezakegu: a sinboloa irakurtzean $[q_0]$ multzoko egoera batera doaz, b sinboloa irakurriz $[q_2]$ multzoko egoera batera doaz eta c sinboloa irakurriz $[q_2]$ multzoko egoera batera doaz. Beraz, $[q_0]$ multzoa ez da zatitu behar.

$[q_2]$ multzoan ere antzekoa gertatzen da. $[q_2]$ multzoan q_2 eta q_3 egoerek jokaera bera dutela ikus dezakegu: a sinboloa irakurtzean $[q_2]$ multzoko egoera batera doaz, b sinboloa irakurriz $[q_2]$ multzoko egoera batera doaz eta c sinboloa irakurriz $[q_2]$ multzoko egoera batera doaz. Beraz, $[q_2]$ multzoa ez da zatitu behar.

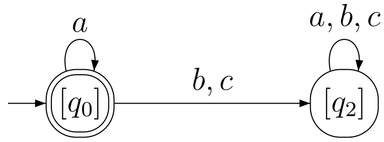
AFD txikiena Beraz bi multzo gelditu dira:

$$[q_0] = \{q_0, q_1\}$$
$$[q_2] = \{q_2, q_3\}$$

Multzo bakoitza egoera bat izango da AFD minimizatuan. Egoeren arteko trantsizioak eraiki dugun azkeneko taula erabiliz kalkulatu behar dira:

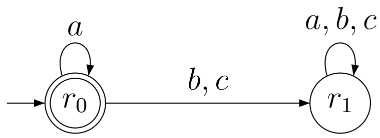
δ	a	b	c
q_0	$[q_0]$	$[q_2]$	$[q_2]$
q_2	$[q_2]$	$[q_2]$	$[q_2]$

Minimizatu beharreko AFDko q_0 hasierako egoera $[q_0]$ multzoan dagoenez, $[q_0]$ izango da AFD berriko hasierako egoera. Bestalde, onarpen egoera $[q_0]$ izango da, minimizatu beharreko AFDan onarpen egoerak diren egoerez osatuta dagoelako. AFD berria 3.2.43 irudian ikus daiteke.



3.2.43 irudia. 3.2.42 irudiko AFDa minimizatuz lortu den AFDa.

AFD berriko egoerak $0, 1, 2, 3, \ldots$ ordenean zenbatuta joan daitezen, berrizendatu egin ohi dira. Berrizendatu ondoren gelditu den AFD berria 3.2.44 irudian ikus daiteke.



3.2.44 irudia. 3.2.43 irudiko AFDko egoerak berrizendatuz lortutako AFDa.

3.2.3.2.3 $a^j b^k$ erako hitzez osatutako lengoaiari dagokion AFD baten minimizazioa

3.2.45 irudian $A = \{a, b\}$ alfabetoaren gainean definitutako honako lengoaia honi dagokion AFD bat ikus dezakegu:

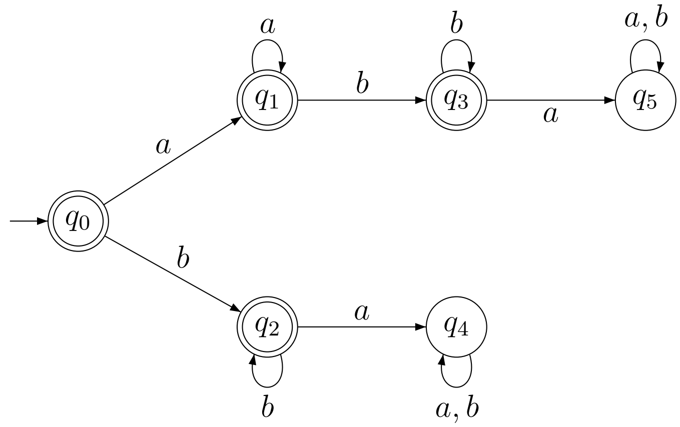
$$\{w \mid w \in A^* \wedge \exists u, v(u \in A^* \wedge v \in A^* \wedge |u| = |u|_a \wedge |v| = |v|_b \wedge w = uv)\}$$

Lengoaia hori beste era honetara ere defini daiteke:

$$\{w \mid w \in A^* \wedge \exists j, k(j \in I\!N \wedge k \in I\!N \wedge w = a^j b^k)\}$$

Hor $I\!N$ zenbaki arrunten multzoa da: $I\!N = \{0, 1, 2, 3, \ldots\}$.

AFD hori minimizatzeko jarraitu beharreko urratsak azalduko dira jarraian. Aurreko adibideetan ere esan den bezala, metodo honetan jokaera bera duten egoerak zein diren erabakitzen da eta jokaera mota bakoitzeko egoera bat bakarrik lagatzen da, beste egoerak ezabatuz. AFDetan egoerak memoria bezala erabiltzen direnez eta egoera bakoitzak baldintzaren bat gogoratzeko balio duenez, baldintza bera gogoratzen duten egoera bat baino gehiago edukitzea alferrikakoa da. Horregatik metodo honek baldintza bera gogoratzen duten egoeretatik bat bakarrik mantenduko du.



3.2.45 irudia. $a^j b^k$ erako hitzez osatutako lengoaiari dagokion AFDa.

Lehenengo zatiketa Lehenengo zatiketan bi multzo sortuko dira. Batean Y multzokoak direnak sartuko dira (onarpen egoerak) eta bestean Y multzokoak ez direnak. Multzo batean agertzen den azpiindize txikiena j baldin bada, multzoa $[q_j]$ bezala identifikatuko dugu.

$$[q_0] = \{q_0, q_1, q_2, q_3\}$$
$$[q_4] = \{q_4, q_5\}$$

Bigarren zatiketa δ trantsizio funtzioari dagokion taula honako hau da:

δ	a	b
q_0	q_1	q_2
q_1	q_1	q_3
q_2	q_4	q_2
q_3	q_5	q_3
q_4	q_4	q_4
q_5	q_5	q_5

Orain egoera bakoitzetik zein egoeratara joango garen adierazi beharrean, egoera bakoitzetik zein multzotara joango garen adieraziz taula hori eraldatu egingo dugu. Horrela, q_1 egoeran gaudenean b sinboloa irakurriz q_3 egoerara joango garenez, eta q_3 egoera $[q_0]$ multzoan dagoenez, q_3 ipini beharrean $[q_0]$ ipiniko dugu. Kasu guztietan aldaketa hori egin behar da:

δ	a	b
q_0	$[q_0]$	$[q_0]$
q_1	$[q_0]$	$[q_0]$
q_2	$[q_4]$	$[q_0]$
q_3	$[q_4]$	$[q_0]$
q_4	$[q_4]$	$[q_4]$
q_5	$[q_4]$	$[q_4]$

Jarraian $[q_0]$ eta $[q_4]$ multzoak zatitu behar al diren aztertuko da.

$[q_0]$ multzoko egoeren jokaera aztertuz, q_0 eta q_1 egoerek jokaera bera dutela ikus dezakegu: a sinboloa irakurriz gero $[q_0]$ multzoko egoera batera doaz eta b sinboloa irakurriz $[q_0]$ multzoko egoera batera doaz. Beste aldetik, q_2 eta q_3 egoerek beste jokaera bat dute: a sinboloa irakurriz gero $[q_4]$ multzoko egoera batera doaz eta b sinboloa irakurriz $[q_0]$ multzoko egoera batera doaz. Ondorioz, bi jokaera desberdin horiek kontuan hartuz $[q_0]$ multzoa bitan zatituko dugu.

$$[q_0] = \{q_0, q_1\}$$
$$[q_2] = \{q_2, q_3\}$$

$[q_4]$ multzoan, q_4 eta q_5 egoerek jokaera bera dute: a sinboloa irakurriz gero $[q_4]$ multzoko egoera batera doaz eta b sinboloa irakurriz $[q_4]$ multzoko egoera batera doaz. Beraz, $[q_4]$ multzoa ez da zatitu behar.

Bigarren zatiketaren ondoren hiru multzo gelditzen dira:

$$[q_0] = \{q_0, q_1\}$$
$$[q_2] = \{q_2, q_3\}$$
$$[q_4] = \{q_4, q_5\}$$

Ez dago hirugarren zatiketarik Bigarren zatiketa kontuan hartuz trantsizio-taula eguneratu egingo dugu:

δ	a	b
q_0	$[q_0]$	$[q_2]$
q_1	$[q_0]$	$[q_2]$
q_2	$[q_4]$	$[q_2]$
q_3	$[q_4]$	$[q_2]$
q_4	$[q_4]$	$[q_4]$
q_5	$[q_4]$	$[q_4]$

Jarraian $[q_0]$, $[q_2]$ eta $[q_4]$ multzoak zatitu beharrik ba al dagoen erabakiko dugu.

$[q_0]$ multzokoak diren q_0 eta q_1 egoerek jokaera bera dutela ikus dezakegu: a sinboloa irakurtzean $[q_0]$ multzoko egoera batera doaz eta b sinboloa irakurriz $[q_0]$ multzoko egoera batera doaz. Beraz, $[q_0]$ multzoa ez da zatitu behar. $[q_2]$ multzoan ere antzekoa gertatzen da. Multzo horretakoak diren q_2 eta q_3 egoerek jokaera bera dute: a sinboloa irakurtzen bada, $[q_4]$ multzoko egoera batera doaz eta b sinboloa irakurtzen bada, $[q_2]$ multzoko egoera batera doaz. Beraz, $[q_2]$ multzoa ere ez da zatitu behar. Bukatzeko, $[q_4]$ multzoa ere ez da zatitu behar. Izan ere, q_4 eta q_5 egoerek jokaera bera dute: a sinboloa irakurtzen bada, $[q_4]$ multzoko egoera batera doaz eta b sinboloa irakurtzen bada, $[q_4]$ multzoko egoera batera doaz.

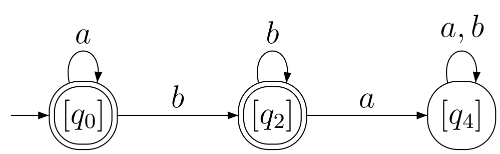
AFD txikiena Hiru egoera multzo daudela ikusi dugu metodoa aplikatuz:

$$\begin{aligned}[q_0] &= \{q_0, q_1\}\\ [q_2] &= \{q_2, q_3\}\\ [q_5] &= \{q_4, q_5\}\end{aligned}$$

Egoera-multzo bakoitza egoera bat izango da AFD berrian. Azkeneko taula kontuan hartuz egoera berrien arteko trantsizioak kalkulatu beharko dira:

δ	a	b
q_0	$[q_0]$	$[q_2]$
q_2	$[q_4]$	$[q_2]$
q_4	$[q_4]$	$[q_4]$

Minimizatu beharreko AFDko q_0 hasierako egoera $[q_0]$ multzoan dagoenez, $[q_0]$ izango da AFD berriko hasierako egoera. Bestalde, onarpen egoerak $[q_0]$ eta $[q_2]$ izango dira, minimizatu beharreko AFDan onarpen egoerak diren egoerez osatuta daudelako. AFD berria 3.2.46 irudian ikus daiteke.



3.2.46 irudia. 3.2.45 irudiko AFDa minimizatuz lortutako AFDa.

AFD berriko egoerak $0, 1, 2, 3, \ldots$ ordenean zenbatuta joan daitezen, berrizendatu egin ohi dira. Egoerak berrizendatu ondoren gelditu den AFD berria 3.2.47 irudian ikus daiteke.

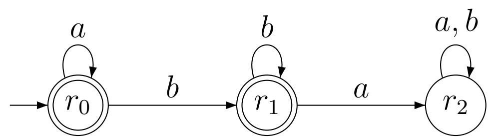
3.2.3.2.4 a eta b sinboloak kopuru bikoitian dituzten hitzen lengoaiari dogokion AFD baten minimizazioa

3.2.48 irudian honako lengoaia honi dagokion AFD bat ikus daiteke:

$$\{w \mid w \in A^* \wedge |w|_a \bmod 2 = 0 \wedge |w|_b \bmod 2 = 0\}$$

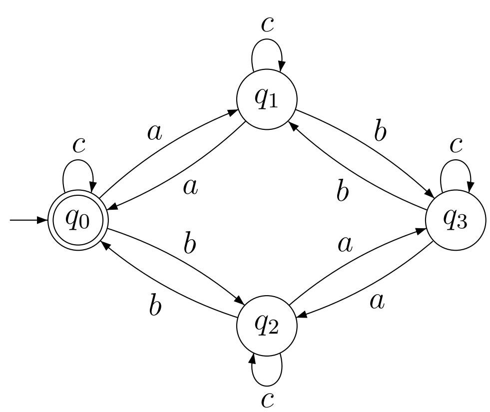
Lengoaia hori $A = \{a, b, c\}$ alfabetoaren gainean definituta dago.

AFD hori minimizatzeko jarraitu beharreko urratsak azalduko dira jarraian. Metodo honetan jokaera bera duten egoerak zein diren erabakitzen da eta jokaera mota bakoitzeko egoera bat



3.2.47 irudia. 3.2.46 irudiko AFDa berrizendatuz lortu den AFDa.

bakarrik lagatzen da, beste egoerak ezabatuz. AFDetan egoerak memoria bezala erabiltzen direnez eta egoera bakoitzak baldintzaren bat gogoratzeko balio duenez, baldintza bera gogoratzen duten egoera bat baino gehiago edukitzea alferrikakoa da. Horregatik metodo honek baldintza bera gogoratzen duten egoeretatik bat bakarrik mantenduko du.



3.2.48 irudia. a eta b kopuru bikoitian dituzten hitzen lengoaiari dagokion AFD bat.

Lehenengo zatiketa Lehenengo zatiketan bi multzo sortuko dira. Batean Y multzokoak direnak sartuko dira (onarpen egoerak) eta bestean Y multzokoak ez direnak. Multzo batean agertzen den azpiindize txikiena j baldin bada, multzoa $[q_j]$ bezala identifikatuko dugu.

$$[q_0] = \{q_0\}$$
$$[q_1] = \{q_1, q_2, q_3\}$$

Bigarren zatiketa δ trantsizio-funtzioari dagokion taula honako hau da:

δ	a	b	c
q_0	q_1	q_2	q_0
q_1	q_0	q_3	q_1
q_2	q_3	q_0	q_2
q_3	q_2	q_1	q_3

Orain egoera bakoitzetik zein egoeratara joango garen adierazi beharrean, egoera bakoitzetik zein multzotara joango garen adieraziz taula hori eraldatu egingo dugu. Horrela, q_1 egoeran gaudenean b sinboloa irakurriz q_3 egoerara joango garenez, eta q_3 egoera $[q_1]$ multzoan dagoenez, q_3 ipini beharrean $[q_1]$ ipiniko dugu. Kasu guztietan aldaketa hori egin behar da:

δ	a	b	c
q_0	$[q_1]$	$[q_1]$	$[q_0]$
q_1	$[q_0]$	$[q_1]$	$[q_1]$
q_2	$[q_1]$	$[q_0]$	$[q_1]$
q_3	$[q_1]$	$[q_1]$	$[q_1]$

Jarraian $[q_0]$ eta $[q_1]$ multzoak zatitu behar al diren aztertuko da.

$[q_0]$ multzoak egoera bakarra duenez, ezin da gehiago zatitu. $[q_1]$ multzoko egoeren jokaera aztertuz, q_0, q_1 eta q_2 egoerek jokaera desberdina dutela ikus dezakegu. Hasteko, q_1 egoeran egonda, a edo b sinboloak irakurriz gero $[q_1]$ multzoko egoera batera joan beharko da eta c sinboloa irakurriz gero $[q_0]$ multzoko egoera batera joan beharko da. Bestalde, q_2 egoeran egonda, a edo c sinboloak irakurriz gero $[q_1]$ multzoko egoera batera joan beharko da eta b sinboloa irakurriz gero $[q_0]$ multzoko egoera batera joan beharko da. Azkenik, q_3 egoeran egonda, a, b edo c sinboloak irakurriz gero $[q_1]$ multzoko egoera batera joan beharko da. Ondorioz, hiru jokaera desberdin horiek kontuan hartuz $[q_1]$ multzoa hirutan zatituko dugu.

$$
\begin{aligned}
[q_0] &= \{q_0\} \\
[q_1] &= \{q_1\} \\
[q_2] &= \{q_2\} \\
[q_3] &= \{q_3\}
\end{aligned}
$$

Ez dago hirugarren zatiketarik Multzo bakoitzak egoera bakarra duenez, ez dago multzo horiek gehiago zatitzerik. Beraz, behin betiko zatiketa honako hau da:

$$
\begin{aligned}
[q_0] &= \{q_0\} \\
[q_1] &= \{q_1\} \\
[q_2] &= \{q_2\} \\
[q_3] &= \{q_3\}
\end{aligned}
$$

Ondorioz, 3.2.48 irudiko AFDtik ezin da egoerarik ezabatu, AFD hori izan daitekeen txikiena da adibide honetako lengoaiarentzat.

3.3.

Automata Finitu Ez-Deterministak (AFEDak)

3.3.1 AFEDak definitzeko arrazoiak: AFDetan hobetu daitezkeen ezaugarriak

Automata finitu deterministek (AFDek) hobetu daitezkeen ezaugarri batzuk badituzte. Ezaugarri horiek hobetzeko helburuarekin aldaketak sartzeak, automata finitu ez-deterministen (AFEDen) agerpena dakar berarekin. Ondorengo ataletan AFDetan hobetu daitezkeen ezaugarriak zehaztuko dira eta AFEDen definizioa emango da.

AFDetan hobetu daitezkeen oinarrizko ezaugarriak honako bi gai hauekin lotuta daude batez ere:

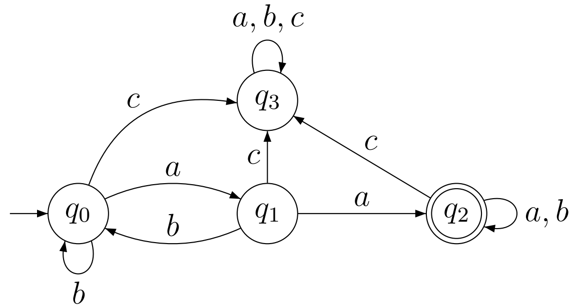
- Alde batetik, batzuetan hitz osoa irakurri baino lehen erantzuna zein izango den jakin daiteke ("Bai" edo "Ez") baina hala ere AFDak hitz osoa irakurri beharko du beti.
- Bestetik, batzuetan disjuntzio baten aurrean disjuntzioaren osagai bakoitza bere aldetik aztertzeak bukaerako erantzuna zein izango den ("Bai" edo "Ez") erabakitzea azkartu dezake, baina AFD batean disjuntzioen osagaiak ez dira bakoitza bere aldetik kalkulu independenteen bidez (paraleloan) aztertzen.

Orain ezaugarri horiek xehetasun handiagoarekin aztertuko ditugu eta, gainera, disjuntzioaren barnean bi azpikasu bereiziko ditugu: disjuntzio inplizitua (edo ezkutua) eta disjuntzio esplizitua (edo agerikoa).

3.3.1.1 Behin betiko erantzuna ezagutu bezain laster gelditzeko aukera

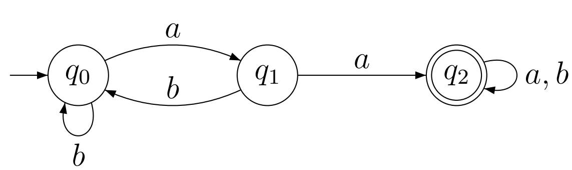
3.3.1 irudian aa duten baina c-rik ez duten hitzez eratutako lengoaiarentzat diseinatutako AFD baten trantsizio-diagrama erakusten da (3.2.9 irudiko diagrama bera da).

3.3.1 irudiko trantsizio-diagraman, edozein egoeratan egonda ere c sinboloa irakurtzen bada q_3 egoerara igaro beharko dela zehazten da. Behin q_3 egoerara iritsi ondoren behin betiko



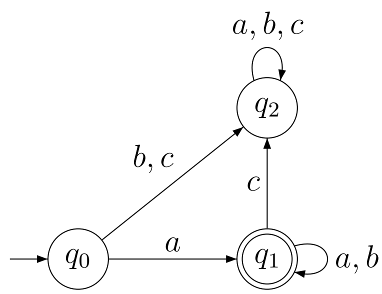
3.3.1 irudia. aa katea baduten baina c-rik ez duten hitzen lengoaiarentzat AFD bat.

erantzuna “Ez” izango dela badakigu, baina AFD hori kontuan hartuz, hitza irakurtzen jarraitu beharko litzateke hitz osoa bukatu arte. Horrek eraginkortasun maila txikiagoa izatea dakar berarekin, denbora alferrik galtzen baita. AFDen ezaugarri horren aurrean, behin betiko erantzuna “Ez” izango dela jakindakoan automatak berehala gelditu beharko duela (hitza osatzen duten gainerako sinboloak irakurri gabe) eta “Ez” erantzuna aurkeztu beharko duela dioen araua ezarri daiteke. Arau berri hori formalizatzeko, q_3 egoera ezabatuko da eta beste egoeratan c sinboloarekin ez da trantsiziorik egongo. Automatak c sinboloa irakurtzen badu, trantsiziorik ez dagoenez berehala gelditu egingo da eta “Ez” erantzungo du. Beraz, automata mota berri honetan automata egoera batean dagoenean eta sinbolo bat irakurtzen duenean, sinbolo horrentzat egoera horretatik trantsiziorik ez badago, automata gelditu egingo da eta “Ez” erantzungo du. Arau berri hau kontuan hartzen duten automatei Automata Finitu Ez-Deterministak (AFEDak) deitzen zaie, sinbolo eta egoera batzuentzat ez baitago zehaztuta edo determinatuta trantsizioa. Eskuetan dugun adibidera itzuliz, 3.3.2 irudian aldaketa hau kontuan hartuz lortuko litzatekeen AFEDaren trantsizio-diagrama ikus daiteke.



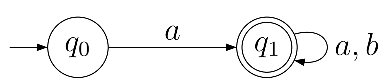
3.3.2 irudia. aa katea baduten baina c-rik ez duten hitzen lengoaiarentzat AFED bat.

3.3.2 irudiko AFEDa gelditu egingo da erantzuna ezezkoa izango dela detektatu bezain laster. Erantzunak ezezkoa izan behar duela jakindakoan berehala gelditzea posible dela ikusi ondoren, erantzuna baiezkoa izango dela jakindakoan gelditzerik ba al dagoen aztertuko da orain. Horretarako beste arau berri bat beharko litzateke. Esate baterako, AFEDa zirkulu bikoitza duen egoera batean baldin badago eta irakurritako sinboloarekin trantsiziorik ez badago, gelditu eta “Bai” erantzun dezala AFEDak. Horrelako arau bat ipin daiteke baina ezin da lortu AFEDa bai baiezko kasuetan eta bai ezezko kasuetan gelditzea. Izan ere, batzuetan AFEDak ezin badu jarraitu irakurtzen, ez litzateke jakingo AFEDak “Bai” ala “Ez” erantzun behar duen.



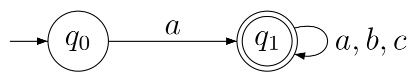
3.3.3 irudia. a sinboloaz hasi eta c-rik ez duten hitzen lengoaiarentzat AFD bat.

Adibidez, 3.3.3 irudian a sinboloaz hasi eta c sinboloaren agerpenik ez duten hitzen lengoaiari dagokion AFD baten trantsizio-diagrama erakusten da. Diagrama horretan ikus daitekeen bezala, q_0 egoeran b sinboloa irakurtzen bada edo edozein egoeratan c irakurtzen bada, q_2 egoerara igaro beharko da. q_2 egoerara iritsitakoan badakigu erantzunak “Ez” izan behar duela baina AFD horren diseinua kontuan hartuz hitz osoa irakurri beharko da. Bukaerako erantzuna “Ez” izango dela jakin bezain laster gelditu ahal izateko q_2 egoera ezaba daiteke eta, horrela, ez da trantsiziorik egongo c sinboloarekin edozein egoeratan egonda ere eta ezta b sinboloarekin ere q_0 egoeran gaudenean.



3.3.4 irudia. a sinboloaz hasi eta c-rik ez duten hitzen lengoaiarentzat AFED bat.

3.3.4 irudian erakusten da aldaketa horiek kontuan hartuz eraikitako AFEDaren trantsizio-diagrama. AFED horretan, q_0 egoeran b irakurtzen bada edo q_0 egoeran edo q_1 egoeran c irakurtzen bada, trantsiziorik ez dagoenez, AFEDa gelditu egingo da eta “Ez” erantzungo du. Baina AFEDa bi zirkulu dituen egoera batean dagoenean irakurri den sinboloarentzat trantsiziorik ez badago, gelditu eta “Bai” erantzun behar dela dioen araua ipini badugu, orduan q_1 egoeran c irakurritakoan arazoa izango genuke, “Bai” erantzungo bailuke AFEDak eta erantzuna zuzena izateko “Ez” erantzun beharko luke AFEDak. Adibide horretan garbi erakusten den bezala, automata finituen oraingo eredua mantenduz ezin da erantzuna ezagutu bezain laster gelditu bai baiezko kasuetan eta bai ezezko kasuetan. Automata finituen oraingo eredua mantendu nahi dugunez, erantzuna ezezkoa izango dela jakindakoan berehala gelditzea aukeratuko dugu. Erantzuna baiezkoa izango dela jakinda ere hitz osoa irakurri beharko da.



3.3.5 irudia. a sinboloaz hasten diren hitzen lengoaiarentzat AFED bat.

3.3.5 irudian a sinboloaz hasten diren hitzez osatutako lengoaiari dagokion AFED baten trantsizio-diagrama dugu. AFED horretan, $abbcaac$ bezalako hitzen kasuan lehenengo sinboloa irakurritakoan behin betiko erantzuna baiezkoa izango dela jakin arren hitz osoa irakurri beharko da. Aldiz, $bcaabbabbc$ erako hitzen kasuan, lehenengo sinboloa irakurritakoan badakigu behin betiko erantzuna ezezkoa izango dela eta AFEDa gelditu egingo da hitzeko gainerako sinboloak irakurri gabe.

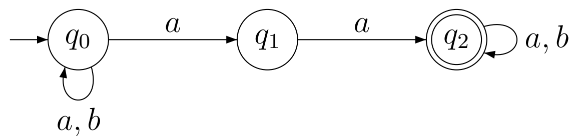
3.3.1.2 Sinbolo bati lotutako disjuntzio inplizitua edo ezkutua

Automata finitu bateko egoera batean gaudenean, gerta daiteke sinbolo baten esanahiari dagokionez aukera desberdinak egotea. 3.3.2 irudiko AFEDa hartuko dugu berriro. AFED horretako q_0 egoeran a sinboloa irakurtzen bada, a hori a bakarti bat izan daiteke eta, beraz, aa agertzearen baldintza betearazteko balio ez duena ($babbb$ hitzean bezala) edo a hori aa erako kate bateko lehenengo osagaia izan daiteke ($baabbb$ hitzeko lehenengo a bezala). Adibidez, $babbaabaa$ hitzean lehenengo a ez da aa erako kate baten hasiera baina bigarren a eta laugarren a badira aa erako kate baten hasiera. Lehenengo a-ren kasuan q_1 egoerara joan beharko da eta gero berriro q_0 egoerara itzuli. Baina badago hori planteatzeko beste era bat ere. q_0 egoeran gaudela a irakurritakoan a-ri dagokion esanahiarekin lotuta dagoen disjuntzio inplizitu edo ezkutu baten aurrean gaude (oraintxe irakurri den a hori ez da aa erako kate baten hasiera edo oraintxe irakurri den a hori aa erako kate baten hasiera da). Disjuntzio hori q_0 egoeratik a sinboloarekin bi trantsizio ipiniz adieraz daiteke: trantsizio batek q_0 egoeran bertan mantentzea ahalbidetuko luke eta besteak q_1 egoerara igarotzea ahalbidetuko luke. Bi trantsizio horietakoren bat zuzena izango da. Kalkulu horiek biak edo aukera biak bakoitza bere aldetik gara daitezke eta kalkulu horietakoren baten garapenean hitz osoa irakurtzea eta q_2 egoeran bukatzea lortzen bada, automatak “Bai” erantzungo du. Aldaketa hori kontuan hartuz diseinatutako AFEDari dagokion trantsizio-diagrama 3.3.6 irudian dago.

Egoera batean sinbolo batentzat trantsizio bat baino gehiago ipintzeko aukerak lengoaiako hitzen egitura hobeto islatzea edo adieraztea ahalbidetzen du:

$$\{w|w \in A^* \wedge \exists u, v(u \in A^* \wedge v \in A^* \wedge |u|_c = 0 \wedge |v|_c = 0 \wedge w = uaav)\}$$

q_0 egoerak u zatia adierazten edo islatzen du, gero bi a datoz eta, bukatzeko, q_2 egoerak v zatia adierazten du.



3.3.6 irudia. aa katea bai baina c-rik ez duten hitzen lengoaiarentzat bigarren AFEDa.

3.3.1.3 Disjuntzio esplizitua edo agerikoa lengoaiako hitzek bete behar duten propietatean

Har dezagun $A = \{a, b, c\}$ alfabetoa. aaa katea duten edo a sinboloarekin bukatzen diren hitzez osatutako L lengoaia kontsideratuko dugu:

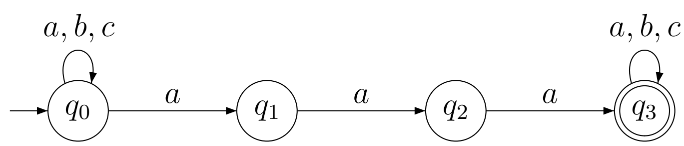
$$L = \{w | w \in A^* \wedge (\exists u, v(u \in A^* \wedge v \in A^* \wedge w = uaaav) \vee \exists u(u \in A^* \wedge w = ua))\}$$

Lengoaia horretako hitzek bete beharreko propietatea disjuntiboa da eta honako bi lengoaia hauek kontsidera ditzakegu:

$$L_1 = \{w | w \in A^* \wedge \exists u, v(u \in A^* \wedge v \in A^* \wedge w = uaaav)\}$$
$$L_2 = \{w | w \in A^* \wedge \exists u(u \in A^* \wedge w = ua)\}$$

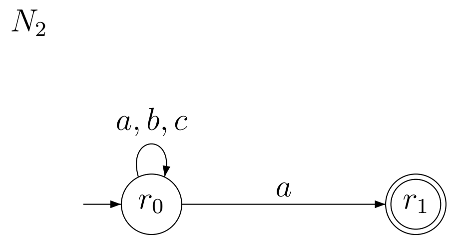
L lengoaiarentzat automata finitu bat eraikitzea nahi izanez gero, aukera bat hasteko L_1 eta L_2 lengoaientzat AFD bana eraikitzea eta gero 3.2.2.5 atalean aurkeztutako metodoa erabiliz L lengoaiarentzat AFDa diseinatzea litzateke. Beste aukera bat, L_1 eta L_2 lengoaientzako eraikitako bi AFEDtatik abiatzea eta kalkulu paraleloak bakoitza bere aldetik aurrera eramateko AFEDek duten ahalmenari etekina ateratzea litzateke.

N_1



3.3.7 irudia. aaa duten hitzen L_1 lengoaiarentzat AFED bat.

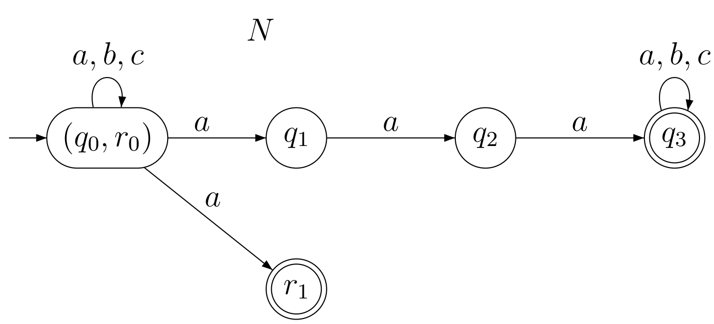
3.3.7 irudian L_1 lengoaiarentzat diseinatutako N_1 AFED bati dagokion trantsizio-diagrama dugu eta 3.3.8 irudian L_2 lengoaiarentzat diseinatutako N_2 AFED bati dagokion trantsizio-diagrama dugu. L_2 lengoaiarentzat diseinatutako N_2 AFEDan, a sinboloaren aurrean bi aukera



3.3.8 irudia. a sinboloaz bukatzen diren hitzen L_2 lengoaiarentzat AFED bat.

daude: bukaerako a izatea eta kasu horretan q_1 egoerara joan behar izatea edo bukaerako a ez izatea, eta kasu horretan q_0 egoeran mantendu behar izatea. Azken batean, a bat agertzen denean ez dakigu azkeneko posizioan dagoen a al den ala azkeneko posiziokoa ez den a bat al den. Horregatik, AFEDak aukera biak aztertuko ditu: alde batetik, "demagun ez dela azkeneko posizioko a" eta, bestetik, "demagun azkeneko posizioko a dela". Suposizio horietako bat zuzena izango da.

L lengoaiarentzat AFED bat lortzeko L_1 eta L_2 lengoaientzat diseinatutako AFEDak elkartzea nahikoa da, hasierako egoerak bat eginez. Oro har, hasierako egoerak bat egitean kontuz ibili beharko da, askotan moldaketa batzuk egitea beharrezkoa izan ohi baita. N_1 eta N_2 AFED horiek elkartzetik lortutako N AFEDari dagokion trantsizio-diagrama 3.3.9 irudian dugu.



3.3.9 irudia. aaa duten edo a sinboloarekin bukatzen diren hitzen lengoaiarentzat AFED bat.

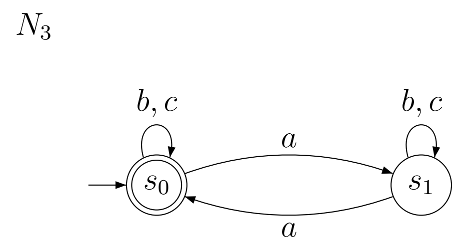
Kasu honetan, hasierako egoera biak bat egitean ez da beharrezkoa inolako moldaketarik. N AFED horretako (q_0, r_0) egoeran gaudenean, a sinboloa irakurtzen bada, hiru aukera daude:

- a hori aaa kate bateko lehenengo a izatea eta q_1 egoerara igaro behar izatea.
- Bukaerako a izatea eta r_1 egoerara igaro behar izatea
- Esanguratsua ez den a bat izatea, ez delako ez bukaerako a eta ez aaa bateko lehenengo a edo aaa bateko lehenengo a izanda ere, geroago beste aaa bat agertuko delako. Hirugarren aukera honetan AFEDa (q_0, r_0) egoeran mantenduko da.

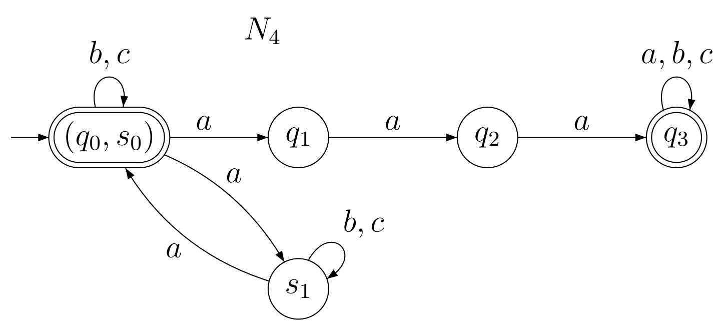
Bi AFED bat egitean, askotan doitze edo aldaketa batzuen beharra sortzen da trantsizioei dagokienez. Hori gertatzen deneko kasu bat ikusi ahal izateko, a kopuru bikoitia duten hitzez eratutako L_3 lengoaia eta aaa katea edo a kopuru bikoitia duten hitzez eratutako L_4 lengoaia hartuko ditugu.

$$L_3 = \{w | w \in A^* \wedge |w|_a \bmod 2 = 0\}$$
$$L_4 = \{w | w \in A^* \wedge (\exists u, v(u \in A^* \wedge v \in A^* \wedge w = uaaav) \vee (|w|_a \bmod 2 = 0))\}$$

L_4 lengoaia L_1 eta L_3 lengoaien bildura da: $L_4 = L_1 \cup L_3$.



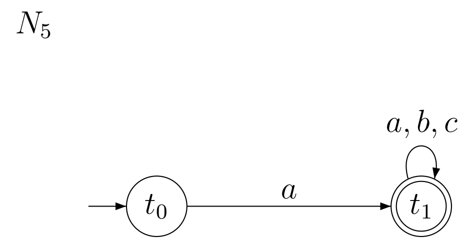
3.3.10 irudia. a kopuru bikoitia duten hitzen L_3 lengoaiarentzat AFED bat.



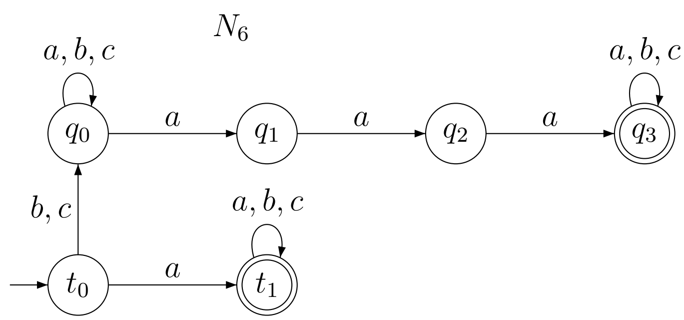
3.3.11 irudia. aaa duten edo a kopuru bikoitia duten hitzen L_4 lengoaiarentzat AFED bat.

3.3.10 irudian L_3 lengoaiarentzat eraikitako N_3 AFED baten trantsizio-diagrama dugu eta 3.3.11 irudian L_4 lengoaiarentzat eraikitako N_4 AFED baten trantsizio-diagrama dugu. N_4 AFEDan, (q_0, s_0) hasierako egoerak zirkulu bikoitza behar du nahiz eta N_1 AFEDko q_0 egoerak zirkulu bikoitzik ez izan (ikus 3.3.7 irudia). Gainera, N_4 AFEDko (q_0, s_0) egoerak ez du a sinboloa bere begiztan nahiz eta N_1 AFEDko q_0 egoerak a sinboloa izan bere begiztan.

Egin beharreko doitzeak edo aldaketak oraindik zailagoak edo nahasiagoak ere izan daitezke. Hori garbi ikusten da N_1 AFEDa (3.3.7 irudia) eta a sinboloarekin hasten diren hitzez eratutako lengoaiari dagokion N_5 AFEDa (3.3.12 irudia) elkartzen edo bat egiten saiatzen bagara. Bat-egite horretatik lortzen den N_6 AFEDa (3.3.42 irudia) aaa katea duten edo a sinboloarekin hasten diren hitzez eratutako L_6 lengoaiari dagokio:



3.3.12 irudia. a sinboloarekin hasten diren hitzen L_5 lengoaiarentzat AFED bat.



3.3.13 irudia. aaa duten edo a sinboloarekin hasten diren hitzen L_6 lengoaiarentzat AFED bat.

$$L_5 = \{w | w \in A^* \wedge \exists u(u \in A^* \wedge w = au)\}$$
$$L_6 = \{w | w \in A^* \wedge (\exists u, v(u \in A^* \wedge v \in A^* \wedge w = uaaav) \vee \exists u(u \in A^* \wedge w = au))\}$$

N_6 eraikitzean, q_0 eta t_0 egoerak ezin dira (q_0, t_0) egoera bakar bat bezala ipini, t_0 egoerak ezin baitezake begiztarik eduki. Izan ere, t_0 egoerak b eta c sinboloak etiketa bezala dituen begizta bat balu, bai N_5 AFEDak —eta baita N_6 AFEDak— baiezkoa erantzungo lukete $bbcabbb$ erako hitzentzat, baina hitz hori ez da hasten a sinboloarekin eta hitz horrek ez du aaa katerik. Hala ere, L lengoaia bat beste bi lengoairen bildura gisa adieraz daitekeenean, bi lengoaia horiei dagozkien AFEDak erabiliz L-rentzat AFED bat eraikitzeko metodo sistematiko bat azalduko da AFEDen diseinñuari buruzko atalean.

3.3.2 AFEDen definizioa

Atal honetan AFEDen definizio formala emango da. Gainera, AFEDen funtzionalitatea eta beste nozio garrantzitsu batzuk ere aurkeztuko dira. Besteak beste, *konfigurazio ez-determinista*, AFED batek egiten duen *konputazioa* eta AFED bati dagokion *lengoaia*. Adibide batzuk ere emango dira.

3.3.2.1 AFEDen definizio formala

Automata finitu ez-determinista bat boskote bat da (bost osagai dituen egitura bat da):

$$(Q, A, \nu, q_0, Y)$$

Bost osagai horien definizioa honako hau da:

- Q egoeren multzo finitua da: $q_0, q_1, q_2, q_3, \ldots$
- A alfabetoa da
- ν trantsizio-funtzio ez-determinista da (ν sinboloa *nu* letra grekoa da). Q multzoko egoera bat eta A multzoko sinbolo bat emanda, egoera horretan gaudenean sinbolo hori irakurtzen badugu zein egoeretara joango garen adierazten du ν trantsizio-funtzioak. ν funtzioaren mota honako hau da:

  $$\nu : Q \times A \to 2^Q$$

  Hor, 2^Q multzoa $Q \to 2$ erako funtzio denez osatutako multzoa da eta $2 = \{0, 1\}$ da, 2 multzo horretan 0 balioak *False* eta 1 balioak *True* adierazten dutela ulertuz. Beste era batera esanda, 2^Q multzoa Q-ren azpimultzo guztiez osatutako multzoa da.
- q_0 hasierako egoera da. Hasierako egoera beti bakarra izango da.
- Y onarpen egoeren multzoa da. Y multzoa Q multzoaren azpimultzoa izango da, hau da, $Y \subseteq Q$ edo, beste era batera esanda, $Y \in 2^Q$. Hitz bat irakurritakoan AFEDa Y multzoko egoera batean gelditzen bada, orduan AFEDak "Bai" erantzungo du.

AFD eta AFEDen arteko desberdintasun nagusiak honako hiru hauek dira:

1. AFDetan, q_j egoeran egonda α sinbolo bat irakurtzen badugu, q_h beste egoera batera igaroko gara. AFED batean aldiz, q_j egoeran α sinbolo bat irakurtzen badugu, egoera-multzo batera igaroko gara. Multzo hori hutsa, elementu bakarrekoa edo elementu bat baino gehiagokoa izan daiteke.

2. AFDetan hitzeko sinbolo denak irakurri behar dira, hitz hutsa geratu arte. Gainera beti bukatu ahal izango dugu hitzen irakurketa. AFEDetan, q_j egoera batean egonda α sinbolo bat irakurtzean gerta daiteke inora joaterik ez edukitzea. Kasu horretan, oraindik hitza irakurtzea bukatu ez bada ere, automata gelditu egingo da eta "Ez" erantzungo du, Y multzoko egoera batean al dagoen ala ez kontuan hartu gabe.

3. AFDetan hitzaren irakurketa beti burutu ahal izango denez, hitza bukatutakoan Y multzoko egoera batean baldin bagaude “Bai” erantzungo da. AFED-etan, “Bai” erantzuteko hitz osoa bukatu beharko da eta gainera Y multzokoa den egoera batean bukatu beharko da. Beraz, AFED batean “Ez” erantzuteko bi aukera daude: hitza osorik irakurri ondoren Y-koa den egoera batean ez aurkitzea edo hitza irakurri ezinda gelditzea. Hitza irakurri ezinda gelditzen bagara, berdin da azkeneko egoera Y-koa al den ala ez.

AFED baten **funtzionamendua** honako hau da: hasieran AFEDa q_0 egoeran egongo da eta sarrera bezala A^* multzoko hitz bat jasoko du. Hitz hori osatzen duten sinboloak banan-banan irakurriko ditu. Sinbolo bakoitza irakurtzean zein egoeretara igaro behar duen erabakiko da ν trantsizio-funtzioa erabiliz. Beraz, q_i egoera batean gaudela A alfabetokoa den α sinboloa irakurtzen badugu eta $\nu(q_i, \alpha) = \{q_{j_1}, q_{j_2}, \ldots, q_{j_n}\}$ baldin bada, $\{q_{j_1}, q_{j_2}, \ldots, q_{j_n}\}$ egoeretara igaroko gara. Hor $\nu(q_i, \alpha) = \varnothing$ baldin bada, ezin da aurrera jarraitu eta “Ez” erantzungo da. Hitzak beti finituak izango direnez, hitz bat irakurtzeko prozesua ere finitua izango da. Gainera, lehenago esan den bezala, hitza irakurri ezinda ere geldi gaitezke. Kasu horretan “Ez” erantzungo da. Hitza irakurtzea bukatutakoan 2^Q multzokoa den S egoera-multzo batean egongo gara (beraz, $S \in 2^Q$). Kontuan izan $S \subseteq Q$ ere beteko dela. S multzoan Y multzokoa den egoeraren bat baldin badago, orduan automatak “Bai” erantzungo du, hitza automatari lotutako lengoaiakoa dela adieraziz. Baina hitza bukatzean S multzoan Y multzokoa den egoerarik ez badago, automatak “Ez” erantzungo du, hitza automatari lotutako lengoaiakoa ez dela adieraziz. Beste era batera esanda, $S \cap Y \neq \varnothing$ baldin bada, “Bai” erantzungo du eta $S \cap Y = \varnothing$ baldin bada, “Ez” erantzungo du.

S multzoa 2^Q multzoko elementu bat izanda, hau da, S multzoa Q-ren azpimultzo bat izanda eta w elementua A^* multzoko hitz bat izanda, (S, w) bikoteari **konfigurazio ez-determinista** deituko diogu. Konfigurazio ez-deterministak $2^Q \times A^*$ motakoak dira. (S, w) erako konfigurazio ez-determinista baten bidez S multzoko egoeretan gaudela eta oraindik w hitza irakurtzeko daukagula adierazten da. Adibidez, A alfabetoa $\{a, b, c\}$ baldin bada, oraingo konfigurazio ez-determinista $(\{q_1, q_4\}, caaba)$ baldin bada eta ν trantsizio-funtzioak $\nu(q_1, c) = \{q_2, q_6\}$ eta $\nu(q_4, c) = \{q_3, q_5, q_6\}$ betetzen dela adierazten badu, $caaba$ hitzeko lehenengo sinboloa irakurritakoan $\{q_2, q_3, q_5, q_6\}$ egoera-multzora igaroko gara eta konfigurazio ez-determinista berria $(\{q_2, q_3, q_5, q_6\}, aaba)$ izango da. Konfigurazio ez-determinista berrian ikus daitekeen bezala, irakurtzeko gelditzen den hitza laburragoa da orain (irakurri den sinboloa desagertu egin baita).

Konfigurazio deterministak $Q \times A^*$ motakoak direnez eta konfigurazio ez-deterministak $2^Q \times A^*$ motakoak direnez, konfigurazio deterministak eta ez-deterministak bereiztea erraza da. Konfigurazio deterministak (q_j, w) erakoak izango dira eta konfigurazio ez-deterministak $(\{q_{j_1}, q_{j_2}, \ldots, q_{j_n}\}, w)$ erakoak izango dira. Egitura aztertuz bi konfigurazio motak bereiztea erraza denez, *konfigurazio determinista* edo *konfigurazio ez-determinista* idatzi beharrean *konfigurazioa* idatziko dugu automata finituei buruzko gai honetan zehar. Hala ere, konfigurazio bat determinista edo ez-determinista dela azpimarratu nahi dugunean, ezaugarri hori zehaztuko dugu.

AFED batek egindako **konputazioa** edo kalkulu-prozesua ν trantsizio-funtzioan oinarritzen den ν^* funtzioaren bidez definitzen da. ν^* funtzioaren mota honako hau da:

$$\nu^* : 2^Q \times A^* \rightarrow 2^Q$$

Beraz, S egoera-multzo bat eta w hitz bat emanda, (hau da, konfigurazio ez-determinista bat emanda), ν^* funtzioak egoera-multzo bat itzuliko du (S multzoko egoeretatik abiatuz eta, ahal bada, w hitzeko sinbolo denak irakurriz, zein egoeretara iritsiko garen adieraziz).

ν^* funtzioaren definizio formala honako hau da:

- $\nu^*(S, \varepsilon) = S$
- $\nu^*(\varnothing, w) = \varnothing$
- $\nu^*(S, \alpha w) = \nu^*(\bigcup_{q_j \in S} \nu(q_j, \alpha), w)$

Definizio horren bidez honako hau adierazten da: irakurtzeko gelditzen den hitza hutsa baldin bada, hau da, ε, orduan S egoera-multzo berean jarraituko dugu. Irakurtzeko gelditzen den hitza hutsa ez bada, eta S multzo hutsa baldin bada, hau da, $\varnothing$ baldin bada, ez gaude inon eta beraz ezin dugu inora joan eta hori multzo hutsean egongo garela esanez adierazten da. S multzoa ere hutsa ez bada eta irakurtzeko gelditzen den hitza hutsa ez bada, eta hitz horretako lehenengo sinboloa A alfabetokoa den α sinboloa baldin bada, ν trantsizio-funtzioa aplikatuko zaio S-ko egoera bakoitzari eta α sinboloari eta S-ko egoeretatik α-ren bidez irisgarriak diren egoeretara igaroko gara. Egoera-multzo berrian kokatu ondoren, w hitzeko gainontzeko sinboloak irakurriz jarraitu beharko da era berean:

Esate baterako, demagun S multzoa $\{q_2, q_4, q_5, q_7\}$ dela. ν^*-ren bidezko urrats bat honela planteatuko genuke:

$$\nu^*(\{q_2, q_4, q_5, q_7\}, \alpha w) = \nu^*(\nu(q_2, \alpha) \cup \nu(q_4, \alpha) \cup \nu(q_5, \alpha) \cup \nu(q_7, \alpha), w)$$

Hor $\nu(q_2, \alpha)$-ren bidez q_2-tik α sinboloaren bidez irisgarriak diren egoeren multzoa adierazten da, $\nu(q_4, \alpha)$-ren bidez q_4-tik α sinboloaren bidez irisgarriak diren egoeren multzoa adierazten da eta abar.

Beste adibide baten laguntzaz, ν^*-ren funtzionamendua hobeto zehaztuko dugu. Demagun alfabetoa $A = \{a, b, c\}$ dela, oraingo konfigurazioa $(\{q_1, q_4\}, caaba)$ dela eta ν honela definituta dagoela (hor agertzen diren kasuetarako):

- $\nu(q_1, c) = \{q_2, q_6\}$
- $\nu(q_2, a) = \varnothing$
- $\nu(q_3, a) = \{q_7\}$
- $\nu(q_4, c) = \{q_3, q_5, q_6\}$
- $\nu(q_5, a) = \varnothing$
- $\nu(q_5, b) = \{q_6\}$
- $\nu(q_6, a) = \{q_5, q_6\}$

- $\nu(q_6, b) = \{q_2, q_4\}$
- $\nu(q_7, a) = \{q_7\}$
- $\nu(q_7, b) = \{q_1, q_2\}$

$\nu^*(q_3, caab)$ konputazioa urratsez urrats honela garatuko litzateke:

- $caab$ hitza hutsa ez denez, $\nu^*(\{q_1, q_4\}, caab) = \nu^*(\nu(q_1, c) \cup \nu(q_4, c), aab)$.
- $\nu(q_1, c) = \{q_2, q_6\}$ eta $\nu(q_4, c) = \{q_3, q_5, q_6\}$ direnez, $\nu^*(\nu(q_1, c) \cup \nu(q_4, c), aab)$ espresioa $\nu^*(\{q_2, q_3, q_5, q_6\}, aab)$ bezala geratuko da.
- aab hitza hutsa ez denez, $\nu^*(\{q_2, q_3, q_5, q_6\}, aab) = \nu^*(\nu(q_2, a) \cup \nu(q_3, a) \cup \nu(q_5, a) \cup \nu(q_6, a), ab)$.
- $\nu(q_2, a) = \varnothing$, $\nu(q_3, a) = \{q_7\}$, $\nu(q_5, a) = \varnothing$ eta $\nu(q_6, a) = \{q_5, q_6\}$ direla kontuan hartuz, $\nu^*(\nu(q_2, a) \cup \nu(q_3, a) \cup \nu(q_5, a) \cup \nu(q_6, a), ab)$ espresiotik $\nu^*(\{q_5, q_6, q_7\}, ab)$ lortuko da.
- ab hitza hutsa ez denez, $\nu^*(\{q_5, q_6, q_7\}, ab) = \nu^*(\nu(q_5, a) \cup \nu(q_6, a) \cup \nu(q_7, a), b)$.
- $\nu(q_5, a) = \varnothing$, $\nu(q_6, a) = \{q_5, q_6\}$ eta $\nu(q_7, a) = \{q_7\}$ betetzen denez, $\nu^*(\nu(q_5, a) \cup \nu(q_6, a) \cup \nu(q_7, a), b)$ espresiotik $\nu^*(\{q_5, q_6, q_7\}, b)$ espresioa lortuko da.
- b hitza hutsa ez denez, $\nu^*(\{q_5, q_6, q_7\}, b) = \nu^*(\nu(q_5, b) \cup \nu(q_6, b) \cup \nu(q_7, b), \varepsilon)$.
- $\nu(q_5, b) = \{q_6\}$, $\nu(q_6, b) = \{q_2, q_4\}$ eta $\nu(q_7, b) = \{q_1, q_2\}$ izateagatik, $\nu^*(\nu(q_5, b) \cup \nu(q_6, b) \cup \nu(q_7, b), \varepsilon)$ espresiotik $\nu^*(\{q_1, q_2, q_4, q_6\}, \varepsilon)$ espresioa lortuko da.
- ε hitz hutsa geratzen denez, $\nu^*(\{q_1, q_2, q_4, q_6\}, \varepsilon)$ espresioaren balioa $\{q_1, q_2, q_4, q_6\}$ da.

Beraz, $\nu^*(\{q_1, q_4\}, caab) = \{q_1, q_2, q_4, q_6\}$.

Konputazio hori konfigurazio ez-deterministez eratutako honako sekuentzia honen bidez adieraz daiteke:

$$
\begin{array}{c}
(\{q_1, q_4\}, caab) \\
| \\
(\{q_2, q_3, q_5, q_6\}, aab) \\
| \\
(\{q_5, q_6, q_7\}, ab) \\
| \\
(\{q_5, q_6, q_7\}, b) \\
| \\
(\{q_1, q_2, q_4, q_6\}, \varepsilon)
\end{array}
$$

Egoera batean gaudenean sinbolo batentzat gezirik ez, gezi bakarra edo gezi bat baino gehiago egon daitezkeenez, automata hauek **ez-deterministak** direla esaten da.

$N = (Q, A, \nu, q_0, Y)$ **AFED bati lotutako** $L(N)$ **lengoaia** N automatarengandik "Bai" erantzuna jasotzen duten hitzez osatutako lengoaia da. Beste era batera esanda, q_0-tik abiatuta osorik irakurri daitezkeen eta Y multzokoa den egoeraren bat baduen egoera-multzo batean lagatzen gaituzten A^*-ko hitzez osatutako lengoaia da. Era formalean definizioa honako hau izango litzateke:

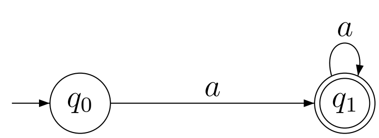
$$L(N) = \{w \mid w \in A^* \wedge (\nu^*(\{q_0\}, w) \cap Y) \neq \varnothing\}$$

3.3.2.2 AFEDen adibideak

3.3.2.2.1 Hutsak ez diren eta bakarrik a sinboloaren errepikapenez eratuta dauden hitzez osatutako lengoaiari dagokion AFED-a

Adibide honetako AFEDak (3.3.14 irudia begiratu) a, b eta c sinboloak eduki ditzakeen hitz bat jasoko du sarrera bezala. AFED horri hitz bat emandakoan, hitz hori hutsa ez den eta bakarrik a sinboloaren errepikapenez eratuta dagoen hitza al den erabakiko du. Horrela bada, "Bai" erantzuna itzuliko du eta bestela "Ez" erantzuna itzuliko du. Lengoaiaren definizio formala honako hau da:

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



3.3.14 irudia. Hutsak ez diren eta bakarrik a sinboloaren errepikapenez eratuta dauden hitzei dagokien AFEDa.

AFEDa bi egoerez osatuta dago. q_0 hasierako egoera da. w hitz bat edo karaktere-kate bat irakurtzen hasten garenean q_0 egoeran gaude beraz. Lehenengo karakterea (edo sinboloa) a bada, AFEDa q_1 egoerara igaroko da, beraz, $\nu(q_0, a) = \{q_1\}$. Lehenengo karakterea b edo c bada, ezin da inora joan eta ondorioz $\nu(q_0, b) = \nu(q_0, c) = \varnothing$. AFEDa q_1 egoeran dagoenean, a sinboloa etortzen bada, AFEDa egoera berean mantenduko da, hau da, q_1 egoeran, hau da, $\nu(q_1, a) = \{q_1\}$. Bestalde, q_1 egoeran egonda, b edo c etortzen bada, ezingo da inora joan, $\nu(q_1, b) = \nu(q_1, c) = \varnothing$. w karaktere-kate osoa irakurtzea lortzen bada eta gainera q_1 egoeran bukatzen bada, horrek esan nahiko du karaktere-katea ez dela hutsa eta a sinboloaren errepikapenez osatuta dagoela. Kasu horretan $\nu^*(\{q_0\}, w) = \{q_1\}$ beteko da eta ondorioz $\nu^*(\{q_0\}, w) \cap Y \neq \varnothing$ izango da —izan ere, $Y = \{q_1\}$ da eta $\{q_1\} \cap \{q_1\} \neq \varnothing$ izango da— eta erantzuna “Bai” izango da. Hasierako w hitza ε baldin bada, AFEDak q_0 egoeran bukatuko du, hau da, $\nu^*(\{q_0\}, \varepsilon) = \{q_0\}$. Eta $\{q_0\} \cap Y = \varnothing$ denez, hau da, $\{q_0\} \cap \{q_1\} = \varnothing$ denez, AFEDak “Ez” erantzungo du. Aldiz, AFEDak ez badu lortzen w karaktere-kate osoa irakurtzea, inora joan ezinda geldituko da. Beraz kasu horretan $\nu^*(\{q_0\}, w) = \varnothing$ izango da eta $\varnothing \cap Y = \varnothing$ denez, hau da, $\varnothing \cap \{q_1\} = \varnothing$ denez, AFED-ak “Ez” erantzungo du.

Jarraian AFEDaren osagaiak definituko dira boskote eran:

$$(Q, A, \nu, q_0, Y)$$

Osagai bakoitza honela geratuko da:

- $Q = \{q_0, q_1\}$
- $A = \{a, b, c\}$
- $\nu : Q \times A \to 2^Q$ honako taula honen bidez definitutako trantsizio-funtzioa da:

ν	a	b	c
q_0	$\{q_1\}$	$\varnothing$	$\varnothing$
q_1	$\{q_1\}$	$\varnothing$	$\varnothing$

  Taula horren bidez $\nu(q_0, a)$-ren balioa $\{q_1\}$ dela, $\nu(q_0, b)$-ren balioa $\varnothing$ dela eta abar adierazten da.

- q_0 hasierako egoera da. Hasierako egoera beti bakarra izango da.
- $Y = \{q_1\}$.

Orain $(\{q_0\}, aaa)$ konfigurazioari dagokion konputazioa garatuko dugu urratsez urrats:

1. urratsa: $\nu^*(\{q_0\}, aaa) = \nu^*(\nu(q_0, a), aa) = \nu^*(\{q_1\}, aa)$

2. urratsa: $\nu^*(\{q_1\}, aa) = \nu^*(\nu(q_1, a), a) = \nu^*(\{q_1\}, a)$

3. urratsa: $\nu^*(\{q_1\}, a) = \nu^*(\{q_1\}, a\varepsilon) = \nu^*(\nu(q_1, a), \varepsilon) = \nu^*(\{q_1\}, \varepsilon)$

4. urratsa: $\nu^*(\{q_1\}, \varepsilon) = \{q_1\}$

Konputazioa $\{q_1\}$ multzoan bukatzen denez eta multzo horretan gutxienez elementu bat, q_1 elementua, Y multzokoa denez, $\{q_1\} \cap Y \neq \varnothing$ betetzen da eta, ondorioz, hitz hori adibide honetako lengoaiakoa da (ez da hutsa eta a sinboloaren errepikapenez eratutako dago).

Konputazio hau konfigurazio ez-deterministez osatutako sekuentzia bezala honela adieraz daiteke:

$$\begin{array}{c} (\{q_0\}, aaa) \\ | \\ (\{q_1\}, aa) \\ | \\ (\{q_1\}, a) \\ | \\ (\{q_1\}, \varepsilon) \end{array}$$

Jarraian $(\{q_0\}, abca)$ konfigurazioari dagokion konputazioa garatuko dugu urratsez urrats:

1. urratsa: $\nu^*(\{q_0\}, abca) = \nu^*(\nu(q_0, a), bca) = \nu^*(\{q_1\}, bca)$

2. urratsa: $\nu^*(\{q_1\}, bca) = \nu^*(\nu(q_2, b), ca) = \nu^*(\varnothing, ca)$

3. urratsa: $\nu^*(\varnothing, ca) = \varnothing$

Bukaerako egoera-multzoa hutsa denez ($\varnothing$) eta $\varnothing$ multzoan Y multzoko elementurik ez dagoenez, hitza ez da adibide honetako lengoaiakoa.

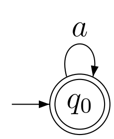
Konputazio hori grafikoki honela adieraz daiteke:

$$\begin{array}{c} (\{q_0\}, abca) \\ | \\ (\{q_1\}, bca) \\ | \\ (\varnothing, bca) \end{array}$$

3.3.2.2.2 b-rik eta c-rik ez duten hitzez osatutako lengoaiari dagokion AFEDa

Adibide honetako AFEDak (3.3.15 irudian begiratu) a, b eta c sinboloak izan ditzakeen hitz bat jasoko du sarrera bezala. Emandako hitza b-rik eta c-rik ez duen hitz bat al den erabaki beharko du AFEDak. Horrela, b-rik eta c-rik ez badu “Bai” erantzun beharko du eta bestela “Ez” erantzun beharko du. Lengoaiaren definizio formala honako hau da:

$$\{w \mid w \in A^* \land |w|_a = |w|\}$$



3.3.15 irudia. b-rik eta c-rik ez duten hitzez osatutako lengoaiari dagokion AFED bat.

Adibide honetako AFEDak egoera bakarra du: q_0. Egoera bakar hori izango da hasierako egoera eta, ondorioz, hitz bat irakurtzen hasterakoan q_0 egoeran egongo da. Lehenengo karakterea a baldin bada, AFEDa q_0 egoeran geratuko da. Lehenengo karakterea b edo c baldin bada, AFEDak ezingo du inora joan eta ezta q_0-n geratu ere. q_0 egoeran mantenduz hitz osoa irakurtzea lortzen bada, badakigu karaktere-kateak ez duela b-rik eta c-rik eta “Bai” erantzuna itzuliko da. Bestalde, karaktere-katea irakurtzea ez bada lortzen, badakigu karaktere-kateak baduela b-ren bat edo c-ren bat eta “Ez” erantzuna itzuliko da.

Jarraian AFEDa boskote bezala definituko da.

$$(Q, A, \nu, q_0, Y)$$

Bost osagaiak honako hauek dira:

- $Q = \{q_0\}$
- $A = \{a, b, c\}$
- $\nu : Q \times A \rightarrow 2^Q$ honako taula honen bidez definitutako trantsizio-funtzioa da:

ν	a	b	c
q_0	$\{q_0\}$	$\varnothing$	$\varnothing$

  Taularen bidez $\nu(q_0, a)$-ren balioa $\{q_0\}$ dela, $\nu(q_0, b)$-ren balioa $\varnothing$ dela eta abar adierazten da.

- q_0 hasierako egoera da. Hasierako egoera beti bakarra da.
- $Y = \{q_0\}$.

Orain $(\{q_0\}, aaa)$ konfigurazioari dagokion konputazioa garatuko dugu urratsez urrats:

1. urratsa: $\nu^*(\{q_0\}, aaa) = \nu^*(\nu(q_0, a), aa) = \nu^*(\{q_0\}, aa)$

2. urratsa: $\nu^*(\{q_0\}, aa) = \nu^*(\nu(q_0, a), a) = \nu^*(\{q_0\}, a)$

3. urratsa: $\nu^*(\{q_0\}, a) = \nu^*(\{q_0\}, a\varepsilon) = \nu^*(\nu(q_0, a), \varepsilon) = \nu^*(\{q_0\}, \varepsilon)$

4. urratsa: $\nu^*(\{q_0\}, \varepsilon) = \{q_0\}$

Konputazioa $\{q_0\}$ multzoan bukatzen denez eta multzo horretan gutxienez elementu bat, q_0 elementua, Y multzokoa denez, $\{q_0\} \cap Y \neq \varnothing$ betetzen da eta ondorioz hitz hori adibide honetako lengoaiakoa da (ez du ez b-rik eta ez c-rik).

Konputazio hori grafikoki honako konfigurazio ez-deterministen zerrenda bezala adieraz daiteke:

$$\begin{array}{c}(\{q_0\}, aaa)\\ |\\ (\{q_0\}, aa)\\ |\\ (\{q_0\}, a)\\ |\\ (\{q_0\}, \varepsilon)\end{array}$$

Jarraian $(\{q_0\}, abca)$ konfigurazio ez-deterministari dagokion konputazioa garatuko dugu urratsez urrats:

1. urratsa: $\nu^*(\{q_0\}, abca) = \nu^*(\nu(q_0, a), bca) = \nu^*(\{q_0\}, bca)$

2. urratsa: $\nu^*(\{q_0\}, bca) = \nu^*(\nu(q_0, b), ca) = \nu^*(\varnothing, ca)$

3. urratsa: $\nu^*(\varnothing, ca) = \varnothing$

Konputazioa multzo hutsean ($\varnothing$) bukatzen denez eta multzo hutsean Y multzokoa den elementurik ez dagoenez, $\varnothing \cap Y = \varnothing$ betetzen da eta ondorioz hitz hori ez da adibide honetako lengoaiakoa (badu b-ren bat edo c-ren bat).

Konputazio hori grafikoki honako konfigurazio ez-deterministen zerrenda bezala adieraz daiteke:

$$\begin{array}{c}(\{q_0\}, abca)\\ |\\ (\{q_0\}, bca)\\ |\\ (\varnothing, ca)\end{array}$$

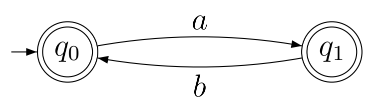
3.3.2.2.3 c-rik ez duten eta a-ren eta b-ren agerpenak orden horretan tartekatuta dituzten hitzez osatutako lengoaiari dagokion AFEDa

Adibide honetako AFEDak (3.3.16 irudia begiratu) c-rik ez duten eta a eta b orden horretan tartetakuta dituzten $A = \{a, b, c\}$ alfabetoaren gainean definitutako hitzentzat “Bai” erantzungo du. Beraz, AFED honi dagokion lengoaiako hitzetan ezin dira bi a jarraian agertu eta ezta bi b ere. Hitz hutsa lengoaiakoa da eta hutsak ez diren hitzen kasuan lehenengo sinboloak a izan behar du. Hala ere, azkeneko sinboloa a edo b izan daiteke. Beraz, ε, $abab$ eta aba hitzentzat AFEDak “Bai” erantzungo luke, baina $abca$, $bababa$, aaa, cc eta aab hitzentzat “Ez” erantzungo luke. Lengoaiaren definizio formala honako hau da:

$$\{w \mid w \in A^* \wedge \exists k(k \in I\!N \wedge (w = (ab)^k \vee w = ((ab)^k)a))\}$$

Lengoaia hori beste era honetara ere defini daiteke:

$$\begin{array}{ll}\{w \mid & w \in A^* \wedge \forall k((k \in I\!N \wedge 1 \leq k \leq |w| \wedge k \bmod 2 = 1) \rightarrow w(k) = a)\\ & \wedge \forall k((k \in I\!N \wedge 1 \leq k \leq |w| \wedge k \bmod 2 = 0) \rightarrow w(k) = b)\}\end{array}$$



3.3.16 irudia. c-rik ez eta a eta b, orden horretan, tartekatuta dituzten hitzen lengoaiari dagokion AFED bat.

AFEDak bi egoera ditu: q_0 eta q_1. Hasierako egoera q_0 da eta hitz osoa irakurtzea lortzen bada, bai q_0 egoeran bukatzen bada eta bai q_1 egoeran bukatzen bada, AFEDaren erantzuna "Bai" izango da.

Aibide honetako AFEDa boskote bezala honela adieraz daiteke:

$$(Q, A, \nu, q_0, Y)$$

Boskote horretako osagaien definizioa honako hau da:

- $Q = \{q_0, q_1\}$
- $A = \{a, b, c\}$
- $\nu : Q \times A \to 2^Q$ honako taula honen bidez definitutako trantsizio funtzioa da:

ν	a	b	c
q_0	$\{q_1\}$	$\varnothing$	$\varnothing$
q_1	$\{q_0\}$	$\varnothing$	$\varnothing$

  Taula horren bidez $\delta(q_0, a)$-ren balioa $\{q_1\}$ dela, $\delta(q_0, b)$-ren balioa $\varnothing$ dela eta abar adierazten da.

- q_0 hasierako egoera da. Hasierako egoera beti bakarra da.
- $Y = \{q_0, q_1\}$. Adibide honetan hasierako egoera Y multzoan ager daitekeela ikus dezakegu.

Orain $(\{q_0\}, abab)$ konfigurazio ez-deterministari dagokion konputazioa garatuko dugu urratsez urrats:

1. urratsa: $\nu^*(\{q_0\}, abab) = \nu^*(\nu(q_0, a), bab) = \nu^*(\{q_1\}, bab)$

2. urratsa: $\nu^*(\{q_1\}, bab) = \nu^*(\nu(q_1, b), ab) = \nu^*(\{q_0\}, ab)$

3. urratsa: $\nu^*(\{q_0\}, ab) = \nu^*(\nu(q_0, a), b) = \nu^*(\{q_1\}, b)$

4. urratsa: $\nu^*(\{q_1\}, b) = \nu^*(\nu(q_1, b), \varepsilon) = \nu^*(\{q_0\}, \varepsilon)$

5. urratsa: $\nu^*(\{q_0\}, \varepsilon) = \{q_0\}$

Konputazioa $\{q_0\}$ multzoan bukatzen denez eta multzo horretan gutxienez elementu bat, q_0 elementua, Y multzokoa denez, $\{q_0\} \cap Y \neq \varnothing$ betetzen da eta, ondorioz, hitz hori adibide honetako lengoaiakoa da.

Konputazio hori grafikoki honako konfigurazio ez-deterministen zerrenda bezala adieraz daiteke:

$$\begin{array}{c} (\{q_0\}, abab) \\ | \\ (\{q_1\}, bab) \\ | \\ (\{q_0\}, ab) \\ | \\ (\{q_1\}, b) \\ | \\ (\{q_0\}, \varepsilon) \end{array}$$

Jarraian $(\{q_0\}, abc)$ konfigurazioari dagokion konputazioa garatuko dugu urratsez urrats:

1. urratsa: $\nu^*(\{q_0\}, abc) = \nu^*(\nu(q_0, a), bc) = \nu^*(\{q_1\}, bc)$

2. urratsa: $\nu^*(\{q_1\}, bc) = \nu^*(\nu(q_1, b), c) = \nu^*(\{q_0\}, c)$

3. urratsa: $\nu^*(\{q_0\}, c) = \nu^*(\nu(q_0, c), \varepsilon) = \nu^*(\varnothing, \varepsilon)$

4. urratsa: $\nu^*(\varnothing, \varepsilon) = \varnothing$

Konputazioa multzo hutsean ($\varnothing$) bukatzen denez eta multzo hutsean Y multzokoa den elementurik ez dagoenez, $\varnothing \cap Y \neq \varnothing$ betetzen da eta, ondorioz, hitz hori ez da adibide honetako lengoaiakoa.

Konputazio hori grafikoki honako konfigurazio ez-deterministen zerrenda bezala adieraz daiteke:

$$\begin{array}{c} (\{q_0\}, abc) \\ | \\ (\{q_1\}, bc) \\ | \\ (\{q_0\}, c) \\ | \\ (\varnothing, \varepsilon) \end{array}$$

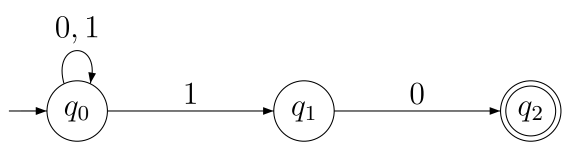
3.3.2.2.4 10 katearekin bukatzen diren hitzez osatutako lengoaiari dagokion AFEDa

Adibide honetako AFEDak (3.3.17 irudia begiratu) 10 katearekin bukatzen diren $A = \{0, 1\}$ alfabetoaren gainean definitutako hitzentzat “Bai” erantzungo du. Beraz, esate baterako 0011010 eta 0010 hitzen kasuan “Bai” erantzungo du, baina 000, 1101, ε eta 0 hitzen kasuan “Ez” erantzungo du. Lengoaiaren definizio formala honako hau da:

$$\{w \mid w \in A^* \wedge \exists u(u \in A^* \wedge w = u10)\}$$

Lengoaia hori beste era honetara ere defini daiteke:

$$\{w \mid w \in A^* \wedge |w| \geq 2 \wedge w(|w| - 1) = 1 \wedge w(|w|) = 0\}$$



3.3.17 irudia. 10 azpikatearekin bukatzen diren hitzen lengoaiari dagokion AFED bat.

3.3.17 irudian erakusten den AFED horrek hiru egoera ditu: q_0, q_1 eta q_2. Hasierako egoera q_0 da eta hitz osoa irakurtzea lortzen bada eta gainera q_2 egoeran bukatzen bada, erantzuna “Bai” izango da.

AFED hori honako boskote hau bezala adieraz daiteke:

$$(Q, A, \nu, q_0, Y)$$

Boskote horretako osagaiak honela definituko genituzke:

- $Q = \{q_0, q_1, q_2\}$
- $A = \{0, 1\}$
- $\nu : Q \times A \to 2^Q$ honako taula honen bidez definitutako trantsizio-funtzioa da:

ν	0	1
q_0	$\{q_0\}$	$\{q_0, q_1\}$
q_1	$\{q_2\}$	$\varnothing$
q_2	$\varnothing$	$\varnothing$

Taula horren bidez $\nu(q_0, 0)$-ren balioa $\{q_0\}$ dela, $\nu(q_0, 1)$-en balioa $\{q_0, q_1\}$ dela eta abar adierazten da.

- q_0 hasierako egoera da. Hasierako egoera beti bakarra da.
- $Y = \{q_2\}$.

Hona hemen $(\{q_0\}, 0010)$ konfigurazio ez-deterministari dagokion urratsez urrats garatutako konputazioa:

1. urratsa: $\nu^*(\{q_0\}, 0010) = \nu^*(\nu(q_0, 0), 010) = \nu^*(\{q_0\}, 010)$

2. urratsa: $\nu^*(\{q_0\}, 010) = \nu^*(\nu(q_0, 0), 10) = \nu^*(\{q_0\}, 10)$

3. urratsa: $\nu^*(\{q_0\}, 10) = \nu^*(\nu(q_0, 1), 0) = \nu^*(\{q_0, q_1\}, 0)$

4. urratsa: $\nu^*(\{q_0, q_1\}, 0) = \nu^*(\nu(q_0, 0) \cup \nu(q_1, 0), \varepsilon) = \nu^*(\{q_0\} \cup \{q_2\}, \varepsilon) = \nu^*(\{q_0, q_2\}, \varepsilon)$

5. urratsa: $\nu^*(\{q_0, q_2\}, \varepsilon) = \{q_0, q_2\}$

Konputazioa $\{q_0, q_2\}$ multzoan bukatzen denez eta multzo horretan gutxienez elementu bat, q_2 elementua, Y multzokoa denez, $\{q_0, q_2\} \cap Y \neq \varnothing$ betetzen da eta ondorioz hitz hori adibide honetako lengoaiakoa da.

Konputazio hori grafikoki honako konfigurazio ez-deterministen zerrenda bezala adieraz daiteke:

$$
\begin{array}{c}
(\{q_0\}, 0010) \\
| \\
(\{q_0\}, 010) \\
| \\
(\{q_0\}, 10) \\
| \\
(\{q_0, q_1\}, 0) \\
| \\
(\{q_0, q_2\}, \varepsilon)
\end{array}
$$

Konfigurazio deterministez eratutako zuhaitz bezala ere adieraz daiteke konputazio hori:

$$
\begin{array}{ccc}
 & (q_0, 0010) & \\
 & | & \\
 & (q_0, 010) & \\
 & | & \\
 & (q_0, 10) & \\
 \swarrow & & \searrow \\
(q_0, 0) & & (q_1, 0) \\
| & & | \\
(q_0, \varepsilon) & & (q_2, \varepsilon) \\
\text{Ez} & & \text{Bai}
\end{array}
$$

Gutxienez adarretako bat Y multzokoa den egoera batez eta ε hitz hutsaz eratutako konfigurazio determinista batean bukatzen denez, 0010 hitza adibide honetako lengoaiakoa da.

Orain, $(\{q_0\}, 111)$ konfigurazioari dagokion konputazioa urratsez urrats garatutako dugu:

1. urratsa: $\nu^*(\{q_0\}, 111) = \nu^*(\nu(q_0, 1), 11) = \nu^*(\{q_0, q_1\}, 11)$

2. urratsa: $\nu^*(\{q_0, q_1\}, 11) = \nu^*(\nu(q_0, 1) \cup \nu(q_1, 1), 1) = \nu^*(\{q_0, q_1\} \cup \varnothing, 1) = \nu^*(\{q_0, q_1\}, 1)$

3. urratsa: $\nu^*(\{q_0, q_1\}, 1) = \nu^*(\nu(q_0, 1) \cup \nu(q_1, 1), \varepsilon) = \nu^*(\{q_0, q_1\} \cup \varnothing, \varepsilon) = \nu^*(\{q_0, q_1\}, \varepsilon)$

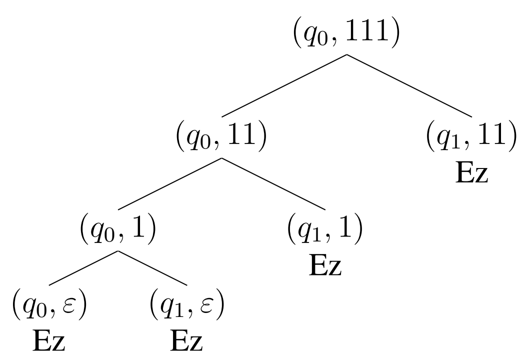
4. urratsa: $\nu^*(\{q_0, q_1\}, \varepsilon) = \{q_0, q_1\}$

Konputazioa $\{q_0, q_1\}$ multzoan bukatzen denez eta multzo horretan Y multzokoa den elementurik ez dagoenez, $\{q_0, q_1\} \cap Y = \varnothing$ betetzen da eta, ondorioz, hitz hori adibide honetako lengoaiakoa ez da.

Konputazio hori honako konfigurazio ez-deterministen zerrenda bezala ipini daiteke:

$$\begin{array}{c} (\{q_0\}, 111) \\ | \\ (\{q_0, q_1\}, 11) \\ | \\ (\{q_0, q_1\}, 1) \\ | \\ (\{q_0, q_1\}, \varepsilon) \end{array}$$

Konfigurazio deterministez eratutako zuhaitz bezala adieraziz honela geratuko litzateke konputazio hori:



Y multzokoa den egoera batez eta ε hitz hutsaz eratutako konfigurazio determinista batean bukatzen den adarrik ez dagoenez, 111 hitza adibide honetako lengoaiakoa ez da.

Bukatzeko, $(\{q_0\}, 11010)$ konfigurazioari dagokion konputazioa urratsez urrats garatutako dugu:

1. urratsa: $\nu^*(\{q_0\}, 11010) = \nu^*(\nu(q_0, 1), 1010) = \nu^*(\{q_0, q_1\}, 1010)$

2. urratsa: $\nu^*(\{q_0,q_1\},1010) = \nu^*(\nu(q_0,1) \cup \nu(q_1,1), 010) = \nu^*(\{q_0,q_1\} \cup \varnothing, 1) = \nu^*(\{q_0,q_1\}, 010)$

3. urratsa: $\nu^*(\{q_0,q_1\},010) = \nu^*(\nu(q_0,0) \cup \nu(q_1,0), 10) = \nu^*(\{q_0\} \cup \{q_2\}, 10) = \nu^*(\{q_0,q_2\}, 10)$

4. urratsa: $\nu^*(\{q_0,q_2\},10) = \nu^*(\nu(q_0,1) \cup \nu(q_2,1), 0) = \nu^*(\{q_0,q_1\} \cup \varnothing, 10) = \nu^*(\{q_0,q_1\}, 0)$

5. urratsa: $\nu^*(\{q_0,q_1\},0) = \nu^*(\nu(q_0,0) \cup \nu(q_1,0), \varepsilon) = \nu^*(\{q_0\} \cup \{q_2\}, \varepsilon) = \nu^*(\{q_0,q_2\}, \varepsilon)$

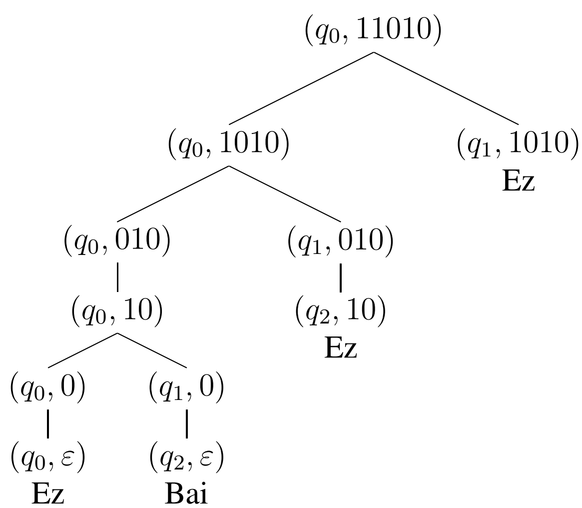
6. urratsa: $\nu^*(\{q_0,q_2\},\varepsilon) = \{q_0,q_2\}$

Konputazioa $\{q_0,q_2\}$ multzoan bukatzen denez eta multzo horretan Y multzokoa den elementu bat badagoenez, $\{q_0,q_2\} \cap Y \neq \varnothing$ betetzen da eta, ondorioz, hitz hori adibide honetako lengoaiakoa da.

Konputazio hori honako konfigurazio ez-deterministen zerrenda bezala ipini daiteke:

$$
\begin{array}{c}
(\{q_0\}, 11010) \\
| \\
(\{q_0,q_1\}, 1010) \\
| \\
(\{q_0,q_1\}, 010) \\
| \\
(\{q_0,q_2\}, 10) \\
| \\
(\{q_0,q_1\}, 0) \\
| \\
(\{q_0,q_2\}, \varepsilon)
\end{array}
$$

Konfigurazio deterministez eratutako zuhaitz bezala adieraziz honela geratuko litzateke konputazio hori:



Y multzokoa den egoera batez eta ε hitz hutsaz eratutako konfigurazio determinista batean bukatzen den adar bat badagoenez, 11010 hitza adibide honetako lengoaiakoa da.

3.3.3 AFEDen diseinua

AFDekin gertatzen den bezala, AFED baten zeregina hitz bat AFED horren bidez definitutako lengoaiakoa al den erabakitzea da. Beste era batera esanda, AFED baten zeregina AFED horri sarrerako datutzat emango zaion hitz bakoitzak propietate jakin bat betetzen al duen erabakitzea da. Beraz, AFDetan bezala, AFEDa sinboloz sinbolo hitza irakurriz joango da eta irakurriz doan hitz horrek betetzen duen propietatea zein den "gogoratzeko" baliabide gisa egoerak erabiliko dira. Egoera bakoitzak propietate bat "gogoratzen" edo adierazten du. Une bakoitzean, une horretara arte irakurri den hitz zatiak (edo azpihitzak) zein propietate betetzen duen jakiteko AFEDa zein egoeratan dagoen begiratu beharko da. AFEDak hitza sinboloz sinbolo irakurriko du eta ezingo du atzeraka egin. Beraz, une bakoitzean AFEDa zein egoeratan dagoen jakinda, une horretara arte irakurri den hitz zatiak zein propietate betetzen duen jakingo da eta hitza osatzen duten gainerako sinboloak irakurriz jarraitu beharko da. **Automaten diseinu metodikoan edo sistematikoan**, funtsezko oinarria hitzek bete ditzaketen propietate garratzitsuak edo esanguratsuak "gogoratzeko" beharrezkoak diren egoerak identifikatzea da. Beharrezkoak diren egoerak zein izango diren erabaki ondoren, egoeren arteko trantsizioak finkatu beharko dira. AFDen diseinurako azaldutako teknika batzuk AFEDen diseinurako ere balio dute. Hain zuzen ere, egoerak identifikatzean oinarritzen den teknika, lehenengo egoera-multzoak identifikatu eta gero egoera-multzo horietako egoerak zehaztean oinarritzen den teknika eta propietateen konjuntzioa dugunerako teknika AFEDak diseinatzeko ere erabil daitezke. Aldiz, 3.2.2.3 atalean propietateen ukapenarentzat azaldutako teknika eta 3.2.2.5 atalean disjuntzioarentzat azaldutako teknika ezingo dira erabili. Propietateen disjuntzioa dugunerako, AFEDak elkartzean edo bat egitean datzan teknika sistematiko bat azalduko da. Dagoeneko 3.3.1 atalean disjuntzioaren kasurako adibide batzuk azaldu dira baina adibide horietan ez da teknika sistematikoa erabili (ikus 3.3.7, 3.3.8 eta 3.3.9 irudiak, baita 3.3.7, 3.3.10 eta 3.3.11 irudiak ere eta baita 3.3.7, 3.3.12 eta 3.3.42 irudiak ere). Gainera, edozein kasutan, AFEDen berezko ezaugarri bereziak kontuan hartu beharko dira. Ezaugarri horiek 3.3.2 atalean aurkeztu dira baina baita 3.3 atala osatzen duten beste azpiataletan ere.

3.3.3.1 AFEDen diseinua egoerak identifikatuz

Automata finituak sistematikoki diseinatzeko aukera bat, automaten diseinu metodikoan edo sistematikoan funtsezkoa den oinarria zuzenean aplikatzen saiatzea da, hau da, urrats bakoitzaren ondoren (sinbolo bakoitza irakurri ondoren) ordura arte irakurritako hitz zatiak bete ditzakeen propietate garrantzitsuak edo esanguratsuak "gogoratzeko" beharrezkoak diren egoerak zuzenean identifikatzea.

Teknika hau hiru adibide garatuz erakutsiko da orain. Lehenengo bi adibideetan lortuko diren AFEDen trantsizo-diagramak 3.2.2.1 atalean diseinatutako AFDen trantsizio-diagramen berdinak dira. Desberdintasuna AFDen eta AFEDen trantsizio-funtzioen motan dago. AFDetan trantsizio-funtzioaren mota $Q \times A \rightarrow Q$ da baina AFEDetan trantsizio-funtzioaren mota $Q \times A \rightarrow 2^Q$ da. Adibide denetan alfabetoa $A = \{a, b, c\}$ da.

3.3.3.1.1 a kopuru bikoitia duten hitzen lengoaiari dagokion AFED bat

Har dezagun $A = \{a, b, c\}$ alfabetoa. Alfabeto horren gainean definituta dauden eta a kopuru bikoitia duten hitzen lengoaiari dagokion AFED bat diseinatu nahi da. Lengoaia hori honela defini daiteke:

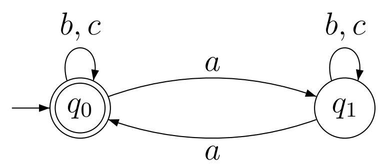
$$\{w | w \in A^* \wedge |w|_a \bmod 2 = 0\}$$

Hitz batek a kopuru bikoitia al duen erabakitzeko, nahikoa da une bakoitzean une horretara arte irakurritako a kopurua bikoitia ala bakoitia izan den gogoratzearekin. Ondorioz, bi egoera beharko dira. Hitza irakurtzen hasi aurretik a kopurua (zero) bikoitia da. Beraz, hasierako egoera a kopurua bikoitia dela "gogoratzeko" erabil daiteke. Aldiz, AFEDa beste egoeran baldin badago, une horretara arte irakurritako a kopurua bakoitia dela esan nahiko du:

q_0: orain arte irakurritako a kopurua bikoitia da.

q_1: orain arte irakurritako a kopurua bakoitia da.

Beharrezkoak diren egoerak zein diren erabaki ondoren, egoeren arteko trantsizioak finkatu beharko dira. Horretarako, b edo c agertzeak a-ren agerpen kopurua bikoitia edo bakoitia izatean eraginik ez duela kontura gaitezke. Hori dela-eta, b edo c irakurtzen bada, ez da egoeraz aldatu beharko. Aldiz, a sinboloaren beste agerpen batek egoera aldaketa dakar berarekin.



3.3.18 irudia. a kopuru bikoitia duten hitzen lengoaiari dagokion AFED bat.

3.3.18 irudian a kopuru bikoitia duten hitzez osatutako lengoaiari dagokion AFED baten trantsizio-diagrama erakusten da. Kasu honetan, ez dago AFEDen ezaugarri bereziak erabiltzeko aukerarik eta, horregatik, lortutako AFEDaren trantsizio-diagrama 3.2.2.1 atalean diseinatutako AFDaren trantsizio-diagramaren berdina da (ikus 3.2.7 irudia). Trantsizio-diagrama bera izan arren, definizio formaltzat hartzen den boskotea desberdina izango da. Izan ere, trantsizio-funtzioaren mota aldatuko da.

Jarraian AFED horren osagaiak definituko dira. Adibide honetako AFEDa honako boskote honen bidez defini daiteke:

$$(Q, A, \nu, q_0, Y)$$

Boskote horretan:

- $Q = \{q_0, q_1\}$

- $A = \{a, b, c\}$
- $\nu : Q \times A \to 2^Q$ honako taula honen bidez definitutako trantsizio-funtzioa da:

ν	a	b	c
q_0	$\{q_1\}$	$\{q_0\}$	$\{q_0\}$
q_1	$\{q_0\}$	$\{q_1\}$	$\{q_1\}$

- q_0 hasierako egoera da.
- $Y = \{q_0\}$.

3.3.3.1.2 a kopuru bikoitia eta b kopuru bikoitia duten hitzen lengoaiari dagokion AFED bat

Har dezagun $A = \{a, b, c\}$ alfabetoa. a kopuru bikoitia eta, gainera, b kopurua ere bikoitia duten hitzez osatutako lengoaiarentzat AFED bat diseinatu nahi da. Lengoaia hori formalki honela defini daiteke:

$$\{w | w \in A^* \wedge (|w|_a \bmod 2 = 0) \wedge (|w|_b \bmod 2 = 0)\}$$

Hitz batek a kopuru bikoitia eta b kopuru bikoitia al dituen erabakitzeko, une bakoitzean une horretara arte irakurritako a kopurua bikoitia ala bakoitia al den eta une horretara arte irakurritako b kopurua bikoitia ala bakoitia al den gogoratzea da aukera bat. Ondorioz, lau egoera beharko dira lau konbinazio posible baitaude. Hitza irakurtzen hasi aurretik a kopurua (zero) bikoitia izango da eta b kopurua ere (zero) bikoitia izango da. Beraz, hasierako egoera a kopurua eta b kopurua, biak, bikoitiak direla "gogoratzeko" erabil daiteke. Beste hiru egoerek gelditzen diren beste hiru aukera edo konbinazioak gogoratzeko balioko dute:

q_0: orain arte irakurritako a kopurua bikoitia da eta b kopurua ere bikoitia da.

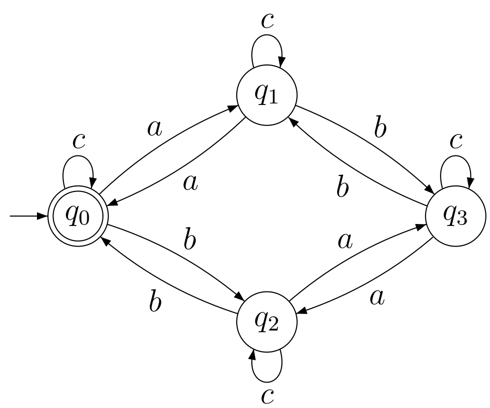
q_1: orain arte irakurritako a kopurua bakoitia da eta, aldiz, b kopurua bikoitia da.

q_2: orain arte irakurritako a kopurua bikoitia da eta, aldiz, b kopurua bakoitia da.

q_3: orain arte irakurritako a kopurua bakoitia da eta b kopurua ere bakoitia da.

Egoerak zein izango diren erabaki ondoren, trantsizioak finkatu behar dira. Hasteko, c agertzeak a kopurua bikoitia edo bakoitia izatean eraginik ez duela eta, era berean, b kopurua bikoitia edo bakoitia izatean ere eraginik ez duela kontura gaitezke. Beraz, c irakurritakoan ez da egoeraz aldatu behar. Bestalde, a edo b agertzeak a kopurua edo b kopurua bikotitik bakoitira edo bakoititik bikoitira igarotzea dakar. Hori dela-eta, a edo b agertzeak egoera-aldaketa dakar berarekin.

3.3.19 irudian a kopuru bikoitia eta b kopuru bikoitia duten hitzez osatutako lengoaiarentzat diseinatutako AFEDari dagokion trantsizio-diagrama erakusten da. Kasu honetan ere, ezin



3.3.19 irudia. a kopuru bikoitia eta b kopuru bikoitia duten hitzen legoaiari dagokion AFED bat.

izan dira AFEDen ezaugarri bereziak erabili eta, horregatik, lortutako AFEDaren trantsizio-diagrama 3.2.2.1 atalean diseinatutako AFDaren trantsizio-diagramaren berdina da (ikus 3.2.8 irudia). Trantsizio-diagrama bera izan arren, definizio formaltzat hartzen den boskotea desberdina izango da. Izan ere, trantsizio-funtzioaren mota aldatuko da.

Jarraian AFED horren osagaiak definituko dira. Adibide honetako AFEDa honako boskote honen bidez formalizatu daiteke:

$$(Q, A, \nu, q_0, Y)$$

Boskote horretan:

- $Q = \{q_0, q_1, q_2, q_3\}$
- $A = \{a, b, c\}$
- $\nu : Q \times A \to 2^Q$ honako taula honen bidez definitutako trantsizio-funtzioa da:

ν	a	b	c
q_0	$\{q_1\}$	$\{q_2\}$	$\{q_0\}$
q_1	$\{q_0\}$	$\{q_3\}$	$\{q_1\}$
q_2	$\{q_3\}$	$\{q_0\}$	$\{q_2\}$
q_3	$\{q_2\}$	$\{q_1\}$	$\{q_3\}$

- q_0 hasierako egoera da.
- $Y = \{q_0\}$.

3.3.3.1.3 aa katea bai baina c-rik ez duten hitzen lengoaiarentzat AFED bat

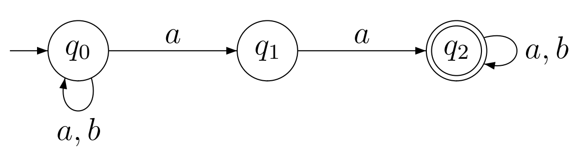
Har dezagun $A = \{a, b, c\}$ alfabetoa. aa katea bai baina c-rik ez duten hitzen lengoaiarentzat AFED bat diseinatu nahi da. Lengoaia hori formalki honela defini daiteke:

$$\begin{aligned}\{w \mid w \in A^* \wedge \quad & \exists u, v(u \in A^* \wedge v \in A^* \wedge w = uaav) \\ & \wedge \neg\exists u, v(u \in A^* \wedge v \in A^* \wedge w = ucv)\}\end{aligned}$$

Hitz batek aa katea gutxienez behin izatea eta c-ren agerpenik ez izatea betetzen al duen erabakitzeko, alde batetik c-rik agertu al den gogoratu beharko da —kasu horretan behin betiko erantzuna ezezkoa izango da— eta bestetik, c ez bada oraindik agertu, aa katea irakurri al den —baldintza hori betez— edo aa oraindik ez dela irakurri baina irakurritako azkeneko sinboloa a izan al den edo aa oraindik ez dela irakurri eta irakurritako azkeneko sinboloa ere a ez dela izan gogoratu beharko da. Beraz, lau egoera beharko dira:

- q_0: orain arte c ez da agertu, aa ere ez da agertu eta irakurritako azkeneko sinboloa ez da a izan.
- q_1: orain arte c ez da agertu, aa ere ez da agertu baina irakurritako azkeneko sinboloa a izan da. Oraintxe irakurri den a hori aa bateko lehenengo a izan daiteke.
- q_2: orain arte c ez da agertu baina aa agertu da.
- q_3: c agertu da. Berdin da aa agertu al den ala ez.

AFEDen berezko ezaugarriak kontuan hartuz, q_3 egoera ezaba daiteke. Horrela, c sinboloarentzat ez da trantsiziorik egongo eta c sinboloa agertu bezain laster AFEDa gelditu egingo da eta "Ez" erantzungo du. Gainerako trantsizioei dagokienez, AFEDa q_0 egoeran dagoela b sinboloa irakurtzen bada, ez gaude aa baten atarian eta, ondorioz, q_0 egoeran jarraituko du AFEDak. q_0 egoeran a irakurtzen bada, aa baten atarian egon gaitezke eta q_1 egoerara igaro beharko da a bat dugula eta aa baten aurrean egon gintezkeela gogoratzeko edo, bestela, aa baten lehenengo osagaia ez den a baten aurrean egon gintezke eta kasu horretan q_0 egoeran bertan mantendu beharko genuke. Bestalde, q_1 egoeran baldin bagaude, badakigu a bat irakurri dela eta jarraian beste a bat irakurtzen bada, q_2 egoerara igaro beharko da, aa katea osatu delako. Azkenik, q_2 egoeran baldin bagaude, bai a eta bai b irakurrita ere AFEDa egoera berean mantenduko da, aa dagoeneko agertu baita eta orain arte c-rik ez da agertu.



3.3.20 irudia. aa katea duten baina c-rik ez duten hitzen lengoaiarentzat diseinatutako AFEDa.

aa duten baina c-rik ez duten hitzez osatutako lengoaiarentzat diseinatu den AFEDaren trantsizio-diagrama 3.3.20 irudian dugu. Adibide honetan AFEDen berezko ezaugarriak erabili ahal izan dira.

Diseinatutako AFEDa formalki honako boskote honen bidez adieraz daiteke:

$$(Q, A, \nu, q_0, Y)$$

Boskote horretan:

- $Q = \{q_0, q_1, q_2\}$
- $A = \{a, b, c\}$
- $\nu : Q \times A \to 2^Q$ honako taula honen bidez definitutako trantsizio-funtzioa da:

ν	a	b	c
q_0	$\{q_0, q_1\}$	$\{q_0\}$	$\varnothing$
q_1	$\{q_1\}$	$\varnothing$	$\varnothing$
q_2	$\{q_2\}$	$\{q_2\}$	$\varnothing$

- q_0 hasierako egoera da.
- $Y = \{q_2\}$.

3.3.3.1.4 $a^j b^k$ egitura duten hitzen lengoaiarentzat diseinatutako AFED bat

Har dezagun $A = \{a, b, c\}$ alfabetoa. $j \geq 0$ eta $k \geq 0$ izanda, $a^j b^k$ egitura duten hitzez osatutako lengoaiarentzat AFED bat diseinatu nahi da. Lengoaia hori formalki honela defini daiteke:

$$\{w \mid w \in A^* \wedge \exists j, k(j \in I\!N \wedge k \in I\!N \wedge w = a^j b^k)\}$$

Hitz batek, $j \geq 0$ eta $k \geq 0$ izanda, $a^j b^k$ egitura al duen erabakitzeko, une bakoitzean gogoratu beharreko propietate esanguratsutzat honako hauek har ditzakegu:

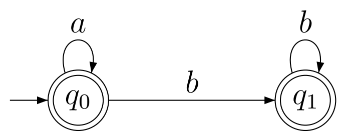
- Lehenengo, a sinboloaren errepikapenez eratutako blokean al gauden (bloke hori hutsa izan daiteke).
- Bigarren lekuan, a-ren errepikapenez eratuako blokea bukatu eta b sinboloaren errepikapenez eratutako blokean al gauden (bloke hori ere hutsa izan daiteke).
- Hirugarren lekuan, c-rik agertu al den edo b sinboloaren errepikapenez eratutako blokea bukatu ondoren a-ren bat agertu al den $a^j b^k$ egitura apurtuz.

Hirugarren kasuan behin betiko erantzuna “Ez” izango denez (hitzeko gainerako sinboloak irakurri beharrik gabe), ez dago kasu horretarako egoera bat ipini beharrik. Planteamendu horri jarraituz, bi egoera nahikoa dira:

q_0: orain arte c-rik ez da agertu eta a sinboloaren agerpenez eratutako blokean gaude.

q_1: orain arte c-rik ez da agertu eta a sinboloaren agerpenez eratutako blokea bukatu da eta b sinboloaren agerpenez eratutako blokean gaude.

Zein egoera beharko diren erabaki ondoren, trantsizioak zehaztu beharko dira. Edozein egoeratan egonda ere, c agertzen bada badakigu "Ez" erantzun beharko dugula eta, ondorioz, c-rekin ez da trantsiziorik ipini behar. AFEDa q_0 egoeran baldin badago eta a irakurtzen bada, q_0 egoeran mantendu beharko du AFEDak baina b irakurtzen bada, q_1 egoerara igaro beharko du. Oraingo egoera q_1 baldin bada eta b agertzen bada, AFEDak q_1 egoeran jarraitu beharko du baina a agertzen bada, behin betiko erantzuna "Ez" izango da eta, horregatik, q_1 egoeratik ez da trantsiziorik ipini behar a sinboloarekin. 3.3.21 irudian oraintxe aipatutako irizpideak kontuan hartuz eraiki den AFEDaren trantsizio-diagrama dago.



3.3.21 irudia. $j \geq 0$ eta $k \geq 0$ izanda, $a^j b^k$ egitura duten hitzen lengoaiari dagokion AFED bat.

Diseinatutako AFEDa formalki honako boskote honen bidez defini daiteke:

$$(Q, A, \nu, q_0, Y)$$

Boskote horretan:

- $Q = \{q_0, q_1\}$
- $A = \{a, b, c\}$
- $\nu : Q \times A \to 2^Q$ honako taula honen bidez definitutako trantsizio-funtzioa da:

ν	a	b	c
q_0	$\{q_0\}$	$\{q_1\}$	$\varnothing$
q_1	$\varnothing$	$\{q_1\}$	$\varnothing$

- q_0 hasierako egoera da.
- $Y = \{q_0, q_1\}$.

3.3.3.2 AFEDen diseinua egoera-multzoak identifikatuz eta gero multzo bakoitzari dagozkion egoerak zehaztuz

AFDentzat 3.2.2.1 atalean azaldu den bezala, lengoaia batekoak diren hitzek betetzen duten baldintza oso erraza ez denean eta, ondorioz, gogoratu beharreko propietate esanguratsuak zein diren identifikatzea eta beharko diren egoerak zein diren identifikatzea erraza ez denean, aukera bat abstrakzio-maila handitzea eta orokorragoak diren propietateak identifikatzea da. Oro har, bigarren urrats batean edo behar bada, urrats-kopuru handiago batean propietate orokor horiek xehatuz joan beharko da. Kasu gehienetan, hasierako propietate orokor bakoitzari egoera multzo bat egokituko zaio xehatze-urratsen ondoren.

Jarraian AFEDak diseinatzeko teknika hau erakusten duten bi adibide emango dira.

3.3.3.2.1 aaa katea gutxienez behin eta b kopuru bikoitia duten hitzen lengoaiarentzat AFED baten diseinua

Har ditzagun $A = \{a, b, c\}$ alfabetoa eta aaa katea gutxienez behin eta b kopuru bikoitia duten hitzez eratutako lengoaia. Lengoaia horrentzat AFED bat diseinatuko dugu orain. Lengoaia hori formalki honela defini daiteke:

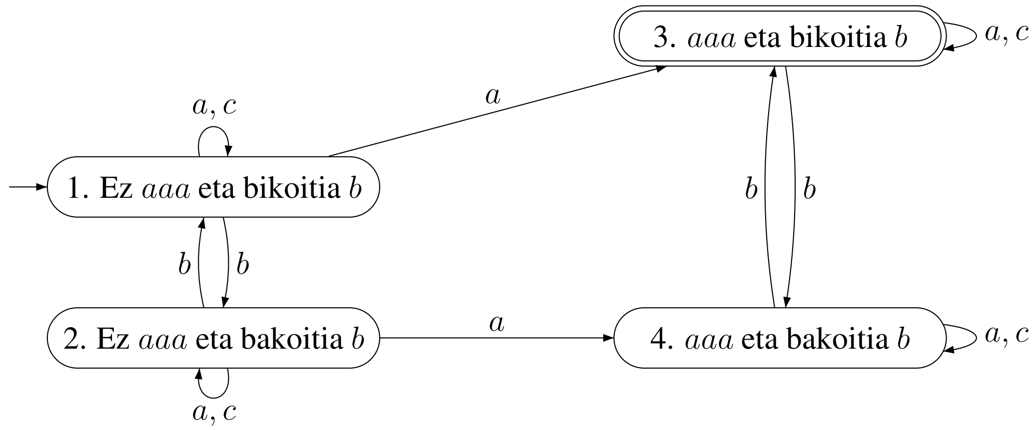
$$\{w | w \in A^* \wedge \exists u, v(u \in A^* \wedge v \in A^* \wedge w = uaaav) \wedge (|w|_b \bmod 2 = 0)\}$$

Lehenengo urrats bezala lau kasu bereiz ditzakegu:

1. Oraindik aaa ez da agertu eta b kopurua bikoitia da.
2. Oraindik aaa ez da agertu eta b kopurua bakoitia da.
3. aaa agertu da eta b kopurua bikoitia da.
4. aaa agertu da eta b kopurua bakoitia da.

3.3.22 irudian lehenengo urrats horren ondorio bezala identifikatutako blokeak eta bloke (edo egoera-multzo) horien arteko trantsizioak erakusten dituen eskema dugu (eskema hori 3.2.12 irudian emandako eskema bera da). Bigarren urrats bezala, bloke horietako bakoitza xehatu egin beharko da. Bai lehenengo blokean eta bai bigarren blokean, jarraian dauden zero a, a bat eta jarraian dauden bi a daudela gogoratzeko hiru egoera beharko dira. Hirugarren eta laugarren blokeetan egoera bat nahikoa da. Beharrezkoak diren egoera denak identifikatu ondoren, egoera horien arteko trantsizioak zehaztu beharko dira.

3.3.23 irudian adibide honetan eraiki den AFEDaren trantsizio-diagrama dugu. Eraiki den AFED hori 3.2.2.2 atalean diseinatu den AFDarekin alderatzea edo konparatzea komeni da (AFD horren trantsizio-diagrama 3.2.13 irudian ikus daiteke). Lehenengo blokea xehatuz q_0, q_1 eta q_2 egoerak sortu dira; bigarren blokea xehatuz q_4, q_5 eta q_6 egoerak sortu dira; hirugarren eta laugarren blokeetatik q_3 eta q_7 egoerak sortu dira hurrenez hurren. Baina AFEDek dituzten berezko ezaugarriak kontuan hartuz, oraintxe aipatu den AFD horrekiko aldaketa batzuk egin



3.3.22 irudia. aaa katea gutxienez behin eta b kopuru bikoitia duten hitzen lengoaiarentzat AFED bat eraikitzeko eskema.

dira. Zehatzago esanez, c sinboloarekin q_1 eta q_2 egoeretatik q_0 egoerara zeuden trantsizioak eta q_5 eta q_6 egoeretatik q_4 egoerara zeuden trantsizioak kendu egin dira. b sinboloarekin q_1 eta q_2 egoeretatik q_4 egoerara zeuden trantsizioak eta q_5 eta q_6 egoeretatik q_0 egoerara zeuden trantsizioak kendu egin dira. q_0 eta q_4 egoeretako begiztetan a sinboloa ere ipini da; izan ere, egoera horietan gaudenean a sinboloa aaa kate baten hasiera izan daiteke edo ez. Lortu den automata (AFEDa) 3.2.13 irudian aurkeztutako automata (AFDa) baino sinpleagoa da.

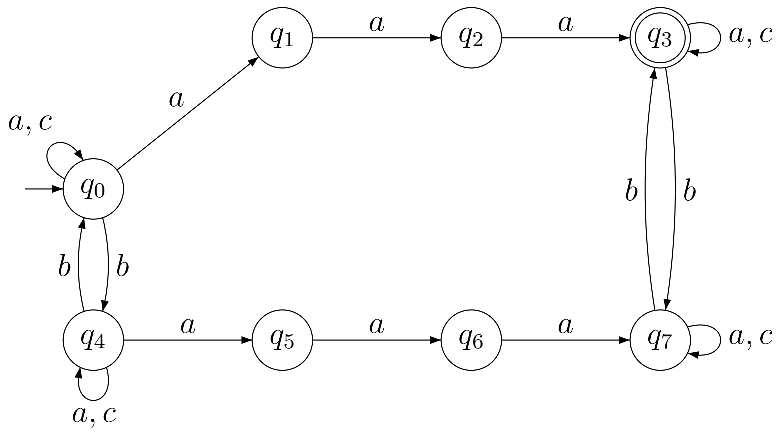
Diseinatutako AFEDa formalki honako boskote honen bidez defini daiteke:

$$(Q, A, \nu, q_0, Y)$$

Boskote horretan:

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7\}$
- $A = \{a, b, c\}$
- $\nu : Q \times A \to 2^Q$ honako taula honen bidez definitutako trantsizio-funtzioa da:

ν	a	b	c
q_0	$\{q_0, q_1\}$	$\{q_4\}$	$\{q_0\}$
q_1	$\{q_2\}$	$\varnothing$	$\varnothing$
q_2	$\{q_3\}$	$\varnothing$	$\varnothing$
q_3	$\{q_3\}$	$\{q_7\}$	$\{q_3\}$
q_4	$\{q_4, q_5\}$	$\{q_0\}$	$\{q_4\}$
q_5	$\{q_6\}$	$\varnothing$	$\varnothing$
q_6	$\{q_7\}$	$\varnothing$	$\varnothing$
q_7	$\{q_7\}$	$\{q_3\}$	$\{q_7\}$



3.3.23 irudia. aaa katea gutxienez behin eta b kopuru bikoitia duten hitzen lengoaiarentzat AFED bat.

- q_0 hasierako egoera da.
- $Y = \{q_3\}$.

3.3.3.2.2 aa baduten baina bb ez duten hitzen lengoaiarentzat AFED baten diseinua

Har dezagun $A = \{a, b, c\}$ alfabetoa. Gutxienez aa katearen agerpen bat eta bb katearen agerpenik ez duten hitzez eratutako lengoaiarentzat AFED bat diseinatu nahi da. Lengoaia hori formalki honela defini daiteke:

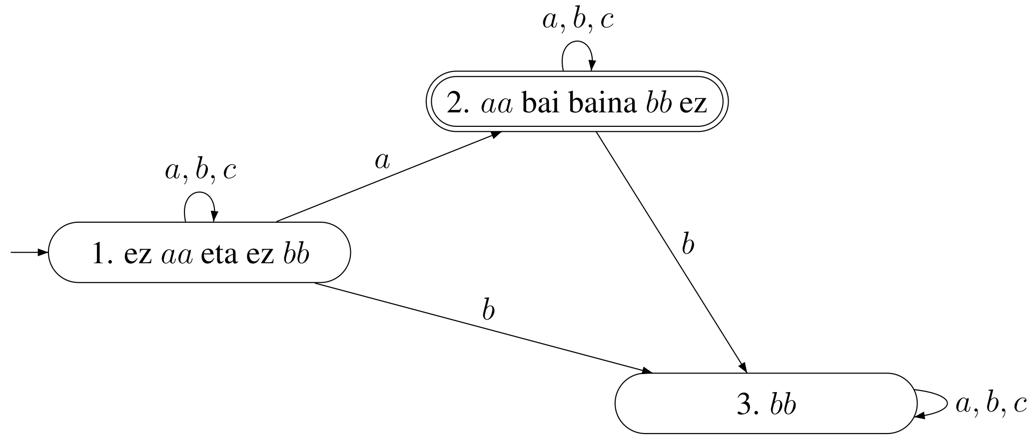
$$\{w|w \in A^* \wedge \exists u, v(u \in A^* \wedge v \in A^* \wedge w = uaav) \wedge \neg\exists u, v(u \in A^* \wedge v \in A^* \wedge w = ubbv)\}$$

Lehenengo urratsean honako hiru kasu hauek kontsidera daitezke:

1. Oraindik ez da agertu ez aa eta ez bb.
2. aa agertu da baina bb ez.
3. bb agertu da eta berdin da aa agertu al den ala ez.

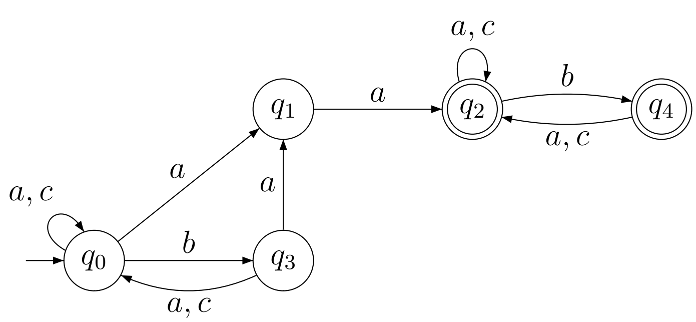
3.3.24 irudian lehenengo urrats horretan identifikatutako bloke edo egoera multzoen arteko trantsizioak erakusten dituen eskema dugu (3.2.2.2 ataleko 3.2.14 irudian dagoen eskema bera da).

Eskema eraiki ondoren, bigarren urrats bezala bloke bakoitza xehatu egin beharko da. Lehenengo blokean hiru egoera beharko dira une honetan jarraian dauden zero a eta zero b ditugula edo a bat dugula edo b bat dugula gogoratzeko edo adierazteko. Bigarren blokean bi egoera



3.3.24 irudia. aa bai baina bb-rik ez duten hitzez eratutako lengoaiari dagokion AFED baten eskema.

beharko dira une honetan jarraian dauden zero b ditugula edo b bat dugula gogoratzeko. Hirugarren blokean bb katea osatuta dago eta AFEDak "Ez" erantzun beharko du, baina horretarako nahikoa da egoerarik ez ipintzearekin. Beharrezkoak diren egoerak identifikatu ondoren, trantsizioak zehaztu beharko dira.



3.3.25 irudia. aa bai baina bb-rik ez duten hitzez eratutako lengoaiari dagokion AFED bat.

Lortutako AFEDari dagokion trantsizio-diagrama 3.3.25 irudian ikus daiteke. Eskemako lehenengo blokea xehatuz q_0, q_1 eta q_3 egoerak sortu dira; bigarren blokea xehatuz q_2 eta q_4 egoerak sortu dira; hirugarren blokea xehatuz ez da egoerarik sortu. Bai q_3 egoeran eta bai q_0 egoeran a bat aa kate baten hasiera izan daiteke edo ez. AFED hau 3.2.15 irudiko AFDarekin alderatzea edo konparatzea komeni da (3.2.2.2 atala).

Diseinatutako AFEDa formalki honako boskote honen bidez defini daiteke:

$$(Q, A, \nu, q_0, Y)$$

Boskote horretako osagaiak honako hauek dira:

- $Q = \{q_0, q_1, q_2, q_3, q_4\}$
- $A = \{a, b, c\}$
- $\nu : Q \times A \to 2^Q$ honako taula honen bidez definitutako trantsizio-funtzioa da:

ν	a	b	c
q_0	$\{q_0, q_1\}$	$\{q_3\}$	$\{q_0\}$
q_1	$\{q_2\}$	$\varnothing$	$\varnothing$
q_2	$\{q_2\}$	$\{q_4\}$	$\{q_2\}$
q_3	$\{q_0, q_1\}$	$\varnothing$	$\varnothing$
q_4	$\{q_2\}$	$\varnothing$	$\{q_2\}$

- q_0 hasierako egoera da.
- $Y = \{q_2, q_4\}$.

3.3.3.3 AFEDen diseinua propietateen ukapenentzat

3.2.2.3 atalean erakutsi den bezala, lengoaia bateko hitzek propietate baten ukapena bete behar dutenean lengoaia horri dagokion AFD bat eraikitzeko estrategia honako hau izan daiteke: lehenengo, lengoaia osagarriari, hau da, propietatea betetzen duten hitzen lengoaiari dagokion AFD bat eraiki eta, gero, lengoaia osagarriarentzat eraikitako AFD horretan zirkulu bikoitzik ez duten egoerei zirkulu bikoitza ipini eta zirkulu bikoitza duten egoerei zirkulu bikoitza kendu. Baina teknika horrek ez du balio AFEDentzat. Ondorioz, propietate baten ukapena bete behar duten hitzez eratutako lengoaia bati dagokion AFEDa eraikitzeko, egoerak identifikatzean eta egoera-multzoak identifikatzean eta egoera multzo horiek osatzen dituzten egoerak identifikatzean oinarritzen diren teknikak erabili beharko dira. Teknika horiek 3.3.3.1 eta 3.3.3.2 ataletan aurkeztu dira.

Ukatutako propietate bat betetzen duten hitzez osatutako lengoaia bati dagokion AFEDa eraiki nahi denean, propietate hori betetzen duten hitzez osatutako lengoaiari dagokion AFEDtik era sistematikoan ezin dela lortu erakusten duten bi adibide aurkeztuko dira jarraian.

3.3.3.3.1 aaa ez duten hitzez eratutako lengoaiarentzat AFED bat diseinatzeko arrakastarik gabeko saiakera

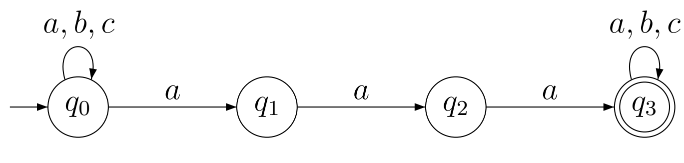
Har dezagun $A = \{a, b, c\}$ alfabetoa. aaa katea **ez** duten hitzez osatutako L lengoaiarentzat AFED bat diseinatu nahi da. L lengoaia hori formalki honela defini daiteke:

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

L lengoaia horretako hitzek propietate baten ukapena bete behar dute. Propietatea aaa katea edukitzea da. $\overline{L}$ lengoaia osagarria aaa katea duten hitzez eratutakoa izango da:

$$\overline{L} = \{w \mid w \in A^* \wedge \exists u, v(u \in A^* \wedge v \in A^* \wedge w = uaaav)\}$$

$\overline{L}$ lengoaian ez dago ukapenik. Hasteko, $\overline{L}$ lengoaiarentzat AFED bat diseinatuko genuke.



3.3.26 irudia. aaa duten hitzen lengoaiarentzat AFED bat.

3.3.26 irudian $\overline{L}$ lengoaiarentzat diseinatutako AFED baten trantsizio-diagrama dugu. AFED hori formalki honako boskote honen bidez adieraz daiteke:

$$(Q, A, \nu, q_0, Y)$$

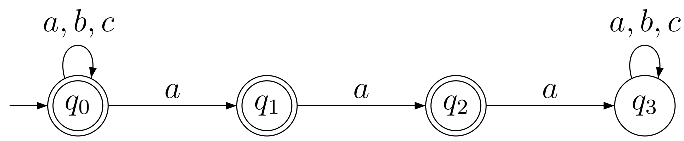
Boskote horretako osagai bakoitza honela definituta dago:

- $Q = \{q_0, q_1, q_2, q_3\}$
- $A = \{a, b, c\}$
- $\nu : Q \times A \to 2^Q$ honako taula honen bidez definitutako trantsizio-funtzioa da:

ν	a	b	c
q_0	$\{q_0, q_1\}$	$\{q_0\}$	$\{q_0\}$
q_1	$\{q_2\}$	$\varnothing$	$\varnothing$
q_2	$\{q_3\}$	$\varnothing$	$\varnothing$
q_3	$\{q_3\}$	$\{q_3\}$	$\{q_3\}$

- q_0 hasierako egoera da.
- $Y = \{q_3\}$.

$\overline{L}$ lengoaiari dagokion AFEDa hartu eta egitura bera mantenduz, zirkulu bikoitza ez duten egoerei zirkulu bikoitza ipini eta zirkulu bikoitza duten egoerei zirkulu bikoitza kendu ondoren lortzen den AFEDa aztertzen badugu (ikus trantsizio-diagrama 3.3.27 irudian), $baaab$ bezalako hitzentzat erantzuna baiezkoa izango da baina erantzunak ezezkoa izan beharko luke. Baiezko erantzuna lortzearen arrazoia honako hau da: $(q_0, baaab)$ konfigurazioari dagokion konputazioak $\{q_0, q_3\}$ itzuliko du erantzun edo emaitza bezala eta L-rentzat eraiki den AFEDan q_0 egoerak zirkulu bikoitza du. Beraz, L-rentzat eraiki den AFED horrek ez du balio.



3.3.27 irudia. aaa ez duten hitzen lengoaiarentzat huts egiten duen AFED bat (ukapenaren teknika erabiliz).

L lengoaiarentzat eraiki den AFED hori honako boskote hau bezala adieraz daiteke:

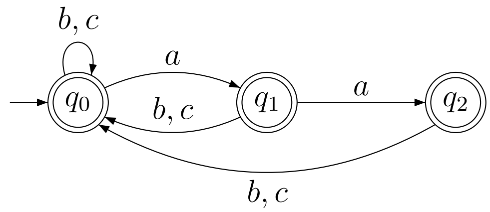
$$(Q, A, \nu, q_0, Y)$$

Boskote horretako osagaiak honako hauek dira:

- $Q = \{q_0, q_1, q_2, q_3\}$
- $A = \{a, b, c\}$
- $\nu : Q \times A \to 2^Q$ honako taula honen bidez definitutako trantsizio-funtzioa da:

ν	a	b	c
q_0	$\{q_0, q_1\}$	$\{q_0\}$	$\{q_0\}$
q_1	$\{q_2\}$	$\varnothing$	$\varnothing$
q_2	$\{q_3\}$	$\varnothing$	$\varnothing$
q_3	$\{q_3\}$	$\{q_3\}$	$\{q_3\}$

- q_0 hasierako egoera da.
- $Y = \{q_0, q_1, q_2\}$.



3.3.28 irudia. aaa ez duten hitzen lengoaiarentzat AFED bat.

3.3.28 irudian, L lengoaiarentzat egokia den AFED baten trantsizio-diagrama erakusten da. AFED hori honako boskote hau bezala adieraz daiteke:

$$(Q, A, \nu, q_0, Y)$$

Boskote horretako osagaiak honako hauek dira:

- $Q = \{q_0, q_1, q_2\}$
- $A = \{a, b, c\}$
- $\nu : Q \times A \to 2^Q$ honako taula honen bidez definitutako trantsizio-funtzioa da:

ν	a	b	c
q_0	$\{q_1\}$	$\{q_0\}$	$\{q_0\}$
q_1	$\{q_2\}$	$\{q_0\}$	$\{q_0\}$
q_2	$\varnothing$	$\{q_0\}$	$\{q_0\}$

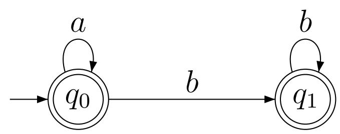
- q_0 hasierako egoera da.
- $Y = \{q_0, q_1, q_2\}$.

3.3.3.3.2 $a^j b^k$ egitura ez duten hitzen lengoaiarentzat AFED bat diseinatzeko arrakastarik gabeko saiakera

Har dezagun $A = \{a, b, c\}$ alfabetoa. $j \geq 0$ eta $k \geq 0$ izanda, $a^j b^k$ egitura **ez** duten hitzez eratutako L lengoaiarentzat AFED bat diseinatu nahi da. L lengoaia hori formalki honela defini daiteke:

$$L = \{w \mid w \in A^* \wedge \neg\exists j, k(j \in I\!N \wedge k \in I\!N \wedge w = a^j b^k)\}$$

L lengoaia horretako hitzek propietate baten ukapena bete behar dute. Propietatea $a^j b^k$ egitura edukitzea da, hor $j \geq 0$ eta $k \geq 0$ izanda. $\overline{L}$ lengoaia osagarria $a^j b^k$ egitura duten hitzez eratutako lengoaia da, hor $j \geq 0$ eta $k \geq 0$ izanda:



3.3.29 irudia. $j \geq 0$ eta $k \geq 0$ izanda, $a^j b^k$ egitura duten hitzen lengoaiarentzat AFED bat.

$$\overline{L} = \{w \mid w \in A^* \land \exists j, k(j \in \mathbb{N} \land k \in \mathbb{N} \land w = a^j b^k)\}$$

$\overline{L}$ lengoaian ez dago ukapenik. L lengoaiarentzat AFED bat eraikitzeko, hasteko $\overline{L}$ lengoaiarentzat AFED bat eraikiko dugu.

3.3.29 irudian $\overline{L}$ lengoaiarentzat eraikitako AFED baten trantsizio-diagrama erakusten da (3.2.2.3 ataleko 3.2.18 irudiko AFDaren trantsizio-diagramarekin alderatzea edo konparatzea komeni da).

$\overline{L}$ lengoaiarentzat eraikitako AFEDa boskote baten bidez formalizatu daiteke:

$$(Q, A, \nu, q_0, Y)$$

Boskote horretako osagai bakoitza honela definituta dago:

- $Q = \{q_0, q_1\}$
- $A = \{a, b, c\}$
- $\nu : Q \times A \to 2^Q$ honako taula honen bidez definitutako trantsizio-funtzioa da:

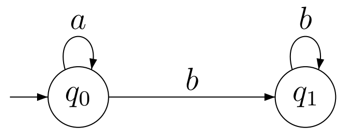
ν	a	b	c
q_0	$\{q_0\}$	$\{q_1\}$	$\varnothing$
q_1	$\varnothing$	$\{q_1\}$	$\varnothing$

- q_0 hasierako egoera da.
- $Y = \{q_0, q_1\}$.

L lengoaiari dagokion AFED bat diseinatzeko, oinarritzat $\overline{L}$ lengoaiarentzat diseinatu dugun AFEDa hartzen badugu eta zirkulu bikoitza duten egoerei zirkulu bikoitza kentzen badiegu eta zirkulu bikoitza ez duten egoerei zirkulu bikoitza ipintzen badiegu, hitz denei “Ez” erantzungo dien AFED bat lortuko dugu. Teknika edo bide hori kontuan hartuz eraikitako AFEDari dagokion trantsizio-diagrama 3.3.30 irudian dugu. AFED horrek ez du balio L lengoaiarentzat.

AFED hori honako boskote honen bidez formaliza daiteke:

$$(Q, A, \nu, q_0, Y)$$



3.3.30 irudia. $j \geq 0$ eta $k \geq 0$ izanda, $a^j b^k$ egitura ez duten hitzen lengoaiarentzat huts egiten duen AFED bat.

Boskote horretako elementu bakoitza honela definituta dago:

- $Q = \{q_0, q_1\}$
- $A = \{a, b, c\}$
- $\nu : Q \times A \to 2^Q$ honako taula honen bidez definitutako trantsizio-funtzioa da:

ν	a	b	c
q_0	$\{q_0\}$	$\{q_1\}$	$\varnothing$
q_1	$\varnothing$	$\{q_1\}$	$\varnothing$

- q_0 hasierako egoera da.
- $Y = \varnothing$.

Jarraian, L lengoaiarentzat egokia den AFED bat aurkeztuko da. AFED egoki horren trantsizio-diagrama 3.2.19 irudian erakutsi den AFDaren trantsizio-diagrama bera da. Trantsizio-diagrama bera eduki arren AFD eta AFED horiek desberdinak dira eta hori garbi ikusten da δ eta ν trantsizio-funtzioen motei begiratuz. L lengoaiarentzat egokia den AFED hori honako boskote hau bezala adieraz daiteke:

$$(Q, A, \nu, q_0, Y)$$

Boskote horretako elementu bakoitza honela definituta dago:

- $Q = \{q_0, q_1, q_2\}$
- $A = \{a, b, c\}$
- $\nu : Q \times A \to 2^Q$ honako taula honen bidez definitutako trantsizio-funtzioa da:

ν	a	b	c
q_0	$\{q_0\}$	$\{q_1\}$	$\{q_2\}$
q_1	$\{q_2\}$	$\{q_1\}$	$\{q_2\}$
q_2	$\{q_2\}$	$\{q_2\}$	$\{q_2\}$

- q_0 hasierako egoera da.
- $Y = \{q_2\}$.

3.3.3.4 AFEDen diseinua propietateen konjuntzioarentzat (Lengoaien arteko ebakidura)

Propietateen konjuntzioa dugunerako AFDak diseinatzeko aurkeztu den teknikak (3.2.2.4 atala) AFEDak diseinatzeko ere balio du.

3.3.3.4.1 aaa katea duten eta a sinboloaz hasten diren hitzez eratutako lengoaiarentzat AFED bat

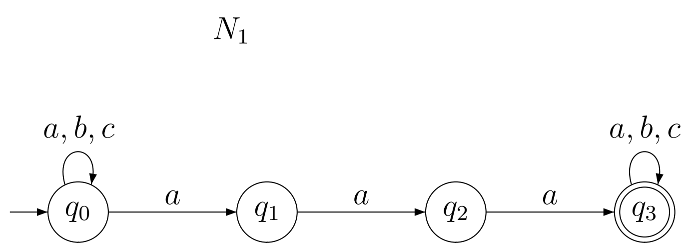
Har dezagun $A = \{a, b, c\}$ alfabetoa. aaa katea gutxienez behin duten eta, gainera, a sinboloaz hasten diren hitzez eratutako L lengoaiarentzat AFED bat diseinatu nahi da. L lengoaia hori formalki honela defini daiteke:

$$L = \{w \mid w \in A^* \wedge \exists u, v(u \in A^* \wedge v \in A^* \wedge w = uaaav) \wedge \exists u(u \in A^* \wedge w = au)\}$$

L lengoaiako hitzek bi propietateren konjuntzioa den baldintza bat bete behar dute: gutxienez aaa bat edukitzea eta a sinboloaz hastea. L lengoaiarentzat AFED bat eraikitzeko, aukera bat propietate horietako bakoitza kontuan hartuz bi AFED eraikitzea da. Beste era batera esanda, honako L_1 eta L_2 lengoaiak kontsidera ditzakegu:

$$\begin{aligned} L_1 &= \{w \mid w \in A^* \wedge \exists u, v(u \in A^* \wedge v \in A^* \wedge w = uaaav)\} \\ L_2 &= \{w \mid w \in A^* \wedge \exists u(u \in A^* \wedge w = au)\} \end{aligned}$$

L lengoaia $L_1 \cap L_2$ da. L_1 lengoaiako hitzek betetzen duten propietatea aaa edukitzea da eta L_2 lengoaiako hitzek betetzen duten propietatea a sinboloaz hastea da. Lehenengo urrats bezala, N_1 AFEDa eta N_2 AFEDa eraikiko ditugu, bata L_1 lengoaiarentzat eta bestea L_2 lengoaiarentzat, hurrenez hurren. 3.3.31 irudian N_1 AFEDari dagokion grafikoa erakusten da eta 3.3.32 irudian N_2 AFEDari dagokion grafikoa erakusten da.



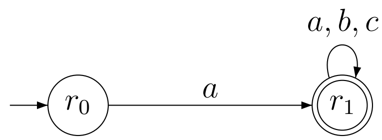
3.3.31 irudia. aaa katea duten hitzez osatutako L_1 lengoaiarentzat diseinatutako AFEDa.

N_1 AFEDa boskote bezala honela defini dezakegu:

$$(Q, A, \nu_1, q_0, Y_1)$$

Boskote horretako osagaiak honako hauek dira:

N_2



3.3.32 irudia. a sinboloaz hasten diren hitzez osatutako L_2 lengoaiarentzat diseinatutako AFEDa.

- $Q = \{q_0, q_1, q_2, q_3\}$
- $A = \{a, b, c\}$
- $\nu_1 : Q \times A \to 2^Q$ honako taula honen bidez definitutako trantsizio-funtzioa da:

ν_1	a	b	c
q_0	$\{q_0, q_1\}$	$\{q_0\}$	$\{q_0\}$
q_1	$\{q_2\}$	$\varnothing$	$\varnothing$
q_2	$\{q_3\}$	$\varnothing$	$\varnothing$
q_3	$\{q_3\}$	$\{q_3\}$	$\{q_3\}$

- q_0 hasierako egoera da.
- $Y_1 = \{q_3\}$.

Bestalde, N_2 AFEDa boskote bezala honela defini dezakegu:

$$(R, A, \nu_2, r_0, Y_2)$$

Boskote horretako osagai bakoitza honela definituta dago:

- $R = \{r_0, r_1\}$
- $A = \{a, b, c\}$
- $\nu_2 : R \times A \to 2^R$ honako taula honen bidez definitutako trantsizio-funtzioa da:

ν_2	a	b	c
r_0	$\{r_1\}$	$\varnothing$	$\varnothing$
r_1	$\{r_1\}$	$\{r_1\}$	$\{r_1\}$

- r_0 hasierako egoera da.

- $Y_2 = \{r_1\}$.

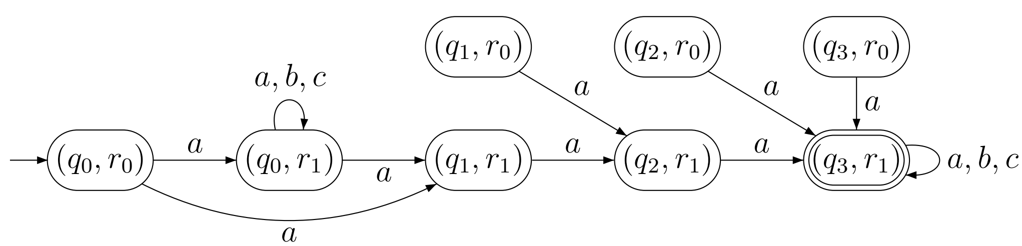
L lengoaiarentzat $N = (S, A, \nu, s_0, Y)$ AFED bat eraikitzeko, hau da, aaa katea gutxienez behin izateaz gain a sinboloaz hasten diren hitzez osatutako L lengoaiarentzat AFED bat eraikitzeko, aukera bat N_1 eta N_2 erabiltzea da. Horretarako, N_1 AFEDko egoera bat eta N_2 AFEDko egoera bat hartuz osa daitezkeen bikote denak hartu beharko dira kontuan. N_1 AFEDko egoerak $Q = \{q_0, q_1, q_2, q_3\}$ baldin badira eta N_2 AFEDko egoerak $R = \{r_0, r_1\}$ baldin badira, N AFEDari dagokion S egoera-multzoa $Q \times R$ izango da. Beraz, Q eta R multzoen arteko biderketa kartesiarrak finkatuko du N AFED berriaren egoera-multzoa:

$$S = Q \times R = \{s_0 = (q_0, r_0), s_1 = (q_0, r_1), s_2 = (q_1, r_0), s_3 = (q_1, r_1), s_4 = (q_2, r_0), s_5 = (q_2, r_1), s_6 = (q_3, r_0), s_7 = (q_3, r_1)\}$$

N AFED berriko hasierako egoera N_1 eta N_2 AFEDetako hasierako egoerez osatutako bikotea izango da: $s_0 = (q_0, r_0)$. Bestalde, N AFEDko onarpen egoerak N_1 AFEDan eta N_2 AFEDan onarpen egoerak direnez osatutako bikoteak izango dira. Beraz, N AFEDan onarpen egoera bakarra egongo da: (q_3, r_1). Era formalagoan esanda, $Y = Y_1 \times Y_2$.

N AFEDko trantsizioei dagokienez, (q_i, r_j) egoeratik (q_k, r_ℓ) egoerara A alfabetokoa den α sinboloarekin trantsizio bat egongo da baldin eta soilik baldin q_i egoeratik q_k egoerara α sinboloarekin N_1 AFEDan trantsizio bat baldin badago eta r_j egoeratik r_ℓ egoerara α sinboloarekin N_2 AFEDan trantsizio bat baldin badago. Formalki, $\nu = \nu_1 \times \nu_2$. Hor, $\nu_1 \times \nu_2$ funtzioa $(Q \times R) \times A \to (Q \times R)$ motakoa izango da eta $\nu_1 \times \nu_2((q_i, r_j), \alpha) = \nu_1(q_i, \alpha) \times \nu_2(r_j, \alpha)$ izango da $Q \times R$ multzoko (q_i, r_j) erako bikote bakoitzeko eta A alfabetoko α sinbolo bakoitzeko.

N



3.3.33 irudia. aaa duten eta a sinboloaz hasten diren hitzez osatutako L lengoaiarentzat AFED bat.

3.3.33 irudian aaa katea duten eta a sinboloaz hasten diren hitzez osatutako L lengoaiarentzat eraiki den N AFEDari dagokion trantsizio-diagrama erakusten da. N AFEDa N_1 eta N_2 AFEDak erabiliz eta propietateen konjuntzioa dugunerako dagoen metodoa aplikatuz lortu da.

N AFED hori honako boskote hau bezala adieraz daiteke matematikoki:

$$(S, A, \nu, s_0, Y)$$

Boskote horietako osagaiak honako era honetan defini daitezke:

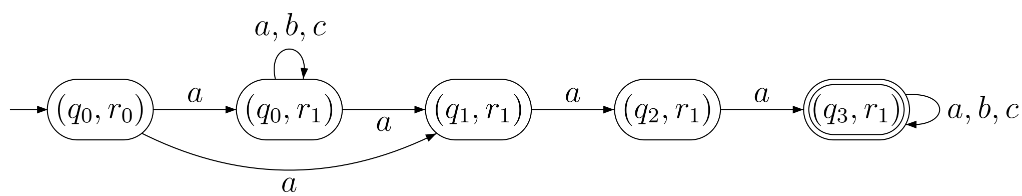
- $S = Q \times R = \{(q_0, r_0), (q_0, r_1), (q_1, r_0), (q_1, r_1), (q_2, r_0), (q_2, r_1), (q_3, r_0), (q_3, r_1)\}$
- $A = \{a, b, c\}$
- $\nu : (Q \times R) \times A \to 2^{(Q \times R)}$ honako taula honen bidez definitutako trantsizio-funtzioa da:

ν	a	b	c
(q_0, r_0)	$\{(q_0, r_1), (q_1, r_1)\}$	$\varnothing$	$\varnothing$
(q_0, r_1)	$\{(q_0, r_1), (q_1, r_1)\}$	$\{(q_0, r_1)\}$	$\{(q_0, r_1)\}$
(q_1, r_0)	$\{(q_2, r_1)\}$	$\varnothing$	$(\varnothing$
(q_1, r_1)	$\{(q_2, r_1)\}$	$\varnothing$	$\varnothing$
(q_2, r_0)	$\{(q_3, r_1)\}$	$\varnothing$	$\varnothing$
(q_2, r_1)	$\{(q_3, r_1)\}$	$\varnothing$	$\varnothing$
(q_3, r_0)	$\{(q_3, r_1)\}$	$\varnothing$	$\varnothing$
(q_3, r_1)	$\{(q_3, r_1)\}$	$\{(q_3, r_1)\}$	$\{(q_3, r_1)\}$

- $s_0 = (q_0, r_0)$ hasierako egoera da.
- $Y = Y_1 \times Y_2 = \{(q_3, r_1)\}$.

(q_0, r_0) bikotea izan ezik, r_0 osagaia duten beste bikote denak iristezinak dira (q_0, r_0) hasierako egoeratik. (q_0, r_0) egoeratik iristezinak diren egoerak ezabatu edo kendu egin behar dira.

N'



3.3.34 irudia. aaa duten eta a sinboloaz hasten diren hitzez osatutako L lengoaiarentzat beste AFED bat (iristezinak diren egoerak kenduta).

3.3.34 irudian, aaa katea gutxienez behin duten eta a sinboloaz hasten diren hitzez eratutako lengoaiari dagokion N AFEDari iristezinak diren egoerak kendu ondoren geratzen den N' AFEDaren trantsizio-diagrama erakusten da. r_1 osagaia duten bikoteetan gaudenean, badakigu hitza a sinboloarekin hasi dela eta q_3 duten bikoteetan gaudenean, badakigu aaa katea agertu dela. Hori dela-eta, bakarrik (q_3, r_1) egoeran bukatzen badugu erantzungo da "Bai". Izan ere, egoera horretan gaudenean aaa agertu da eta hitza a sinboloaz hasi da.

N' AFEDa honako boskote hau bezala adieraz daiteke:

$$(S', A, \nu', s'_0, Y')$$

Boskote horretako osagaiaen definizioa:

- $S' = \{(q_0, r_0), (q_0, r_1), (q_1, r_1), (q_2, r_1), (q_3, r_1)\}$
- $A = \{a, b, c\}$
- $\nu' : S' \times A \to 2^{S'}$ honako taula honen bidez definitutako trantsizio-funtzioa da:

ν'	a	b	c
(q_0, r_0)	$\{(q_0, r_1), (q_1, r_1)\}$	$\varnothing$	$\varnothing$
(q_0, r_1)	$\{(q_0, r_1), (q_1, r_1)\}$	$\{(q_0, r_1)\}$	$\{(q_0, r_1)\}$
(q_1, r_1)	$\{(q_2, r_1)\}$	$\varnothing$	$\varnothing$
(q_2, r_1)	$\{(q_3, r_1)\}$	$\varnothing$	$\varnothing$
(q_3, r_1)	$\{(q_3, r_1)\}$	$\{(q_3, r_1)\}$	$\{(q_3, r_1)\}$

- $s'_0 = (q_0, r_0)$ hasierako egoera da.
- $Y' = Y_1 \times Y_2 = \{(q_3, r_1)\}$.

3.3.3.4.2 a sinboloaz hasi eta a sinboloaz bukatzen diren hitzen lengoaiari dagokion AFED baten diseinua

Har dezagun $A = \{a, b, c\}$ alfabetoa. a sinboloaz hasi eta a sinboloaz bukatzen diren hitzez eratutako L lengoaiarentzat AFED bat diseinatu nahi da. L lengoaia hori formalki honela defini daiteke:

$$L = \{w \mid w \in A^* \wedge \exists u(u \in A^* \wedge w = au) \wedge \exists u(u \in A^* \wedge w = ua)\}$$

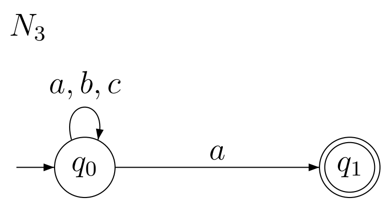
L lengoaiako hitzek bi propietateren konjuntzioa den baldintza bat bete behar dute: a sinboloaz hastea eta a sinboloaz bukatzea. L lengoaiarentzat AFED bat eraikitzeko, aukera bat propietate horietako bakoitza kontuan hartuz bi AFED eraikitzea da. Beste era batera esanda, honako L_2 eta L_3 lengoaiak kontsidera ditzakegu:

$$L_2 = \{w \mid w \in A^* \wedge \exists u(u \in A^* \wedge w = au)\}$$
$$L_3 = \{w \mid w \in A^* \wedge \exists u(u \in A^* \wedge w = ua)\}$$

L lengoaia $L_2 \cap L_3$ da. L_2 lengoaiako hitzek betetzen duten propietatea a sinboloaz hastea da eta L_3 lengoaiako hitzek betetzen duten propietatea a sinboloaz bukatzea da. Hasteko, N_2 AFEDa eta N_3 AFEDa eraikiko ditugu, bata L_2 lengoaiarentzat eta bestea L_3 lengoaiarentzat, hurrenez hurren. 3.3.32 irudian N_2 AFEDari dagokion grafikoa erakusten da eta 3.3.35 irudian N_3 AFEDari dagokion grafikoa erakusten da.

N_2 AFEDa (R, A, ν_2, r_0, Y_2) boskote gisa aurreko atalean definitu da.:

Bestalde, N_3 AFEDa boskote bezala honela adieraz dezakegu:



3.3.35 irudia. a sinboloarekin bukatzen diren hitzen L_3 lengoaiarentzat AFED bat.

$$(Q, A, \nu_3, q_0, Y_3)$$

Boskote horretako osagai bakoitzaren definizioa honako hau da:

- $Q = \{q_0, q_1\}$
- $A = \{a, b, c\}$
- $\nu_3 : Q \times A \to 2^Q$ honako taula honen bidez definitutako trantsizio-funtzioa da:

ν_3	a	b	c
q_0	$\{q_0, q_1\}$	$\{q_0\}$	$\{q_0\}$
q_1	$\varnothing$	$\varnothing$	$\varnothing$

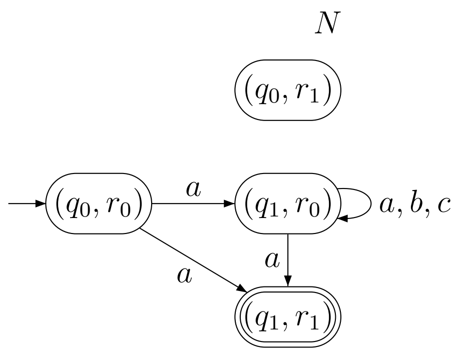
- q_0 hasierako egoera da.
- $Y_3 = \{q_1\}$.

L lengoaiarentzat $N = (S, A, \nu, s_0, Y)$ AFED bat eraikitzeko, hau da, a sinboloaz hasi eta a sinboloaz bukatzen diren hitzez osatutako L lengoaiarentzat AFED bat eraikitzeko, aukera bat N_2 eta N_3 erabiltzea da. Horretarako, N_2 AFEDko egoera bat eta N_3 AFEDko egoera bat hartuz osa daitezkeen bikote denak hartu beharko dira kontuan. N_2 AFEDko egoerak $R = \{r_0, r_1\}$ baldin badira eta N_3 AFEDko egoerak $Q = \{q_0, q_1\}$ baldin badira, N AFEDari dagokion S egoera multzoa $R \times Q$ izango da. Beraz, R eta Q multzoen arteko biderketa kartesiarrak finkatuko du N AFED berriaren egoera multzoa:

$$S = R \times Q = \{s_0 = (r_0, q_0), s_1 = (r_0, q_1), s_2 = (r_1, q_0), s_3 = (r_1, q_1)\}$$

N AFED berriko s_0 hasierako egoera N_2 eta N_3 AFEDetako hasierako egoerez osatutako bikotea izango da: (r_0, q_0). Bestalde, N AFEDan onarpen egoerak N_2 AFEDan eta N_3 AFEDan onarpen egoerak direnez osatutako bikoteak izango dira. Formalki, $Y = Y_2 \times Y_3$. Beraz, N AFEDan onarpen egoera bakarra egongo da: (r_1, q_1).

N AFEDko trantsizioei dagokienez, (r_i, q_j) egoeratik (r_k, q_ℓ) egoerara A alfabetokoa den α sinboloarekin trantsizio bat egongo da baldin eta soilik baldin r_i egoeratik r_k egoerara α



3.3.36 irudia. a sinboloaz hasi eta a sinboloarekin bukatzen diren hitzen L lengoaiarentzat AFED bat.

sinboloarekin N_2 AFEDan trantsizio bat baldin badago eta q_j egoeratik q_ℓ egoerara α sinboloarekin N_3 AFEDan trantsizio bat baldin badago. Formalki, $\nu = \nu_2 \times \nu_3$. Hor, $\nu_2 \times \nu_3$ funtzioa $(Q \times R) \times A \rightarrow (Q \times R)$ motakoa izango da eta $\nu_2 \times \nu_3((q_i, r_j), \alpha) = \nu_2(q_i, \alpha) \times \nu_3(r_j, \alpha)$ izango da $Q \times R$ multzoko (q_i, r_j) erako bikote bakoitzeko eta A alfabetoko α sinbolo bakoitzeko.

3.3.36 irudian, a sinboloaz hasten diren eta a sinboloaz bukatzen diren hitzez osatutako L lengoaiarentzat eraiki den N AFEDari dagokion trantsizio-diagrama erakusten da. N AFEDa N_2 eta N_3 AFEDak erabiliz eta propietateen konjuntzioa dugunerako dagoen metodoa aplikatuz lortu da.

N AFEDa boskote bezala honela adieraz daiteke:

$$(S, A, \nu, s_0, Y)$$

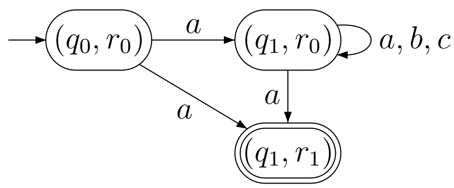
Boskote horretako osagaiak honela definituko lirateke:

- $S = R \times Q = \{(r_0, q_0), (r_0, q_1), (r_1, q_0), (r_1, q_1)\}$
- $A = \{a, b, c\}$
- $\nu : (R \times Q) \times A \rightarrow 2^{(R \times Q)}$ honako taula honen bidez definitutako trantsizio-funtzioa da:

ν	a	b	c
(r_0, q_0)	$\{(r_1, q_0), (r_1, q_1)\}$	$\varnothing$	$\varnothing$
(r_0, q_1)	$\varnothing$	$\varnothing$	$\varnothing$
(r_1, q_0)	$\{(r_1, q_0), (r_1, q_1)\}$	(r_1, q_0)	(r_1, q_0)
(r_1, q_1)	$\varnothing$	$\varnothing$	$\varnothing$

- $s_0 = (r_0, q_0)$ hasierako egoera da.
- $Y = Y_2 \times Y_3 = \{(r_1, q_1)\}$.

N'



3.3.37 irudia. a sinboloaz hasi eta a sinboloarekin bukatzen diren hitzen L lengoaiarentzat AFED bat (iristezinak diren egoerak kenduta).

(r_0, q_1) bikotea iristezina da (r_0, q_0) hasierako egoeratik. (r_0, q_0) egoeratik iristezinak diren egoerak ezabatu edo kendu egin behar dira.

3.3.37 irudian, a sinboloaz hasi eta a sinboloaz bukatzen diren hitzez eratutako lengoaiari dagokion N AFEDari iristezinak diren egoerak kendu ondoren geratzen den N' AFEDaren trantsizio-diagrama erakusten da. r_1 osagaia duten bikoteetan gaudenean, badakigu hitza a sinboloarekin hasi dela eta q_1 duten bikoteetan gaudenean, badakigu irakurri den azken sinboloa a izan dela. Hori dela-eta, bakarrik (r_1, q_1) egoeran bukatzen badugu erantzungo da “Bai”. Izan ere, egoera horretan gaudenean hitza a sinboloaz hasi da eta a sinboloaz bukatu da.

N' AFEDa honako boskote hau bezala formaliza daiteke:

$$(S', A, \nu', s'_0, Y')$$

Boskote horretako osagai bakoitza honela definituko litzateke:

- $S' = \{(r_0, q_0), (r_1, q_0), (r_1, q_1)\}$
- $A = \{a, b, c\}$
- $\nu' : S' \times A \to 2^{S'}$ honako taula honen bidez definitutako trantsizio-funtzioa da:

ν'	a	b	c
(r_0, q_0)	$\{(r_1, q_0), (r_1, q_1)\}$	$\varnothing$	$\varnothing$
(r_1, q_0)	$\{(r_1, q_0), (r_1, q_1)\}$	(r_1, q_0)	(r_1, q_0)
(r_1, q_1)	$\varnothing$	$\varnothing$	$\varnothing$

- $s'_0 = (r_0, q_0)$ hasierako egoera da.
- $Y = Y_2 \times Y_3 = \{(r_1, q_1)\}$.

3.3.3.4.3 a sinboloaz hasi eta b sinboloaz bukatzen diren hitzen lengoaiari dagokion AFED baten diseinua

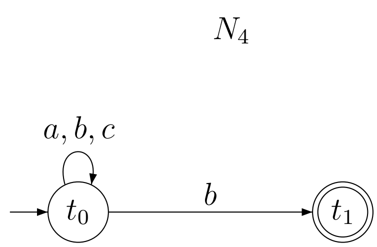
Har dezagun $A = \{a, b, c\}$ alfabetoa. a sinboloaz hasi eta b sinboloaz bukatzen diren hitzez eratutako L lengoaiarentzat AFED bat diseinatu nahi da. L lengoaia hori formalki honela defini daiteke:

$$L = \{w \mid w \in A^* \wedge \exists u(u \in A^* \wedge w = au) \wedge \exists u(u \in A^* \wedge w = ub)\}$$

L lengoaiako hitzek bi propietateren konjuntzioa den baldintza bat bete behar dute: a sinboloaz hastea eta b sinboloaz bukatzea. L lengoaiarentzat AFED bat eraikitzeko, aukera bat propietate horietako bakoitza kontuan hartuz bi AFED eraikitzea da. Beste era batera esanda, honako L_2 eta L_4 lengoaiak kontsidera ditzakegu:

$$L_2 = \{w \mid w \in A^* \wedge \exists u(u \in A^* \wedge w = au)\}$$
$$L_4 = \{w \mid w \in A^* \wedge \exists u(u \in A^* \wedge w = ub)\}$$

L lengoaia $L_2 \cap L_4$ da. L_2 lengoaiako hitzek betetzen duten propietatea a sinboloaz hastea da eta L_4 lengoaiako hitzek betetzen duten propietatea b sinboloaz bukatzea da. Hasteko, N_2 AFEDa eta N_4 AFEDa eraikiko ditugu, bata L_2 lengoaiarentzat eta bestea L_4 lengoaiarentzat, hurrenez hurren. 3.3.32 irudian N_2 AFEDari dagokion grafikoa erakusten da eta 3.3.38 irudian N_4 AFEDari dagokion grafikoa erakusten da.



3.3.38 irudia. b sinboloarekin bukatzen diren hitzen L_4 lengoaiarentzat AFED bat.

N_2 AFEDari dagokion (R, A, ν_2, r_0, Y_2) boskotea 3.3.3.4.1 atalean definitu da.
Bestalde, N_4 AFEDari dagokion boskotea honako hau izango litzateke:

$$(T, A, \nu_4, t_0, Y_4)$$

Boskote horretako osagaiak honela defini daitezke:

- $T = \{t_0, t_1\}$
- $A = \{a, b, c\}$
- $\nu_4 : T \times A \to 2^T$ honako taula honen bidez definitutako trantsizio-funtzioa da:

ν_4	a	b	c
t_0	$\{t_0\}$	$\{t_0, t_1\}$	$\{t_0\}$
t_1	$\varnothing$	$\varnothing$	$\varnothing$

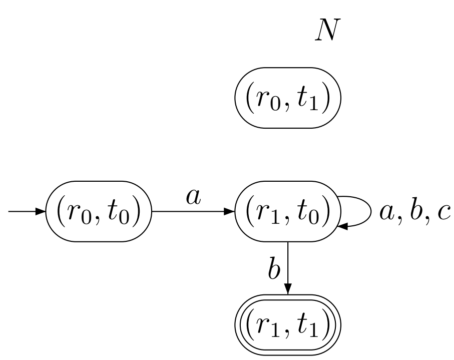
- t_0 hasierako egoera da.
- $Y_4 = \{t_1\}$.

L lengoaiarentzat $N = (S, A, \nu, s_0, Y)$ AFED bat eraikitzeko, hau da, a sinboloaz hasi eta b sinboloaz bukatzen diren hitzez osatutako L lengoaiarentzat AFED bat eraikitzeko, aukera bat N_2 eta N_4 erabiltzea da. Horretarako, N_2 AFEDko egoera bat eta N_4 AFEDko egoera bat hartuz osa daitezkeen bikote denak hartu beharko dira kontuan. N_2 AFEDko egoerak $R = \{r_0, r_1\}$ baldin badira eta N_4 AFEDko egoerak $T = \{t_0, t_1\}$ baldin badira, N AFEDari dagokion S egoera multzoa $R \times T$ izango da. Beraz, R eta T multzoen arteko biderketa kartesiarrak finkatuko du N AFED berriaren egoera-multzoa:

$$S = R \times T = \{s_0 = (r_0, t_0), s_1 = (r_0, t_1), s_2 = (r_1, t_0), s_3 = (r_1, t_1)\}$$

N AFED berriko s_0 hasierako egoera N_2 eta N_4 AFEDetako hasierako egoerez osatutako bikotea izango da: (r_0, t_0). Bestalde, N AFEDko onarpen egoerak N_2 AFEDan eta N_4 AFEDan onarpen egoerak direnez osatutako bikoteak izango dira. Formalki, $Y = Y_2 \times Y_4$. Beraz, N AFEDan onarpen egoera bakarra egongo da: (r_1, t_1).

N AFEDko trantsizioei dagokienez, (r_i, t_j) egoeratik (r_k, t_ℓ) egoerara A alfabetokoa den α sinboloarekin trantsizio bat egongo da baldin eta soilik baldin r_i egoeratik r_k egoerara α sinboloarekin N_2 AFEDan trantsizio bat baldin badago eta t_j egoeratik t_ℓ egoerara α sinboloarekin N_4 AFEDan trantsizio bat baldin badago. Formalki, $\nu = \nu_2 \times \nu_4$. Hor, $\nu_2 \times \nu_4$ funtzioa $(R \times T) \times A \to (R \times T)$ motakoa izango da eta $\nu_2 \times \nu_4((r_i, t_j), \alpha) = \nu_2(r_i, \alpha) \times \nu_4(t_j, \alpha)$ izango da $R \times T$ multzoko (r_i, t_j) erako bikote bakoitzeko eta A alfabetoko α sinbolo bakoitzeko.



3.3.39 irudia. a sinboloaz hasi eta b sinboloarekin bukatzen diren hitzen L lengoaiarentzat AFED bat.

3.3.39 irudian, a sinboloaz hasten diren eta b sinboloaz bukatzen diren hitzez osatutako L lengoaiarentzat eraiki den N AFEDari dagokion trantsizio-diagrama erakusten da. N AFEDa

N_2 eta N_4 AFEDak erabiliz eta propietateen konjuntzioa dugunerako dagoen metodoa aplikatuz lortu da.

N AFEDa boskote bezala honela adieraz daiteke:

$$(S, A, \nu, s_0, Y)$$

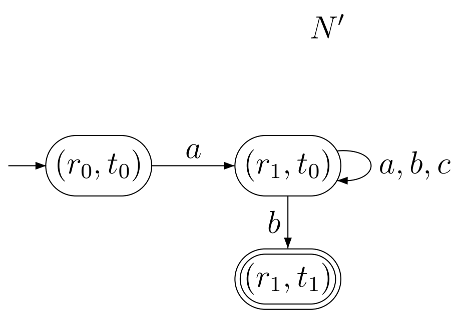
Boskote horretako osagaiak honela definituko lirateke:

- $S = R \times T = \{(r_0, t_0), (r_0, t_1), (r_1, t_0), (r_1, t_1)\}$
- $A = \{a, b, c\}$
- $\nu : (R \times T) \times A \rightarrow 2^{(R \times T)}$ honako taula honen bidez definitutako trantsizio-funtzioa da:

ν	a	b	c
(r_0, t_0)	$\{(r_1, t_0)\}$	$\varnothing$	$\varnothing$
(r_0, t_1)	$\varnothing$	$\varnothing$	$\varnothing$
(r_1, t_0)	$\{(r_1, t_0)\}$	$\{(r_1, t_0), (r_1, t_1)\}$	$\{(r_1, t_0)\}$
(r_1, t_1)	$\varnothing$	$\varnothing$	$\varnothing$

- $s_0 = (r_0, t_0)$ hasierako egoera da.
- $Y = Y_2 \times Y_4 = \{(r_1, t_1)\}$.

(r_0, t_1) bikotea iristezina da (r_0, t_0) hasierako egoeratik. (r_0, t_0) egoeratik iristezinak diren egoerak ezabatu edo kendu egin behar dira.



3.3.40 irudia. a sinboloaz hasi eta b sinboloarekin bukatzen diren hitzen L lengoaiarentzat AFED bat (iristezinak diren egoerak kenduta).

3.3.40 irudian, a sinboloaz hasi eta b sinboloaz bukatzen diren hitzez eratutako lengoaiari dagokion N AFEDari iristezinak diren egoerak kendu ondoren geratzen den N' AFEDaren trantsizio-diagrama erakusten da. r_1 osagaia duten bikoteetan gaudenean, badakigu hitza a sinboloarekin hasi dela eta t_1 duten bikoteetan gaudenean, badakigu irakurri den azken sinboloa b izan dela. Hori dela-eta, bakarrik (r_1, t_1) egoeran bukatzen badugu erantzungo da “Bai”. Izan ere, egoera horretan gaudenean hitza a sinboloaz hasi da eta b sinboloaz bukatu da.

N' AFEDa honako boskote hau dezala adieraz daiteke:

$$(S', A, \nu', s'_0, Y')$$

Boskote horretako osagaiak honela defini daitezke:

- $S' = \{(r_0, t_0), (r_1, t_0), (r_1, t_1)\}$
- $A = \{a, b, c\}$
- $\nu' : S' \times A \to 2^{S'}$ honako taula honen bidez definitutako trantsizio-funtzioa da:

ν'	a	b	c
(r_0, t_0)	$\{(r_1, t_0)\}$	$\varnothing$	$\varnothing$
(r_1, t_0)	$\{(r_1, t_0)\}$	$\{(r_1, t_0), (r_1, t_1)\}$	$\{(r_1, t_0)\}$
(r_1, t_1)	$\varnothing$	$\varnothing$	$\varnothing$

- (r_0, t_0) hasierako egoera da.
- $Y = \{(r_1, t_1)\}$.

3.3.3.5 AFEDen diseinua propietateen disjuntzioarentzat (Lengoaien bildura)

Konjuntzioa dugunerako, aurreko atalean (3.3.3.4 atala) aurkeztu den teknikak bai AFDentzat (ikus 3.2.2.4 atala) eta bai AFEDentzat erabil daiteke. AFDen kasuan, teknika hori bera propietateen disjuntzioa dugunean ere erabil daitekeela erakutsi da 3.2.2.5 atalean. Hori horrela da AFDen kasuan bai konjuntzioarentzat eta bai ukapenarentzat (ikus 3.2.2.3 atala) metodo sistematikoa dugulako eta H_1 eta H_2 bi lengoaien arteko bildura (disjuntzioa) ukapena (osagarria) eta konjuntzioa (ebakidura) erabiliz defini daitekeelako: $H_1 \cup H_2 = \overline{\overline{H_1} \cap \overline{H_2}}$. Baina AFEDen kasuan ukapena daukagunean metodoak ez duenez funtzionatzen, disjuntzioaren kasuan ere metodoak ez du funtzionatzen.

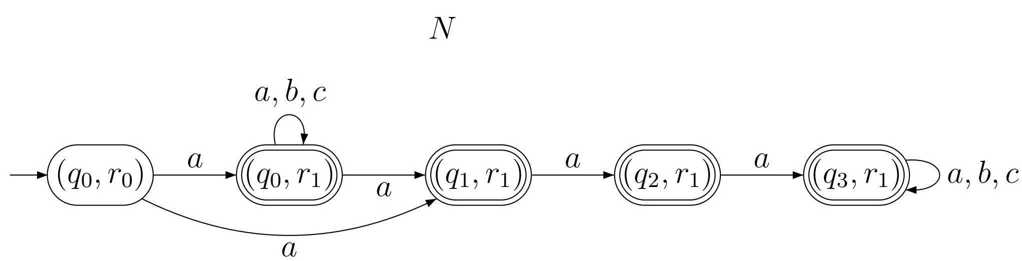
Hasteko, konjuntzioaren metodoak AFEDetan disjuntzioarentzat ez duela balio erakusten duen adibide bat aurkeztuko da. Gero, disjuntzioa dugunean AFEDa eraikitzeko metodo sistematikoa emango da. Bukatzeko, metodo sistematiko hori nola erabili erakusten duten bost adibide azalduko dira.

3.3.3.5.1 aaa katea duten edo a sinboloaz hasten diren hitzez eratutako lengoaiarentzat AFED baten arrakastarik gabeko diseinua

Har dezagun $A = \{a, b, c\}$ alfabetoa. aaa katea gutxienez behin duten edo a sinboloaz hasten diren hitzez eratutako L' lengoaiarentzat AFED bat diseinatu nahi da. L' lengoaia hori formalki honela defini daiteke:

$$L' = \{w \mid w \in A^* \wedge (\exists u, v(u \in A^* \wedge v \in A^* \wedge w = uaaav) \vee \exists u(u \in A^* \wedge w = au))\}$$

L' lengoaiako hitzek bi propietateren disjuntzioa den baldintza bat bete behar dute: gutxienez aaa bat edukitzea edo a sinboloaz hastea. AFDen kasuan, aukera bat, lehenengo urrats bezala propietate horietako bakoitza kontuan hartuz bi AFD eraikitzea da eta gero, bigarren urrats bezala, konjuntzioaren kasuan egiten dena egitea, hau da, propietate horien disjuntzioa betetzen duten hitzen lengoaiari dagokion AFDa eraikitzea egoera bezala hasierako bi AFDetako egoerez eratutako bikoteak hartuz eta trantsizioak finkatuz. Desberdintasun bakarra zirkulu bikoitza izango duten egoeretan egongo da. AFEDetan ere estrategia bera jarraitzen badugu eta 3.3.3.4.1 atalean azaldutako N_1 eta N_2 AFEDak erabiltzen baditugu (ikus 3.3.31 eta 3.3.32 irudiak), 3.3.41 irudian agertzen den trantsizio-diagrama duen AFEDa lortuko genuke. Diagrama horretan osagai bezala q_3 edo r_1 duten egoerek zirkulu bikoitza dute. Baina AFED hori ez da egokia L' lengoaiarentzat. Izan ere, "Ez" erantzungo du $bcaaabb$ erako hitzentzat, "Bai" erantzun beharrean. Hain zuzen ere, AFED horrek ezezko erantzuna emango du $bcaaabb$ hitzarentzat (q_0, r_0) egoeran b sinboloarentzat ez dagoelako trantsiziorik.



3.3.41 irudia. aaa duten edo a sinboloaz hasten diren hitzez osatutako L' lengoaiarentzat huts egiten duen AFED bat.

3.3.3.5.2 AFEDen diseinu sistematikoa disjuntzioa dugunerako

Demagun L_1 eta L_2 lengoaiaen bildura den L lengoaiarentzat AFED bat diseinatu nahi dugula. Beraz, $L = L_1 \cup L_2$. Demagun baita ere L_1 eta L_2 lengoaientzat $N_1 = (Q, A, \nu_1, q_0, Y_1)$ eta $N_2 = (R, A, \nu_2, r_0, Y_2)$ AFEDak eraikita ditugula. L lengoaiari dagokion N AFEDa (S, A, ν, s_0, Y) erako boskote gisa formaliza daiteke. Boskote horretako osagaien definizioa honako hau izango litzateke:

- $S = Q \cup R \cup \{z_0\}$. Hor, z_0 egoera Q multzoan eta R multzoan agertzen ez den egoera berri bat izango da.
- A alfabetoa N_1 eta N_2 AFEDetako alfabeto bera da.
- s_0 hasierako egoera z_0 izango da.

- Onarpen egoeren Y multzoari dagokionez:
  - $q_0 \in Y_1$ edo $r_0 \in Y_2$ betetzen bada, orduan $Y = Y_1 \cup Y_2 \cup \{z_0\}$.
  - $q_0 \notin Y_1$ eta $r_0 \notin Y_2$ betetzen bada, orduan $Y = Y_1 \cup Y_2$.

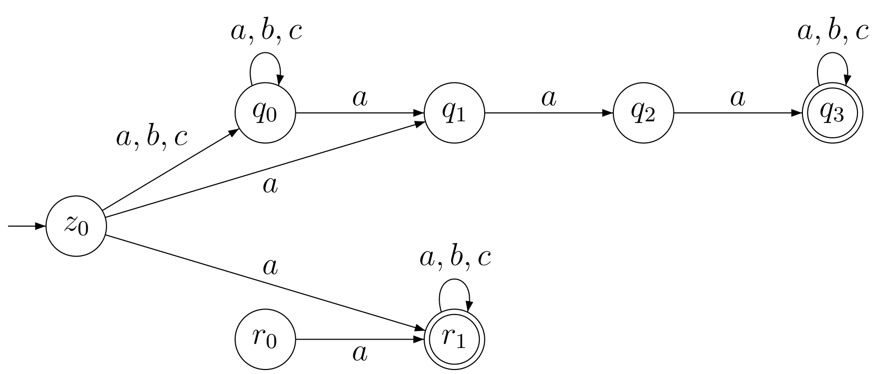
  Beraz, z_0 egoera onarpen egoera izango da baldin eta soilik baldin q_0 egoera edo s_0 egoera onarpen egoera baldin bada.

- Azkenik, ν trantsizio-funtzioa $S \times A \to 2^S$ motakoa izango da eta honako hau beteko da:
  - $\nu(q_i, \alpha) = \nu_1(q_i, \alpha)$, edozein $q_i \in Q$ eta edozein $\alpha \in A$ hartuta.
  - $\nu(r_j, \alpha) = \nu_2(r_j, \alpha)$, edozein $r_j \in R$ eta edozein $\alpha \in A$ hartuta.
  - $\nu(z_0, \alpha) = \nu_1(q_0, \alpha) \cup \nu_2(r_0, \alpha)$, edozein $\alpha \in A$ hartuta.

  Beraz, q_0-tik eta r_0-tik ateratzen den trantsizio bakoitzeko, z_0-tik ere trantsizio bat aterako da sinbolo berarekin eta norako berarekin.

3.3.3.5.3 aaa duten edo a sinboloaz hasten diren hitzez osatutako lengoaiarentzat AFED baten diseinu sistematikoa

aaa azpikatea duten edo a sinboloaz hasten diren hitzez osatutako L lengoaiari dagokion AFEDa diseinatzeko, aaa duten hitzez osatutako L_1 lengoaiari eta a-z hasten diren hitzez osatutako L_2 lengoaiari dagozkien AFEDak har ditzakegu abiapuntutzat eta 3.3.3.5.2 atalean aurkeztutako metodo sistematikoa aplika dezakegu. L_1 lengoaiari dagokion N_1 AFEDaren trantsizio-diagrama 3.3.7 irudian dugu eta L_2 lengoaiari dagokion N_2 AFEDaren trantsizio-diagrama 3.3.12 irudian dugu (3.3.3.4.1 atala).



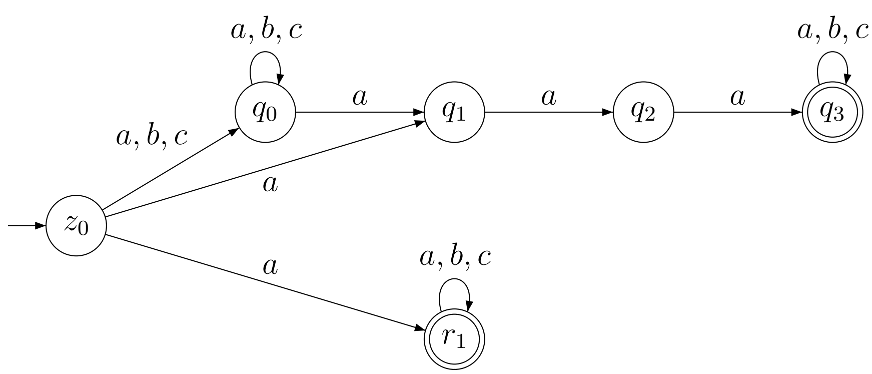
3.3.42 irudia. aaa duten edo a-z hasten diren hitzez osatutako lengoaiarentzat AFED bat.

3.3.42 irudian metodo sistematikoa aplikatuz lortuko litzatekeen N AFEDaren trantsizio-diagrama dugu. N AFED hori (S, A, ν, s_0, Y) erako boskote gisa formaliza daite. Boskote horretako osagaien definizioa honako hau da:

- $S = Q \cup R \cup \{z_0\} = \{q_0, q_1, q_2, q_3, r_0, r_1, z_0\}$.
- A alfabetoa N_1 eta N_2 AFEDetako alfabeto bera da. Beraz, $A = \{a, b, c\}$.
- s_0 hasierako egoera z_0 da.
- Onarpen egoeren Y multzoari dagokionez:
  - $q_0 \notin Y_1$ eta $r_0 \notin Y_2$ betetzen denez, $Y = Y_1 \cup Y_2 = \{q_3, r_1\}$.
- Azkenik, ν trantsizio-funtzioa $S \times A \rightarrow 2^S$ motakoa da eta honako trantsizio-taularen bidez defini daiteke:

ν	a	b	c
z_0	$\{q_0, q_1, r_1\}$	$\{q_0\}$	$\{q_0\}$
q_0	$\{q_0, q_1\}$	$\{q_0\}$	$\{q_0\}$
q_1	$\{q_2\}$	$\varnothing$	$\varnothing$
q_2	$\{q_3\}$	$\varnothing$	$\varnothing$
q_3	$\{q_3\}$	$\{q_3\}$	$\{q_3\}$
r_0	$\{r_1\}$	$\varnothing$	$\varnothing$
r_1	$\{r_1\}$	$\{r_1\}$	$\{r_1\}$

3.3.42 irudiko trantsizio-diagraman ikus daitekeen gisa, q_0 eta r_0 egoeretatik irtetzen den trantsizio bakoitzeko, z_0-tik ere trantsizio bat izango dugu. Bestalde, r_0 iristezina da z_0 egoeratik eta ken daiteke.



3.3.43 irudia. aaa duten edo a-z hasten diren hitzen lengoaiari dagokion AFEDa (iristezinak diren egoerarik gabe).

3.3.43 irudian iristezina den r_0 egoera kendu ondoren gelditzen den AFEDaren trantsizio-diagrama dugu. Ezabaketa hori egin ondoren, ez ditugu abiapuntuko AFED biak era garbian eta, ondorioz, AFEDen bilduraren modularitatea eta argitasuna galdu egiten dira.

3.3.3.5.4 a sinboloaz hasi edo a sinboloaz bukatzen diren hitzen lengoaiari dagokion AFED baten diseinu sistematikoa

Har dezagun $A = \{a, b, c\}$ alfabetoa. a sinboloaz hasi eta a sinboloaz bukatzen diren hitzez eratutako L lengoaiarentzat AFED bat diseinatu nahi da. Hori egiteko, a sinboloaz hasten diren hitzen L_2 lengoaiari dagokion AFED bat eta a sinboloarekin bukatzen diren hitzen L_3 lengoaiari dagokion AFED bat har ditzakegu abiapuntu bezala eta 3.3.3.5.2 atalean aurkeztutako metodo sistematikoa aplika dezakegu.

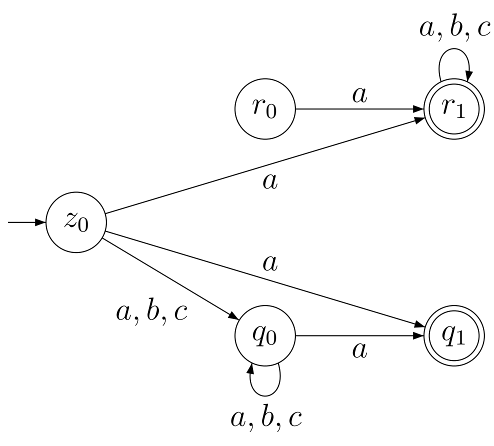
L lengoaia hori formalki honela defini daiteke:

$$L = \{w \mid w \in A^* \wedge (\exists u(u \in A^* \wedge w = au) \vee \exists u(u \in A^* \wedge w = ua))\}$$

Bestalde, L_2 eta L_3 honela defini daitezke:

$$L_2 = \{w \mid w \in A^* \wedge \exists u(u \in A^* \wedge w = au)\}$$
$$L_3 = \{w \mid w \in A^* \wedge \exists u(u \in A^* \wedge w = ua)\}$$

L_2 lengoaiako hitzek betetzen duten propietatea a sinboloaz hastea da eta L_3 lengoaiako hitzek betetzen duten propietatea a sinboloaz bukatzea da. L lengoaia $L_2 \cup L_3$ da. Hasteko, N_2 AFEDa eta N_3 AFEDa eraikiko ditugu, bata L_2 lengoaiarentzat eta bestea L_3 lengoaiarentzat, hurrenez hurren. 3.3.32 irudian N_2 AFEDari dagokion grafikoa erakusten da eta 3.3.35 irudian N_3 AFEDari dagokion grafikoa erakusten da. N_2 AFEDari dagokion boskotea 3.3.3.4.1 atalean definitu da eta N_3 AFEDari dagokion boskotea 3.3.3.4.2 atalean definitu da.



3.3.44 irudia. a-rekin hasi edo a-rekin bukatzen diren hitzen lengoaiari dagkion AFEDa.

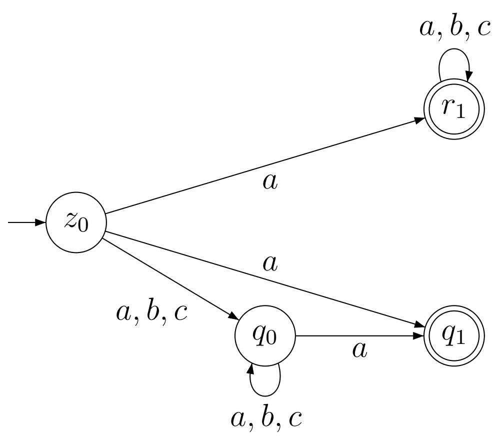
3.3.44 irudian a-rekin hasi edo a-rekin bukatzen diren hitzen lengoaiari dagokion N AFE-Daren trantsizio-diagrama dugu. N AFED hori N_2 eta N_3 AFEDetatik abiatuta eraiki da. N AFEDa (S, A, ν, s_0, Y) boskote gisa formaliza daiteke. Boskote horretako osagaien definizioa honako hau da:

- $S = R \cup Q \cup \{z_0\} = \{r_0, r_1, q_0, q_1, z_0\}$.

- A alfabetoa N_2 eta N_3 AFEDetako alfabeto bera da. Beraz, $A = \{a, b, c\}$.
- s_0 hasierako egoera z_0 da.
- Onarpen egoeren Y multzoari dagokionez:
  - $r_0 \notin Y_2$ eta $q_0 \notin Y_3$ betetzen denez, $Y = Y_2 \cup Y_3 = \{r_1, q_1\}$.
- Azkenik, ν trantsizio-funtzioaren mota $S \times A \rightarrow 2^S$ da eta honako trantsizio-taularen bidez defini daiteke:

ν	a	b	c
z_0	$\{r_1, q_0, q_1\}$	$\{q_0\}$	$\{q_0\}$
r_0	$\{r_1\}$	$\varnothing$	$\varnothing$
r_1	$\{r_1\}$	$\{r_1\}$	$\{r_1\}$
q_0	$\{q_0\}$	$\{q_0\}$	$\{q_0\}$
q_1	$\varnothing$	$\varnothing$	$\varnothing$

3.3.44 irudiko trantsizio-diagraman ikus daitekeen gisa, r_0 eta q_0 egoeretatik ateratzen den trantsizio bakoitzeko, z_0-tik ere trantsizio bat aterako da. Bestalde, r_0 iristezina da z_0-tik eta ezaba daiteke.



3.3.45 irudia. a-rekin hasi edo a-rekin bukatzen diren hitzen lengoaiari dagkion AFEDa (iristezinak diren egoerarik gabe).

3.3.45 irudian iristezina den r_0 egoera kendu ondoren gelditzen den AFEDaren trantsizio-diagrama dugu. Ezabaketa hori egin ondoren, ez ditugu abiapuntuko AFED biak era garbian eta, ondorioz, AFEDen bilduraren modularitatea eta argitasuna galdu egiten dira.

3.3.3.5.5 a sinboloaz hasi edo b sinboloaz bukatzen diren hitzen lengoaiari dagokion AFED baten diseinu sistematikoa

Har dezagun $A = \{a, b, c\}$ alfabetoa. a sinboloaz hasi edo b sinboloaz bukatzen diren hitzen L lengoaiari dagokion AFED bat diseinatu nahi da. Horretarako, abiapuntutzat a sinboloaz hasten diren hitzen L_2 lengoaiari dagokion AFED bat eta b sinboloaz bukatzen diren hitzen L_4 lengoaiari dagokion AFED bat hartuko ditugu.

L lengoaia hori formalki honela defini daiteke:

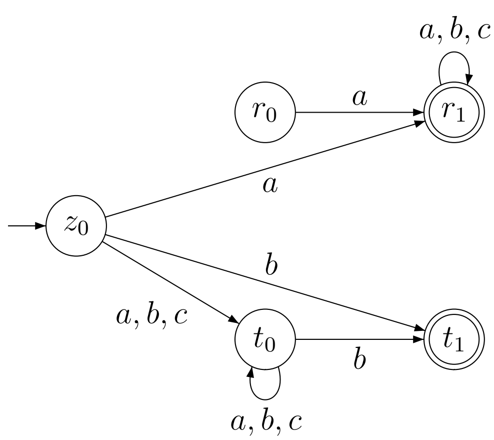
$$L = \{w \mid w \in A^* \wedge (\exists u(u \in A^* \wedge w = au) \vee \exists u(u \in A^* \wedge w = ub))\}$$

Bestalde, L_2 eta L_4 lengoaiak honela formaliza daitezke:

$$L_2 = \{w \mid w \in A^* \wedge \exists u(u \in A^* \wedge w = au)\}$$
$$L_4 = \{w \mid w \in A^* \wedge \exists u(u \in A^* \wedge w = ub)\}$$

L_2 lengoaiako hitzek betetzen duten propietatea a sinboloaz hastea da eta L_4 lengoaiako hitzek betetzen duten propietatea b sinboloaz bukatzea da. L lengoaia $L_2 \cup L_4$ da. Hasteko, N_2 AFEDa eta N_4 AFEDa eraikiko ditugu, bata L_2 lengoaiarentzat eta bestea L_4 lengoaiarentzat, hurrenez hurren. 3.3.32 irudian N_2 AFEDari dagokion grafikoa erakusten da eta 3.3.38 irudian N_4 AFEDari dagokion grafikoa erakusten da.

N_2 AFEDari dagokion boskotea 3.3.3.4.1 atalean definitu da eta N_4 AFEDari dagokion boskotea 3.3.3.4.3 atalean definitu da.



3.3.46 irudia. a-rekin hasi edo b-rekin bukatzen diren hitzen lengoaiari dagokion AFED bat.

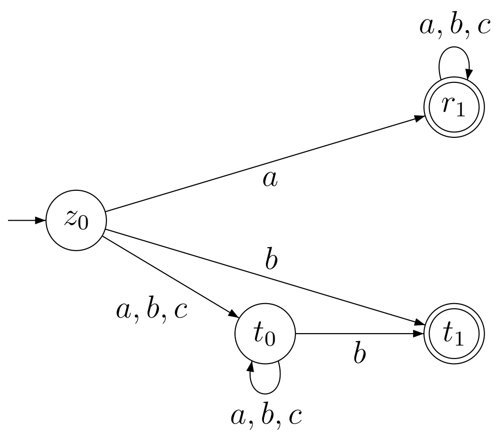
a sinboloaz hasi edo b sinboloaz bukatzen diren hitzen lengoaiarentzat diseinatu den N AFEDaren trantsizio-diagrama 3.3.46 irudian dugu. N AFED hori N_2 eta N_4 AFEDak elkartuz eraiki da 3.3.3.5.2 atalean aurkeztu den metodo sistematikoari jarraituz.

N AFEDa (S, A, ν, s_0, Y) erako boskote gisa adieraz daiteke. Boskote horretako osagaiak honela definituko genituzke:

- $S = R \cup T \cup \{z_0\} = \{r_0, r_1, t_0, t_1, z_0\}$.
- A alfabetoa N_2 eta N_4 AFEDetako alfabeto bera da. Beraz, $A = \{a, b, c\}$.
- s_0 hasierako egoera z_0 da.
- Onarpen egoeren Y multzoari dagokionez:
  - $r_0 \notin Y_2$ eta $t_0 \notin Y_4$ betetzen denez, $Y = Y_2 \cup Y_4 = \{r_1, t_1\}$.
- Azkenik, ν trantsizio-funtzioa $S \times A \rightarrow 2^S$ motakoa izango da eta honako trantsizio-taularen bidez definituko genuke:

ν	a	b	c
z_0	$\{r_1, t_0\}$	$\{t_0, t_1\}$	$\{t_0\}$
r_0	$\{r_1\}$	$\varnothing$	$\varnothing$
r_1	$\{r_1\}$	$\{r_1\}$	$\{r_1\}$
t_0	$\{t_0\}$	$\{t_0, t_1\}$	$\{t_0\}$
t_1	$\varnothing$	$\varnothing$	$\varnothing$

3.3.46 irudiko trantsizio-diagraman ikus daitekeen gisa, r_0 eta t_0 egoeretatik irtetzen den trantsizio bakoitzeko, z_0-tik ere trantsizio bat dugu. Bestalde, r_0 egoera z_0 egoeratik iristezina da eta ezaba daiteke.



3.3.47 irudia. a-rekin hasi edo b-rekin bukatzen diren hitzen lengoaiari dagokion AFED bat (iristezinak diren egoerarik gabe).

3.3.47 irudian, iristezina den r_0 egoera ezabatu ondoren gelditzen den AFEDaren trantsizio-diagrama erakusten da. Era honetako ezabaketak egindakoan hasierako bi AFEDak ez dira garbi ikusten eta, ondorioz, modularitatea eta garbitasuna galdu egiten dira.

3.3.3.5.6 a sinboloaz edo b sinboloaz bukatzen diren hitzen lengoaiari dagokion AFED baten diseinu sistematikoa

Har dezagun $A = \{a, b, c\}$ alfabetoa. a sinboloaz edo b sinboloaz bukatzen diren hitzen L lengoaiari dagokion AFED bat diseinatu nahi da. Horretarako, abiapuntutzat a sinboloaz bukatzen diren hitzen L_3 lengoaiari dagokion AFED bat eta b sinboloaz bukatzen diren hitzen L_4 lengoaiari dagokion AFED bat hartuko ditugu.

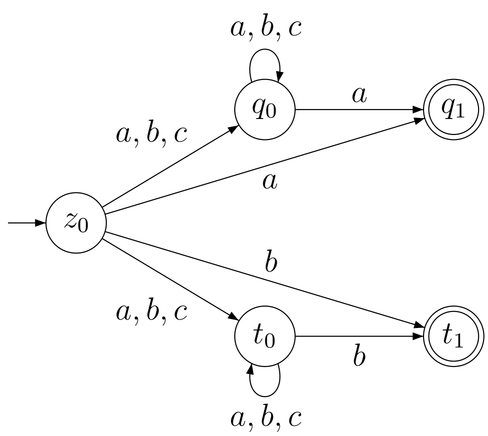
L lengoaia hori formalki honela defini daiteke:

$$L = \{w \mid w \in A^* \wedge (\exists u(u \in A^* \wedge w = ua) \vee \exists u(u \in A^* \wedge w = ub))\}$$

L_3 eta L_4 lengoaiei dagokienez, honela formaliza daitezke:

$$L_3 = \{w \mid w \in A^* \wedge \exists u(u \in A^* \wedge w = ua)\}$$
$$L_4 = \{w \mid w \in A^* \wedge \exists u(u \in A^* \wedge w = ub)\}$$

L_3 lengoaiako hitzek betetzen duten propietatea a sinboloaz bukatzea da eta L_4 lengoaiako hitzek betetzen duten propietatea b sinboloaz bukatzea da. L lengoaia $L_3 \cup L_4$ da. Hasteko, N_3 AFEDa eta N_4 AFEDa eraikiko ditugu, bata L_3 lengoaiarentzat eta bestea L_4 lengoaiarentzat, hurrenez hurren. 3.3.35 irudian N_3 AFEDari dagokion grafikoa erakusten da eta 3.3.38 irudian N_4 AFEDari dagokion grafikoa erakusten da. N_3 AFEDari dagokion boskotea 3.3.3.4.2 atalean definitu da eta N_4 AFEDari dagokion boskotea 3.3.3.4.3 atalean definitu da.



3.3.48 irudia. a-rekin edo b-rekin bukatzen diren hitzen lengoaiarentzat AFED bat.

a sinboloaz edo b sinboloaz bukatzen diren hitzen lengoaiarentzat diseinatu den N AFEDaren trantsizio-diagrama 3.3.48 irudian dugu. N AFED hori N_3 eta N_4 AFEDak elkartuz eraiki da. Horretarako 3.3.3.5.2 ataleko metodo sistematikoa erabili da. N AFEDa (S, A, ν, s_0, Y) erako boskote gisa adieraz daiteke. Boskote horretako osagaiak honela definituko genituzke:

- $S = Q \cup T \cup \{z_0\} = \{q_0, q_1, t_0, t_1, z_0\}$.
- A alfabetoa N_3 yeta N_4 AFEDetako alfabeto bera da. Beraz, $A = \{a, b, c\}$.

- s_0 hasierako egoera z_0 da.
- Onarpen egoeren Y multzoari dagokionez:
  - $q_0 \notin Y_3$ eta $t_0 \notin Y_4$ betetzen denez, $Y = Y_3 \cup Y_4 = \{q_1, t_1\}$.
- Azkenik, ν trantsizio-funtzioa $S \times A \to 2^S$ erakoa da eta honako trantsizio-taularen bidez defini daitekede:

ν	a	b	c
z_0	$\{q_0, q_1, t_0\}$	$\{q_0, t_0, t_1\}$	$\{q_0, t_0\}$
q_0	$\{q_0, q_1\}$	$\{q_0\}$	$\{q_0\}$
q_1	$\varnothing$	$\varnothing$	$\varnothing$
t_0	$\{t_0\}$	$\{t_0, t_1\}$	$\{t_0\}$
t_1	$\varnothing$	$\varnothing$	$\varnothing$

3.3.48 irudian ikus daitekeenez, q_0 eta t_0 egoeretatik irtetzen den trantsizio bakoitzeko, z_0 egoeratik ere trantsizio bat ateratzen da.

3.3.3.5.7 a-rekin hasi edo a kopuru bikoitia duten hitzen lengoaiari dagokion AFED baten diseinu sistematikoa

Har dezagun $A = \{a, b, c\}$ alfabetoa. a-rekin hasi edo a kopuru bikoitia duten hitzen lengoaiari dagokion AFED bat diseinatu nahi da. Horretarako, a-rekin hasten diren hitzez eratutako L_2 lengoaia eta a kopuru bikoitia duten hitzez eratutako L_5 lengoaia har ditzakegu eta 3.3.3.5.2 atalean aurkeztutako metodo sistematikoa aplika dezakegu.

L lengoaia honela adieraz daiteke formalki:

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

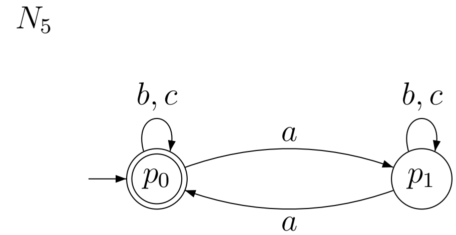
L_2 eta L_5 lengoaiak honela adieraz daitezke era formalean:

$$L_2 = \{w \mid w \in A^* \wedge \exists u(u \in A^* \wedge w = au)\}$$
$$L_5 = \{w \mid w \in A^* \wedge |w|_a \text{ mod } 2 = 0\}$$

L_2 lengoaiako hitzek betetzen duten propietatea a-rekin hastea da eta L_5 lengoaiako hitzek betetzen duten propietatea a kopuru bikoitia edukitzea da. Ondorioz, L lengoaia $L_2 \cup L_5$ da. Hasteko, L_2 lengoaiarentzat N_2 AFED bat eta L_5 lengoaiarentzat N_5 AFED bat diseinatuko genituzke. 3.3.32 irudian N_2 AFEDaren trantsizio-diagrama dugu eta 3.3.49 irudian N_5 AFEDarentzat trantsizio-diagrama dugu.

N_2 AFEDari dagokion boskotea 3.3.3.4.1 atalean definitu da eta N_5 AFEDari dagokion boskotea honela formaliza daiteke:

$$(P, A, \nu_5, p_0, Y_5)$$



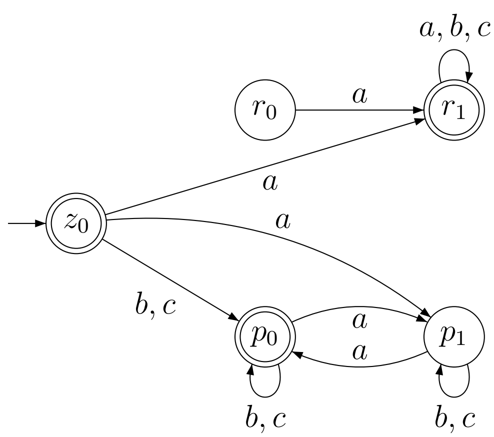
3.3.49 irudia. a kopuru bikoitia duten hitzen L_5 lengoaiari dagokion AFED bat.

Boskote horretan:

- $P = \{p_0, p_1\}$
- $A = \{a, b, c\}$
- $\nu_5 : P \times A \to 2^P$ honako taula honen bidez definitutako trantsizio-funtzioa da:

ν_5	a	b	c
p_0	$\{p_1\}$	$\{p_0\}$	$\{p_0\}$
p_1	$\{p_0\}$	$\{p_1\}$	$\{p_1\}$

- p_0 hasierako egoera da.
- $Y_5 = \{p_0\}$.



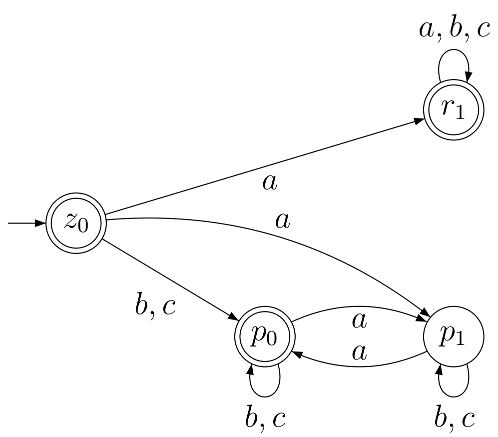
3.3.50 irudia. a-rekin hasi edo a kopuru bikoitia duten hitzen lengoaiarentzat AFED bat.

3.3.50 irudian a-rekin hasi edo a kopuru bikoitia duten L lengoaiari dagokion N AFEDaren trantsizio-diagrama dugu. AFED hori N_2 eta N_5 AFEDei 3.3.3.5.2 ataleko metodo sistematikoa aplikatuz lortu da. N AFED hori (S, A, ν, s_0, Y) boskote gisa formaliza daiteke. Boskote horren osagaiak honela defini daitezke:

- $S = R \cup P \cup \{z_0\} = \{r_0, r_1, p_0, p_1, z_0\}$.
- A alfabeto N_2 eta N_5 AFEDetako alfabeto bera da. Beraz, $A = \{a, b, c\}$.
- s_0 hasierako egoera z_0 da.
- Onarpen egoeren Y multzoari dagokionez:
  - $r_0 \in Y_2$ edo $p_0 \in Y_5$ betetzen denez, $Y = Y_2 \cup Y_5 \cup \{z_0\} = \{r_1, p_0, z_0\}$.
- Azkenik, ν trantsizio-funtzioa $S \times A \to 2^S$ erakoa da eta honako taularen bidez defini daiteke:

ν	a	b	c
z_0	$\{r_1, q_1\}$	$\{p_0\}$	$\{p_0\}$
r_0	$\{r_1\}$	$\varnothing$	$\varnothing$
r_1	$\{r_1\}$	$\{r_1\}$	$\{r_1\}$
p_0	$\{p_1\}$	$\{p_0\}$	$\{p_0\}$
p_1	$\{p_0\}$	$\{p_1\}$	$\{p_1\}$

3.3.50 irudiko trantsizio-diagraman ikus daitekeenez, r_0 eta q_0 egoeretatik ateratzen den trantsizio bakoitzeko, z_0-tik ere trantsizio bat ateratzen da. Adibide honetan z_0 egoerak bi zirkulu ditu p_0 egoerak ere bi zirkulu dituelako. Bestalde, r_0 iristezina da z_0 egoeratik eta ezaba daiteke.



3.3.51 irudia. a-rekin hasi edo a kopuru bikoitia duten hitzen lengoaiari dagokion AFED bat (iristezinak diren egoerarik gabe).

3.3.51 irudian r_0 egoera ezabatu ondoren lortzen den AFEDaren trantsizio-diagrama dugu. Ezabaketa horiek direla-eta, hasierako AFEDak ez daude hor era garbian eta, ondorioz, modularitatea eta argitasuna galdu egiten dira.

3.3.3.6 Lengoaien kateaketari dagokion AFEDaren diseinua

Atal honetan, bi lengoairen kateaketa den lengoaia baten AFEDa bi lengoaia horiei dagozkien AFEDetatik abiatuz eta metoko sistematiko bat aplikatuz nola kalkula daitekeen azalduko da.

3.3.3.6.1 Lengoaien kateaketari dagokion AFEDaren diseinu sistematikoa

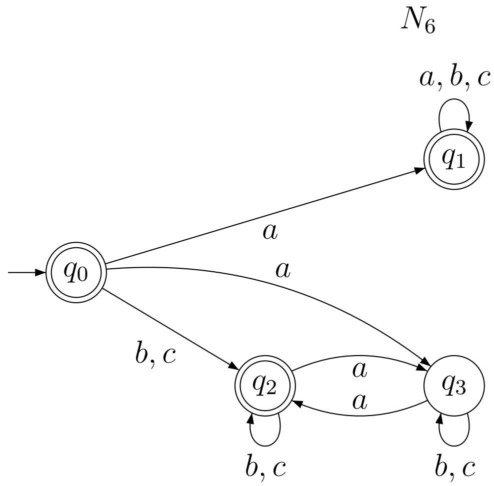
Demagun L lengoaia batentzat AFED bat diseinatu nahi dugula eta L lengoaia hori L_1 eta L_2 lengoaien kateadura dela. Beraz, $L = L_1L_2$. Demagun, baita ere, L_1 eta L_2 lengoaientzat $N_1 = (Q, A, \nu_1, q_0, Y_1)$ eta $N_2 = (R, A, \nu_2, r_0, Y_2)$ AFEDak diseinatuta ditugula. L-ri dagokion N AFEDa (S, A, ν, s_0, Y) erako boskote gisa formula daiteke. Boskote horretako osagaiak honela defini daitezke:

- $S = Q \cup R$.
- A alfabetoa N_1 eta N_2 AFEDetako alfabeto bera da.
- s_0 hasierako egoera q_0 da.
- Onarpen egoeren Y multzoari dagokionez:
  - $r_0 \in Y_2$ betetzen bada, orduan $Y = Y_1 \cup Y_2$.
  - $r_0 \notin Y_2$ betetzen bada, orduan $Y = Y_2$.
- Azkenik, ν trantsizio-funtzioa $S \times A \to 2^S$ erakoa izango da eta honela defini daiteke:
  - $\nu(q_i, \alpha) = \nu_1(q_i, \alpha)$, edozein $q_i \in Q \setminus Y_1$ egoerarentzat eta edozein $\alpha \in A$ sinbolorentzat.
  - $\nu(q_i, \alpha) = \nu_1(q_i, \alpha) \cup \nu_2(r_0, \alpha)$, edozein $q_i \in Y_1$ egoerarentzat eta edozein $\alpha \in A$ sinbolorentzat.
  - $\nu(r_j, \alpha) = \nu_2(r_j, \alpha)$, edozein $r_j \in R$ egoerarentzat eta edozein $\alpha \in A$ sinbolorentzat.

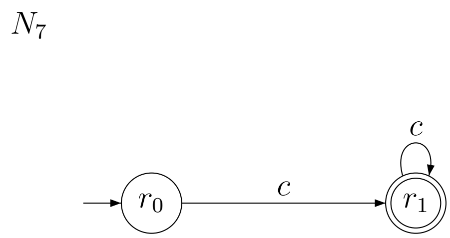
3.3.3.6.2 a-rekin hasi edo a kopuru bikoitia izan eta c sinboloaz osatutako bloke ez-huts batez bukatzen diren hitzen lengoaiari dagokion AFEDaren diseinu sistematikoa

Har dezagun $A = \{a, b, c\}$ alfabetoa. a-rekin hasi edo a kopuru bikoitia izan eta c sinboloaz osatutako bloke ez-huts batez bukatzen diren hitzen L lengoaiari dagokion AFED bat diseinatu nahi da. Horretarako, a-rekin hasi edo a kopuru bikoitia duten hitzen L_6 lengoaiari dagokion AFED bat eta c sinboloaz osatutako bloke ez-huts batez bukatzen diren hitzen L_7 lengoaiari dagokion AFED bat har ditzakegu abiapuntutzat. L-ri dagokion AFEDa eraikitzeko 3.3.3.6.1 atalean azaldutako metodo sistematikoa aplikatuko da.

L lengoaia L_6L_7 eran defini daiteke. L_6 eta L_7 honela adieraz daitezke:



3.3.52 irudia. a-rekin hasi edo a kopuru bikoitia duten hitzen L_6 lengoaiarentzat N_6 AFEDa.



3.3.53 irudia. Bakarrik c-ren errepikapenez osatuta dauden hitzen L_7 lengoaiari dagokion AFED bat.

$$L_6 = \{w \mid w \in A^* \wedge (\exists u(u \in A^* \wedge w = au) \vee (|w|_a \bmod 2 = 0))\}$$
$$L_7 = \{w \mid w \in A^* \wedge |w| \geq 1 \wedge |w| = |w|_c\}$$

3.3.52 irudian L_6 lengoaiari dagokion N_6 AFED baten trantsizio-diagrama erakusten da eta 3.3.53 irudian L_7 lengoaiari dagokion N_7 AFED baten trantsizio-diagrama erakusten da.

N_6 AFEDari dagokion boskotea honela formaliza daiteke:

$$(Q, A, \nu_6, q_0, Y_6)$$

Boskote horretako osagaiak honela defini daitezke:

- $Q = \{q_0, q_1, q_2, q_3\}$
- $A = \{a, b, c\}$
- $\nu_6 : Q \times A \rightarrow 2^Q$ trantsizio-funtzioa honako taula honen bidez defini daiteke:

ν_6	a	b	c
q_0	$\{q_1, q_3\}$	$\{q_2\}$	$\{q_2\}$
q_1	$\{q_1\}$	$\{q_1\}$	$\{q_1\}$
q_2	$\{q_3\}$	$\{q_2\}$	$\{q_2\}$
q_3	$\{q_2\}$	$\{q_3\}$	$\{q_3\}$

- q_0 hasierako egoera da.
- $Y_6 = \{q_0, q_1, q_3\}$.

N_7 AFEDari dagokion boskotea honela formaliza daiteke:

$$(R, A, \nu_7, r_0, Y_7)$$

Boskote horretako osagaiak honako hauek dira:

- $R = \{r_0, r_1\}$
- $A = \{a, b, c\}$
- $\nu_7 : R \times A \to 2^R$ trantsizio-funtzioa da eta honako taula honen bidez defini daiteke:

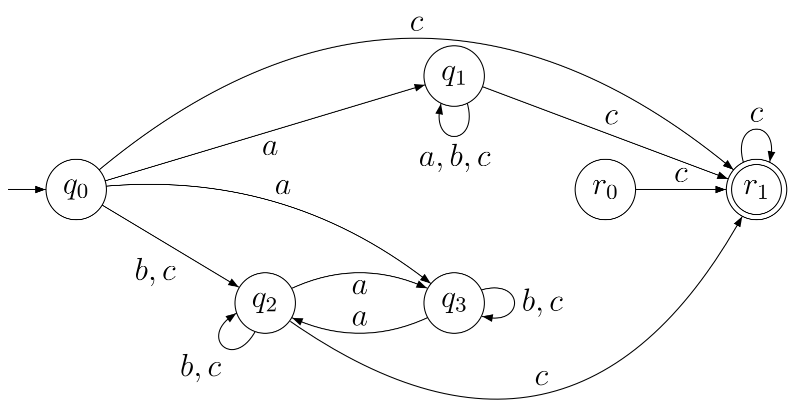
ν_7	a	b	c
r_0	$\varnothing$	$\varnothing$	$\{r_1\}$
r_1	$\varnothing$	$\varnothing$	$\{r_1\}$

- r_0 hasierako egoera da.
- $Y_7 = \{r_1\}$.

3.3.54 irudian a-rekin hasi edo a kopuru bikoitia duten eta c sinboloaz osatutako bloke ez-huts batez bukatzen diren hitzen lengoaiari dagokion N AFEDa erakusten da. N AFED hori N_6 eta N_7 AFEDak erabiliz eta 3.3.3.6.1 atalean aurkeztutako metodo sistematikoa aplikatuz eraiki da. N AFED hori (S, A, ν, s_0, Y) erako boskote gisa formaliza daiteke. Boskote horren osagaiak honako hauek dira:

- $S = Q \cup R = \{q_0, q_1, q_2, q_3, r_0, r_1\}$.
- A alfabetoa N_6 eta N_7 AFEDetako alfabeto bera da. Beraz, $A = \{a, b, c\}$.
- s_0 hasierako egoera q_0 da.
- Onarpen egoeren Y multzoari dagokionez:
  - $r_0 \notin Y_2$ denez, $Y = Y_2 = \{r_1\}$ da.
- Azkenik, ν trantsizio-funtzioa $S \times A \to 2^S$ erakoa da eta honako taula honen bidez defini daiteke:

N



3.3.54 irudia. a-rekin hasi edo a kopuru bikoitia duten eta c sinboloaz osatutako bloke ez-huts batez bukatzen diren hitzen lengoaiari dagokion AFEDa.

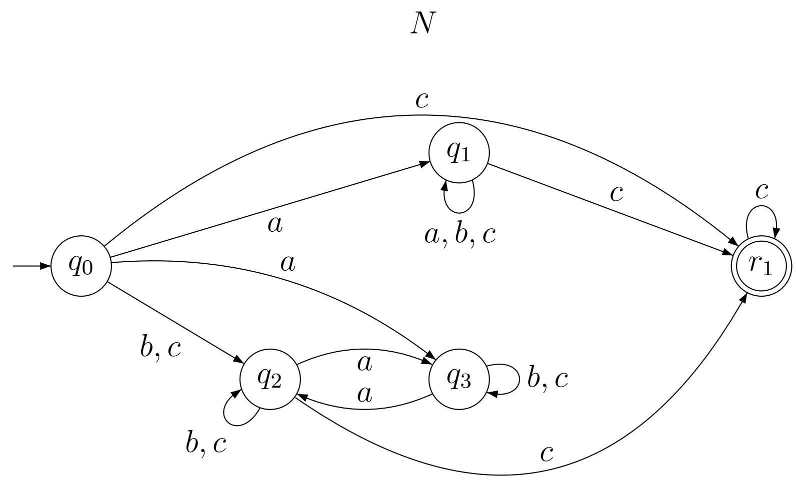
ν	a	b	c
q_0	$\{q_1, q_3\}$	$\{q_2\}$	$\{q_2, r_1\}$
q_1	$\{q_1\}$	$\{q_1\}$	$\{q_1, r_1\}$
q_2	$\{q_3\}$	$\{q_2\}$	$\{q_2, r_1\}$
q_3	$\{q_2\}$	$\{q_3\}$	$\{q_3\}$
r_0	$\varnothing$	$\varnothing$	$\{r_1\}$
r_1	$\varnothing$	$\varnothing$	$\{r_1\}$

3.3.54 irudiko trantsizio-diagraman ikus daiteke r_0 egoeratik ateratzen den trantsizio bakoitzeko, Y_6 multzoko egoera bakoitzetik ere trantsizio bat ateratzen dela. Kasu honetan r_0 egoerak ez ditu bi zirkulu eta, ondorioz, Y_6 multzoko egoerek ere ez. Bestalde, r_0 iristezina da q_0 egoeratik eta ken daiteke.

3.3.55 irudian r_0 egoera iristezina kendutakoan lortzen den AFEDa erakusten da. Era honetako ezabaketak egiten direnean abiapuntuko AFEDak ez dira garbi ikusten AFEDberrian eta, ondorioz, kateaketaren modularitatea eta argitasuna galdu egiten dira.

3.3.3.6.3 a-rekin hasi edo a kopuru bikoitia duten eta luzera bikoitzekoa den c sinboloaz osatutako bloke batez bukatzen diren hitzen lengoaiari dagokion AFED baten diseinu sistematikoa

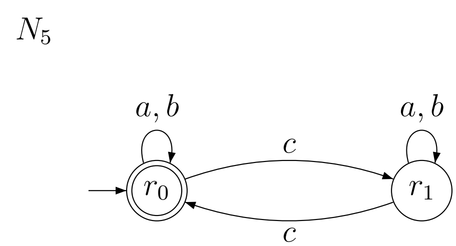
Har dezagun $A = \{a, b, c\}$ alfabetoa. a-rekin hasi edo a kopuru bikoitia duten eta luzera bikoitzekoa den c sinboloaz osatutako bloke batez bukatzen diren hitzen L lengoaiari dagokion AFED bat diseinatu nahi da. Horretarako, a-rekin hasi edo a kopuru bikoitia duten hitzen L_6 lengoaiari dagokion AFED bat eta bakarrik c sinboloaz osatuta dauden luzera bikoitzeko hitzen L_8 lengoaiari dagokion AFED bat hartuko ditugu abiapuntutzat eta 3.3.3.6.1 atalean aurkeztutako metodo sistematikoa aplikatuko da.



3.3.55 irudia. a-rekin hasi edo a kopuru bikoitia duten eta c sinboloaz osatutako bloke ez-huts batez bukatzen diren hitzen lengoaiari dagokion AFEDa (egoera iristezinak kenduta).

L lengoaia L_6L_8 eran defini daiteke. Hor, L_6 eta L_8 lengoaiak honela formaliza daitezke:

$$L_6 = \{w \mid w \in A^* \wedge (\exists u(u \in A^* \wedge w = au) \vee (|w|_a \text{ mod } 2 = 0))\}$$
$$L_8 = \{w \mid w \in A^* \wedge |w| = |w|_c \wedge |w| \text{ mod } 2 = 0\}$$



3.3.56 irudia. c kopuru bikoitia duten hitzen L_8 lengoaiari dagokion AFED bat.

3.3.52 irudian L_6 lengoaiari dagokion N_6 AFEDaren trantsizio-diagrama erakusten da eta 3.3.56 irudian L_8 lengoaiari dagokion N_8 AFEDaren trantsizio-diagrama erakusten da.

N_6 AFEDari dagokion boskotea 3.3.3.6.2 atalean definitu da. N_8 AFEDari dagokion boskotea honela formaliza daiteke:

$$(R, A, \nu_8, r_0, Y_8)$$

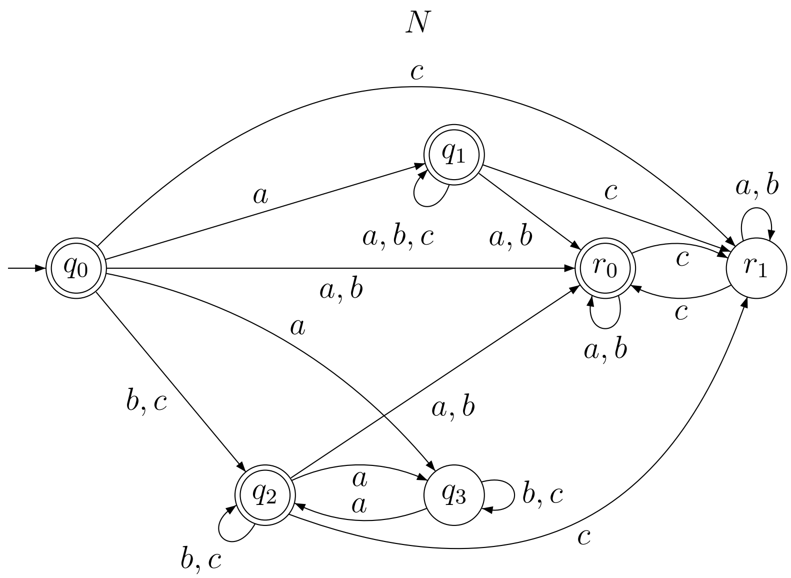
Boskote horretako osagaien definizioa honako hau da:

- $R = \{r_0, r_1\}$

- $A = \{a, b, c\}$
- $\nu_8 : R \times A \to 2^R$ trantsizio-funtzioa da eta honako taula honen bidez defini daiteke:

ν_8	a	b	c
r_0	$\{r_0\}$	$\{r_0\}$	$\{r_1\}$
r_1	$\{r_1\}$	$\{r_1\}$	$\{r_0\}$

- r_0 hasierako egoera da.
- $Y_8 = \{r_0\}$.



3.3.57 irudia. a-rekin hasi edo a kopuru bikoitia duten eta luzera bikoitzekoa den c sinboloaz osatutako bloke batez bukatzen diren hitzen lengoaiari dagokion AFED bat

3.3.57 irudian a-rekin hasi edo a kopuru bikoitia duten eta luzera bikoitzekoa den c sinboloaz osatutako bloke batez bukatzen diren hitzen lengoaiari dagokion N AFED baten trantsizio-diagrama erakusten da. AFED hori N_6 eta N_8 AFEDak hartu eta 3.3.3.6.1 atalean aurkeztutako metodo sistematikoa aplikatuz diseinatu da. N AFEDa (S, A, ν, s_0, Y) boskote gisa formaliza daiteke. Boskote horretako osagaiak honela defini daitezke:

- $S = Q \cup R = \{q_0, q_1, q_2, q_3, r_0, r_1\}$.
- A alfabetoa N_6 eta N_8 AFEDetako alfabeto bera da. Beraz, $A = \{a, b, c\}$.
- s_0 hasierako egoera q_0 da.
- Onarpen egoeren Y multzoari dagokionez:

  - $r_0 \in Y_2$ betetzen denez, $Y = Y_1 \cup Y_2 = \{q_0, q_1, q_2, r_1\}$ da.

- Azkenik, ν trantsizio-funtzioa $S \times A \to 2^S$ erakoa da eta honako taula honen bidez defini daiteke:

ν	a	b	c
q_0	$\{q_1, q_3, r_0\}$	$\{q_2, r_0\}$	$\{q_2, r_1\}$
q_1	$\{q_1, r_0\}$	$\{q_1, r_0\}$	$\{q_1, r_1\}$
q_2	$\{q_3, r_0\}$	$\{q_2, r_0\}$	$\{q_2, r_1\}$
q_3	$\{q_2\}$	$\{q_3\}$	$\{q_3\}$
r_0	$\{r_0\}$	$\{r_0\}$	$\{r_1\}$
r_1	$\{r_1\}$	$\{r_1\}$	$\{r_0\}$

3.3.57 irudiko trantsizio-diagraman ikus daiteke r_0-tik ateratzen den trantsizio bakoitzeko, Y_6-ko egoera bakoitzetik ere trantsizio bat ateratzen dela. Adibide honetan r_0 egoerak bi zirkulu ditu eta, ondorioz, Y_6 multzoko egoerek ere bai.
N_6 eta N_8 elkarri lotzeko hainbeste trantsizio gehitu beharrak modularitatea eta argitasuna galtzea dakar berarekin.

3.4.

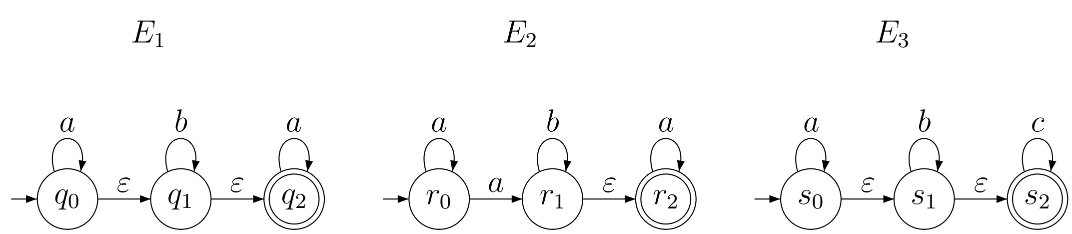
ε trantsizioak dituzten Automata Finitu Ez-Deterministak (ε-AFEDak)

3.4.1 ε-AFEDen definizioa

Automata finituen barruan, hirugarren mota bat bereiz dezakegu: ε trantsizioak dituzten AFEDak. Automata hauek ε-AFED bezala laburtuko dira.

Automata hauetan trantsizio hutsak edo ε bidezko trantsizioak ditugu. Grafikoki adierazten direnean, trantsizio hutsei dagozkien gezietan ε ipini ohi da.

3.4.1.1 Adibide batzuk



3.4.1 irudia. $\{a\}^*\{b\}^*\{a\}^*$, $\{a\}^+\{b\}^*\{a\}^*$ eta $\{a\}^*\{b\}^*\{c\}^*$ lengoaientzako E_1, E_2 eta E_3 ε-AFEDak.

3.4.1 irudian hiru ε-AFED ditugu: E_1, E_2 eta E_3 ε-AFEDak. Automata horietan, ε trantsizioak daudenean egoera batetik bestera joan gaitezke sinbolorik irakurri gabe.

E_1 ε-AFEDari dagokion lengoaia honako hau da:

$$\{w \mid w \in A^* \wedge \exists u, v, x(u \in A^* \wedge v \in A^* \wedge x \in A^* \wedge |u|_a = |u| \wedge |v|_b = |v| \wedge |x|_a = |x| \wedge w = uvx)\}$$

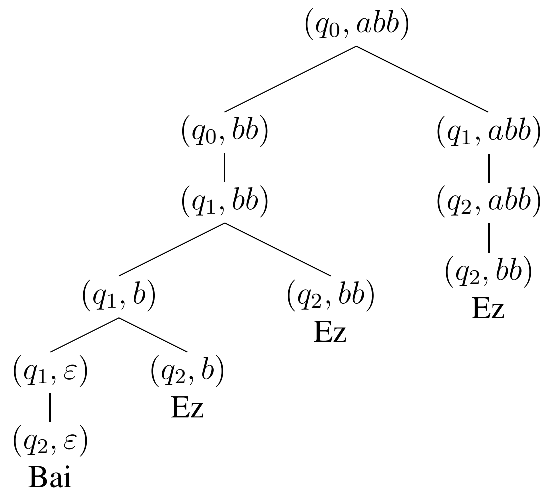
Lengoaia hori beste era honetara ere formaliza daiteke:

$$\{w \mid w \in A^* \land \exists j, k, \ell(j \in I\!N \land k \in I\!N \land \ell \in I\!N \land w = a^j b^k a^\ell)\}$$

Lengoaia hori bera adierazteko beste era bat honako hau izango litzateke:

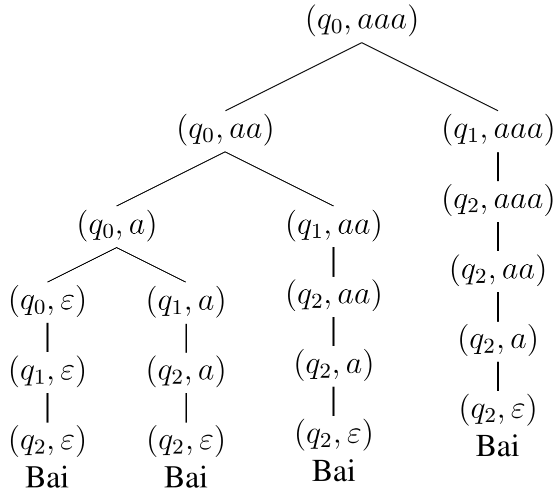
$$\{a\}^*\{b\}^*\{a\}^*$$

abb hitza lengoaia horretakoa dela erabakitzeko jarraituko litzatekeen prozesua konfigurazio deterministez osatutako honako zuhaitz honen bidez adieraz daiteke:



Zuhaitz horretan garbi ikus daiteke urrats batzuetan ez dela sinbolorik irakurtzen edo kontsumitzen. Hori ε trantsizioak daudelako da. ε-AFEDak "Bai" erantzuna itzuliko du gutxienez adar batean hitz osoa irakurtzea eta zirkulu bikoitzeko egoera batera iristea lortu delako: ezkerreko ertzeko adarrean (q_2, ε) konfigurazioa agertzen da hain zuzen ere. Gainerako adarretan, onarpen egoera den q_2 egoerara iritsi arren, ez da lortu hitz osoa irakurtzea. Zuhaitza xehetasun gehiagorekin aztertzen badugu, zuhaitzaren erroa den (q_0, abb) konfigurazioan gaudenean bi aukera daudela ikus dezakegu: a irakurri (edo kontsumitu) eta q_0 egoeran gelditu, (q_0, bb) konfigurazio berria sortuz edo a ez irakurri eta q_1 egoerara joan ε trantsizioa erabiliz. Bigarren kasu horretan, konfigurazio berria (q_1, abb) izango da. Bestalde, (q_0, bb) konfigurazioa daukagunean aukera bakarra dugu: ε trantsizioa erabiliz q_1 egoerara igaro sinbolorik kontsumitu gabe. Konfigurazio berria (q_1, bb) izango da. Era berean, (q_2, abb) konfigurazioan baldin bagaude, aukera bakarra a irakurtzea (edo kontsumitzea) eta q_2 egoeran bertan gelditzea da. Kasu horretan (q_2, bb) konfigurazioa sortuko da. Konfigurazio berri horretan ez dago aurrera jarraitzeko aukerarik, q_2 egoerarentzat ez baitago trantsiziorik b sinboloarekin. Zuhaitzeko gainerako adarretan eta adabegietan, oraintxe aipatu diren kasuetan kontuan hartu diren irizpideei jarraitzen zaie.

aaa hitza lengoaia horretakoa dela erabakitzeko jarraituko litzatekeen prozesua honela adieraz dezakegu trantsizio deterministez osatutako zuhaitz baten bidez:



Zuhaitz horretan ere garbi ikus daiteke urrats batzuetan ez dela sinbolorik irakurtzen edo kontsumitzen. Hori ε trantsizioak daudelako da. ε-AFEDak "Bai" erantzuna itzuliko du gutxienez adar batean hitz osoa irakurtzea eta bi zirkulu dituen egoera batera iristea lortu delako. Adibide honetan adar guztietan (q_2, ε) konfigurazioa agertzen da hain zuzen ere.

E_2 ε-AFEDari dagokion lengoaia honela defini daiteke:

$$\{w \mid w \in A^* \wedge \exists u, v, x(u \in A^* \wedge v \in A^* \wedge x \in A^* \wedge |u|_a = |u| \wedge |v|_b = |v| \wedge |x|_a = |x| \wedge w = auvx)\}$$

Lengoaia hori beste era honetara ere formaliza daiteke:

$$\{w \mid w \in A^* \wedge \exists j, k, \ell(j \in I\!N \wedge k \in I\!N \wedge \ell \in I\!N \wedge j \geq 1 \wedge w = a^j b^k a^\ell)\}$$

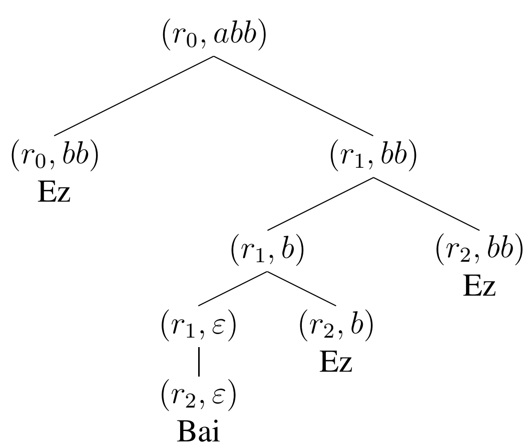
Lengoaia hori adierazteko beste era bat honako hau izango litzateke:

$$\{a\}\{a\}^*\{b\}^*\{a\}^*$$

Honako era honetara ere formaliza genezake:

$$\{a\}^+\{b\}^*\{a\}^*$$

abb hitza lengoaia horretakoa dela erabakitzeko jarraituko den prozesua honela adieraz dezakegu zuhaitz baten bidez:



Zuhaitz honetan ere urrats batzuetan ez da sinbolorik irakurtzen edo kontsumitzen. Hori ε trantsizioak daudelako da. ε-AFEDak "Bai" erantzuna itzuliko du gutxienez adar batean hitz osoa irakurtzea eta zirkulu bikoitzeko egoera batera iristea lortu delako: ezkerretik hasi eta bigarren adarrean (r_2, ε) konfigurazioa dugu. Ezkerreko ertzeko adarrean ez da hitz osoa irakurtzea lortu eta, gainera, ez da r_2 egoerara iritsi. Eskuinena dauden bi adarretan zirkulu bikoitza duen r_2 egoerara iristea lortu arren, ezin izan da hitz osoa irakurri.

E_3 ε-AFED-ari dagokion lengoaia honela defini daiteke:

$$\{w \mid w \in A^* \wedge \exists u, v, x(u \in A^* \wedge v \in A^* \wedge x \in A^* \wedge |u|_a = |u| \wedge |v|_b = |v| \wedge |x|_c = |x| \wedge w = uvx)\}$$

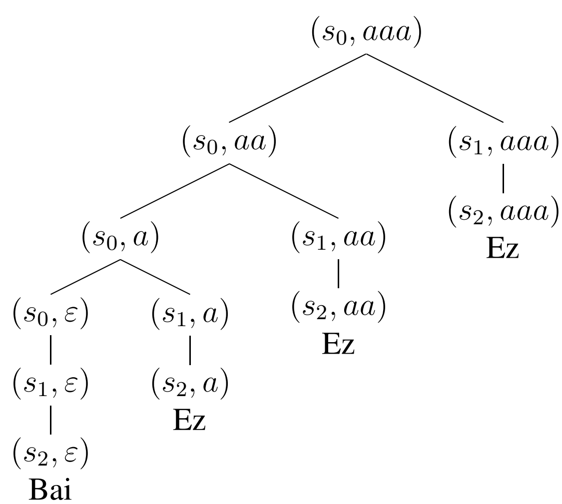
Honako era honetara ere formaliza daiteke lengoaia hori:

$$\{w \mid w \in A^* \wedge \exists j, k, \ell(j \in I\!N \wedge k \in I\!N \wedge \ell \in I\!N \wedge w = a^j b^k c^\ell)\}$$

Lengoaia hori formalizateko beste era bat honako hau litzateke:

$$\{a\}^*\{b\}^*\{c\}^*$$

aaa hitza lengoaia horretakoa dela erabakitzeko jarraituko litzatekeen prozesua honela adieraz dezakegu konfigurazio deterministez eratutako zuhaitz baten bidez:



Zuhaitz horretan erakusten den bezala, urrats batzuetan ez da sinbolorik kontsumitzen edo irakurtzen, hau da, egoeraz aldatu arren hitz berarekin jarraitzen da. Hori ε trantsizioengatik gertatzen da. ε-AFEDak "Bai" erantzuna itzuliko du gutxienez adar batean hitz osoa irakurtzea eta zirkulu bikoitzeko egoera batera iristea lortu delako: ezkerreko ertzeko adarrean (s_2, ε) konfigurazioa dugu hain zuzen ere. Gainerako adarretan, nahiz eta bi zirkulu dituen s_2 egoerara iritsi, ez da lortu hitz osoa irakurtzea.

ccc hitza lengoaia horretakoa dela erabakitzeko jarraituko litzatekeen prozesua honela adieraz dezakegu konfigurazio deterministez eratutako zuhaitz baten bidez:

$$\begin{array}{c} (s_0, ccc) \\ | \\ (s_1, ccc) \\ | \\ (s_2, ccc) \\ | \\ (s_2, cc) \\ | \\ (s_2, c) \\ | \\ (s_2, \varepsilon) \\ \text{Bai} \end{array}$$

ε trantsizioak direla eta, zuhaitz honetan ere urrats batzuetan ez da sinbolorik kontsumitzen edo irakurtzen, hau da, egoeraz aldatu arren hitz berarekin jarraitzen da. E_3 ε-AFEDak "Bai" erantzuna itzuliko du ccc hitzarentzat gutxienez adar batean hitz osoa irakurtzea eta zirkulu bikoitzeko egoera batera iristea lortu delako. Kasu honetan, zuhaitzak adar bakarra du eta adar horretan (s_2, ε) konfigurazioa agertzen da.

3.4.1.2 ε-AFEDak definitzeko arrazoiak: AFEDetan hobetu daitezkeen ezaugarriak

AFEDetan disjuntzioa AFDetan baino errazago maneia daitekeela ikusi da aurreko ataletan. Esate baterako, L_1 eta L_2 lengoaiak hartuz, $L_1 \cup L_2$ lengoaiari dagokion AFEDa nahiko era errazean eraiki daiteke L_1 eta L_2 lengoaiei dagozkien AFEDetatik abiatuta. Hala ere, hasierako egoera berri bat ipini beharko da eta abiapuntuko bi AFEDetako hasierako egoeretatik ateratzen den trantsizio bakoitzeko, hasierako egoera berri horretatik ere trantsizio bat ipini beharko da.

AFEDak erabiliz modularitatea handiagoa da baina oraindik ez da erabatekoa. Modularitatea disjuntzioarekin zerikusia duten kalkulu desberdinak aldi berean aurrera eramatea ahalbidetuz lortzen da edo elkartu beharreko bi AFED bere horretan mantenduz. Trantsizio hutsak (edo ε trantsizioak) ipintzeko ahalmenak aukera disjuntibo desberdinak guztiz banantzea ahalbidetzen du. Horrela, hasieratik aldi berean kalkulu desberdinak aurrera eramatea posible izango da eta gainera AFEDetan beharrezkoak diren doitzeak saihestuz. ε-AFEDekin modularitatea eta berrerabilgarritasuna lengoaien gaineko beste eragiketa batzuetara ere hedatu daiteke. ε-AFEDen erabilera bereziki onuragarria da lengoaien bilketa (disjuntzioa), lengoaien kateaketa, lengoaien itxidura eta lengoaien alderantzizkoari dagozkien ε-AFEDak kalkulatzeko.

3.4.1.3 ε-AFEDen definizio formala

Atal honetan, ε-AFEDen osagaien definizioa, ε-AFEDen funtzionalitatea, ε-AFEDen kasuan konfigurazioak nolakoak diren eta ε-AFEDentzat konputazioa nola formalizatzen den zehaztuko dira.

3.4.1.3.1 ε-AFEDen osagaiak

ε **trantsizioak dituen** E **Automata Finitu Ez-Determinista** bat honelako boskote bat da:

$$(Q, A, \lambda, q_0, Y)$$

Boskote horretako osagai bakoitzaren definizioa honako hau da:

- Q egoren multzo finitua da: $q_0, q_1, q_2, q_3, \ldots$
- A alfabetoa da.
- λ osagaia[1], ε trantsizioak dituen trantsizio-funtzio ez-determinista da. Q multzoko egoera bat eta $A\cup\{\varepsilon\}$ multzoko elementu bat emanda, λ trantsizio-funtzioak zein egoeretara edo egoera-multzora igaro behar den adierazten du. Beraz, λ funtzioaren mota honako hau izango da:

$$\lambda : Q \times (A \cup \{\varepsilon\}) \to 2^Q$$

  Hor, 2^Q-ren bidez $Q \to 2$ erako funtzio denen multzoa adierazten da, $2 = \{0, 1\}$ izanda. Gainera 2 multzoan 0 balioak *False* adierazten du eta 1 balioak *True* adierazten du. Beste era batera esanda, 2^Q multzoa Q multzoaren azpimultzo denez osatutako multzoa da.
- q_0 hasierako egoera da. Hasierako egoera beti bakarra izaten da.
- Y osagaia onarpen egoeren multzoa da. Y multzoa Q multzoaren azpimultzoa da, hau da, $Y \subseteq Q$, edo beste era batera esanda, $Y \in 2^Q$. ε-AFEDari hitz bat emandakoan, hitz osoa irakurtzea lortzen badu eta, gainera, Y-ko den egoera batera iristea lortzen badu, "Bai" erantzungo du. Hitz osoa irakurtzea ez bada lortzen edo hitz osoa irakurri arren Y-koa den egoera batera iristea ez badu lortzen, "Ez" erantzungo du.

ε-AFEDetan egoera batetik beste batera sinbolorik kontsumitu gabe igaro gaitezke bi egoera horien artean trantsizio hutsa, hau da, ε sinboloa duen trantsizioa baldin badago. Eta hori da AFED eta ε-AFEDen arteko oinarrizko desberdintasuna.

3.4.1 irudiko ε-AFEDetara itzuliz, E_1 ε-AFEDa honako boskote honen bidez adieraz daiteke:

$$(Q, A, \lambda_1, q_0, Y_1)$$

Boskote horretako osagaiak honako hauek dira:

- $Q = \{q_0, q_1, q_2\}$
- $A = \{a, b, c\}$

[1] λ *lambda letra grekoa da*

- $\lambda_1 : Q \times (A \cup \{\varepsilon\}) \to 2^Q$ honako taula honen bidez definitutako trantsizio-funtzioa da:

λ_1	a	b	c	ε
q_0	$\{q_0\}$	$\varnothing$	$\varnothing$	$\{q_1\}$
q_1	$\varnothing$	$\{q_1\}$	$\varnothing$	$\{q_2\}$
q_2	$\{q_2\}$	$\varnothing$	$\varnothing$	$\varnothing$

- q_0 hasierako egoera da.
- $Y_1 = \{q_2\}$.

Bestalde, E_2 ε-AFEDa honako boskote honen bidez formaliza daiteke:

$$(R, A, \lambda_2, r_0, Y_2)$$

Boskote horretako osagaien definizioa honako hau da:

- $R = \{r_0, r_1, r_2\}$
- $A = \{a, b, c\}$
- $\lambda_2 : R \times (A \cup \{\varepsilon\}) \to 2^R$ honako taula honen bidez definitutako trantsizio-funtzioa da:

λ_2	a	b	c	ε
r_0	$\{r_0, r_1\}$	$\varnothing$	$\varnothing$	$\varnothing$
r_1	$\varnothing$	$\{r_1\}$	$\varnothing$	$\{r_2\}$
r_2	$\{r_2\}$	$\varnothing$	$\varnothing$	$\varnothing$

- r_0 hasierako egoera da.
- $Y_2 = \{r_2\}$.

Azkenik, E_3 ε-AFEDa boskote bezala honela adierazi daiteke:

$$(S, A, \lambda_3, s_0, Y_3)$$

Boskote horren osagaien definizioa honako hau da:

- $S = \{s_0, s_1, s_2\}$
- $A = \{a, b, c\}$
- $\lambda_3 : S \times (A \cup \{\varepsilon\}) \to 2^S$ honako taula honen bidez definitutako trantsizio-funtzioa da:

λ_3	a	b	c	ε
s_0	$\{s_0\}$	$\varnothing$	$\varnothing$	$\{s_1\}$
s_1	$\varnothing$	$\{s_1\}$	$\varnothing$	$\{s_2\}$
s_2	$\varnothing$	$\varnothing$	$\{s_2\}$	$\varnothing$

- s_0 hasierako egoera da.
- $Y_3 = \{s_2\}$.

3.4.1.3.2 ε-AFEDen funtzionalitatea

(Q, A, q_0, λ, Y) erako ε-AFEDen **funtzionalitatea** AFEDen funtzionalitatearen antzekoa da. Desberdintasun nagusia honako hau da: ε-AFED batek aurrera egin dezake sinbolorik kontsumitu gabe ε trantsizio baten aurrean dagoenean. Beraz, ε-AFED baten jokabidea honako hau izango da: ε-AFEDa hasierako egoera den q_0 egoeratik abiatuko da eta sarrera bezala A^* multzokoa den hitz bat jasoko du. ε-AFEDa sinboloak kontsumituz eta egoeraz aldatuz joango da (batzuetan, egoera berean ere jarraitu dezake sinbolo bat kontsumitutakoan) baina ε trantsizioak eskura dituenean sinbolorik kontsumitu gabe ere egoeraz alda daiteke. Bai ε trantsizioak daudenean (eta, ondorioz, sinbolorik kontsumitu gabe aurrera egin daitekeenean) eta bai sinboloba kontsumitzen denean, λ trantsizio-funtzioa da hurrengo egoera zein izango duen zehazten duena. Ondorioz, q_i egoeran baldin bagaude eta ε trantsizioa baldin badugu eta $\lambda(q_i, \varepsilon) = \{q_{j_0}, q_{j_1}, \ldots, q_{j_n}\}$ baldin bada, $\{q_{j_0}, q_{j_1}, \ldots, q_{j_n}\}$ egoeretara joango gara. Era berean, q_i egoeran baldin bagaude eta A alfabetokoa den α sinboloa irakurtzen badugu eta $\lambda(q_i, \alpha) = \{q_{j_0}, q_{j_1}, \ldots, q_{j_n}\}$ baldin bada, orduan $\{q_{j_0}, q_{j_1}, \ldots, q_{j_n}\}$ egoeretara igaroko gara. Bestalde, $\lambda(q_i, \varepsilon) = \varnothing$ eta $\lambda(q_i, \alpha) = \varnothing$ betetzen badira, hor $\alpha \in A$ izanda, egoera horretatik ezingo da aurrera egin. Oro har, une bakoitzean egoera batean baino gehiagotan egongo gara eta egoera horietako batzuetatik aurrera egiterik ez eduki arren, gerta daiteke egoera horietako beste batzuetatik aurrera egin ahal izatea. Bukaerako erantzuna baiezkoa edo ezezkoa izango al den erabakitzeko irizpidea honako hau da:

- Une honetako egoeretatik ezin bada aurrera egin eta hitz osoa irakurtzea ez bada lortu, “Ez” erantzun beharko da.

- Hitz osoa irakurtzea lortu bada, 2^Q multzokoa den S egoera-multzo batean egongo gara [2]. S multzoko egoeretatik gutxienez bat Y multzokoa baldin bada, orduan ε-AFEDak “Bai” erantzungo du, hitza lengoaiakoa izango delako. Aldiz, S egoera-multzoan ez badago Y multzokoa den elementurik, automatak “Ez” erantzungo du, hitza ez baita automata horri dagokion lengoaiakoa izango. Beste era batera esanda, $S \cap Y \neq \varnothing$ betetzen bada, orduan “Bai” erantzungo du eta $S \cap Y = \varnothing$ betetzen bada, “Ez” erantzungo du.

3.4.1.3.3 Konfigurazio kontzeptua ε-AFEDetan

AFEDetan bezala, ε-AFEDetan ere **konfigurazio ez-determinista** bat (S, w) erako bikote bat da, hor S osagaia Q multzoaren azpimultzo bat (eta ondorioz, 2^Q multzoko elementu bat) eta w osagaia A^* multzoko elementu bat izanda. Beraz, ε-AFEDetan ere konfigurazio ez-deterministak $2^Q \times A^*$ motakoak dira. Ondorioz, AFEDetan eta ε-AFEDetan erabiltzen diren konfigurazio ez-deterministen artean ez dago desberdintasunik.

3.4.1.3.4 Konputazioaren kontzeptua ε-AFEDetan

ε-AFED batek burututako **konputazioa** λ trantsizio-funtzioan oinarritzen den λ^* funtzioaren bidez definitzen da. λ^* funtzioaren mota honako hau da:

[2] $S \in 2^Q$ edo, beste era batera esanda, $S \subseteq Q$

$$\lambda^* : 2^Q \times A^* \to 2^Q$$

δ eta ν trantsizio-funtzioekin gertatzen den bezala, λ funtzioak urrats bakar bat adierazten du eta, δ^* eta ν^* funtzioekin gertatzen den bezala, λ^* funtzioak egoera-multzo batetik abiatuz hitz oso bat irakurriz zein egoeretara iritsi gaitezkeen adierazten du. Beraz, $\lambda^*(\{q_{j_0}, q_{j_1}, \ldots, q_{j_k}\}, w)$ espresioaren esanahia $\{q_{j_0}, q_{j_1}, \ldots, q_{j_k}\}$ egoeretatik w hitzeko sinboloak eta ε erabiliz zein egoeretara iritsi gaitezkeen da.

ε-AFEDetan sinbolorik irakurri gabe aurrera egin daitekeenez (egoeraz alda gaitezkeenez), bi egoeren arteko ε *bidea* eta egoera batetik ε trantsizioak zeharkatuz irisgarriak diren egoeren multzoa kontzeptuak formalizatuko ditugu hasteko.

Har ditzagun q eta q' ε-AFED baten bi egoera eta demagun λ dela ε-AFED horren trantsizio-funtzioa. $k \geq 1$, $q = q_{j_0}$ eta $q' = q_{j_k}$ izanda, $\langle q_{j_0}, q_{j_1}, \ldots, q_{j_k} \rangle$ erako sekuentzia finitu bat, q eta q'egoeren arteko ε **bidea** dela esango da honako hau betetzen bada: edozein $\ell \in \{0, 1, 2, \ldots, k-1\}$ hartuta, $q_{j_{\ell+1}} \in \lambda(q_{j_\ell}, \varepsilon)$.

ε bideek gutxienez bi osagai izango dituzte eta egoera bera behin baino gehiagotan agertzea ere litekeena da.

q egoera bat izanda, λ funtzioko ε trantsizioak zeharkatuz q egoeratik irisgarriak diren egoeren multzoa definituko da orain. Multzo horri $\mathsf{Tr}\text{-}\varepsilon(q, \lambda)$ deituko diogu.

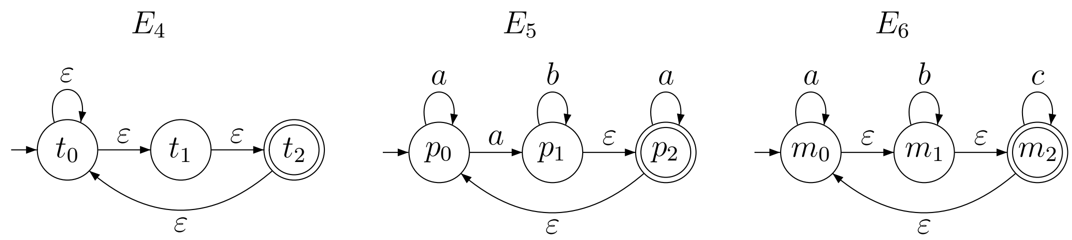
Har dezagun λ trantsizio-funtzioa duen ε-AFED bateko q egoera bat. q egoeratik abiatu eta bakarrik ε trantsizioak zeharkatuz **irisgarriak diren egoeren** $\mathsf{Tr}\text{-}\varepsilon(q, \lambda)$ multzoa, honako bi puntu hauek betetzen dituen multzo txikiena da:

- $\lambda(q, \varepsilon) \subseteq \mathsf{Tr}\text{-}\varepsilon(q, \lambda)$
- $q' \in \mathsf{Tr}\text{-}\varepsilon(q, \lambda)$ betetzen bada, orduan $\lambda(q', \varepsilon) \subseteq \mathsf{Tr}\text{-}\varepsilon(q, \lambda)$.

$\mathsf{Tr}\text{-}\varepsilon(q, \lambda)$ multzoa beti finitua izango da eta gerta daiteke hutsa izatea ere. $\mathsf{Tr}\text{-}\varepsilon$ funtzioaren mota honako hau da:

$$\mathsf{Tr}\text{-}\varepsilon : Q \times \underbrace{(Q \times (A \cup \{\varepsilon\}) \to 2^Q)}_{\lambda\text{-ren mota}} \to 2^Q$$

Itzul gaitezen berriro ere 3.4.1 irudiko E_1, E_2 eta E_3 ε-AFEDetara. E_1 ε-AFEDan hiru ε bide daude: $\langle q_0, q_1 \rangle$, $\langle q_1, q_2 \rangle$ eta $\langle q_0, q_1, q_2 \rangle$. ε trantsizioak zeharkatuz irisgarriak diren egoerei dagokienez, honako hau daukagu: $\mathsf{Tr}\text{-}\varepsilon(q_0, \lambda_1) = \{q_1, q_2\}$, $\mathsf{Tr}\text{-}\varepsilon(q_1, \lambda_1) = \{q_2\}$ eta $\mathsf{Tr}\text{-}\varepsilon(q_2, \lambda_1) = \varnothing$. Bestalde, E_2 ε-AFEDan ε bide bakarra dago: $\langle r_1, r_2 \rangle$. ε trantsizioak zeharkatuz irisgarriak diren egoerei dagokienez, $\mathsf{Tr}\text{-}\varepsilon(r_0, \lambda_2) = \varnothing$ da, $\mathsf{Tr}\text{-}\varepsilon(r_1, \lambda_2) = \{q_2\}$ da eta $\mathsf{Tr}\text{-}\varepsilon(r_2, \lambda_2) = \varnothing$ da. E_3 ε-AFEDaren kasua, E_1 ε-AFEDaren antzekoa da. E_3-ren kasuan hiru ε bide daude: $\langle s_0, s_1 \rangle$, $\langle s_1, s_2 \rangle$ eta $\langle s_0, s_1, s_2 \rangle$. ε trantsizioak zeharkatuz irisgarriak diren egoerei dagokienez, $\mathsf{Tr}\text{-}\varepsilon(s_0, \lambda_3) = \{s_1, s_2\}$, $\mathsf{Tr}\text{-}\varepsilon(s_1, \lambda_3) = \{s_2\}$ eta $\mathsf{Tr}\text{-}\varepsilon(s_2, \lambda_3) = \varnothing$ betetzen da.



3.4.2 irudia. $\{\varepsilon\}$, $(\{a\}^+\{b\}^*\{a\}^*)^+$ eta $(\{a\}^*\{b\}^*\{c\}^*)^* = A^*$ lengoaientzat E_4, E_5 eta E_6 ε-AFEDak.

3.4.2 irudian beste hiru ε-AFED ditugu: E_4, E_5 eta E_6. Automata horien kasuan, E_4 ε-AFEDari dagokion lengoaia $\{\varepsilon\}$ da, E_5 ε-AFEDari dagokion lengoaia $(\{a\}^+\{b\}^*\{a\}^*)^+$ da eta, azkenik, E_6 ε-AFEDari dagokion lengoaia $(\{a\}^*\{b\}^*\{c\}^*)^*$ da. Azkeneko lengoaia hori A^* lengoaia unibertsala da. $\{\varepsilon\}$ eta A^* lengoaientzat automata sinpleagoak definitzea erraza izan arren, E_4 eta E_6 ε-AFEDak aukeratu dira ε-AFEDen ezaugarri batzuk aztertzeko ondo datozelako.

E_4 ε-AFEDa boskote bezala honela adieraz daiteke:

$$(T, A, \lambda_4, t_0, Y_4)$$

Boskote horretako osagaien definizioa honako hau da:

- $T = \{t_0, t_1, t_2\}$
- $A = \{a, b, c\}$
- $\lambda_4 : T \times (A \cup \{\varepsilon\}) \to 2^T$ honako taula honen bidez definitutako trantsizio-funtzioa da:

λ_4	a	b	c	ε
t_0	$\varnothing$	$\varnothing$	$\varnothing$	$\{t_0, t_1\}$
t_1	$\varnothing$	$\varnothing$	$\varnothing$	$\{t_2\}$
t_2	$\varnothing$	$\varnothing$	$\varnothing$	$\{t_0\}$

- t_0 hasierako egoera da.
- $Y_4 = \{t_2\}$.

E_5 ε-AFEDa boskote bezala honela formaliza daiteke:

$$(P, A, \lambda_5, p_0, Y_5)$$

Boskote horretako osagaien definizioa honako hau da:

- $P = \{p_0, p_1, p_2\}$
- $A = \{a, b, c\}$
- $\lambda_5 : P \times (A \cup \{\varepsilon\}) \to 2^P$ honako taula honen bidez definitutako trantsizio-funtzioa da:

λ_5	a	b	c	ε
p_0	$\{p_0, p_1\}$	$\varnothing$	$\varnothing$	$\varnothing$
p_1	$\varnothing$	$\{p_1\}$	$\varnothing$	$\{p_2\}$
p_2	$\varnothing$	$\varnothing$	$\varnothing$	$\{p_0\}$

- p_0 hasierako egoera da.
- $Y_5 = \{p_2\}$.

E_6 ε-AFEDa boskote bezala honela defini daiteke:

$$(M, A, \lambda_6, m_0, Y_6)$$

Boskote horretako osagaien definizioa honako hau da:

- $M = \{m_0, m_1, m_2\}$
- $A = \{a, b, c\}$
- $\lambda_6 : M \times (A \cup \{\varepsilon\}) \to 2^M$ honako taula honen bidez definitutako trantsizio-funtzioa da:

λ_6	a	b	c	ε
m_0	$\{m_0\}$	$\varnothing$	$\varnothing$	$\{m_1\}$
m_1	$\varnothing$	$\{m_1\}$	$\varnothing$	$\{m_2\}$
m_2	$\varnothing$	$\varnothing$	$\{m_2\}$	$\{m_0\}$

- m_0 hasierako egoera da.
- $Y_6 = \{m_2\}$.

E_4 ε-AFEDan infinitu ε bide daude. Bide horietako bakoitza finitua izango da. Adibidez, $\langle t_0, t_0\rangle$, $\langle t_0, t_0, t_0\rangle$, $\langle t_0, t_0, t_0, t_0\rangle$, $\langle t_0, t_1\rangle$, $\langle t_1, t_2\rangle$, $\langle t_0, t_1, t_2\rangle$, $\langle t_1, t_2, t_0\rangle$, $\langle t_1, t_2, t_0, t_1\rangle$, $\langle t_0, t_1, t_2, t_0, t_1, t_2, t_0\rangle$ eta abar. ε trantsizioak zeharkatuz irisgarriak diren egoerei dagokienez, honako hau daukagu: $\mathsf{Tr}\text{-}\varepsilon(t_0, \lambda_4) = \{t_0, t_1, t_2\}$, $\mathsf{Tr}\text{-}\varepsilon(t_1, \lambda_4) = \{t_0, t_1, t_2\}$ eta $\mathsf{Tr}\text{-}\varepsilon(t_2, \lambda_4) = \{t_0, t_1, t_2\}$. Multzo hauek beti finituak izango dira eta egoera bakoitzeko horrelako multzo bakarra egongo da. Bestalde, E_5 ε-AFEDan hiru ε bide daude bakarrik: $\langle p_1, p_2\rangle$, $\langle p_2, p_0\rangle$ eta $\langle p_1, p_2, p_0\rangle$. ε trantsizioak zeharkatuz irisgarriak diren egoerei dagokienez, honako hau daukagu: $\mathsf{Tr}\text{-}\varepsilon(p_0, \lambda_5) = \varnothing$, $\mathsf{Tr}\text{-}\varepsilon(p_1, \lambda_5) = \{p_0, p_2\}$ eta $\mathsf{Tr}\text{-}\varepsilon(p_2, \lambda_5) = \{p_0\}$. Azkenik, E_6 ε-AFEDaren kasuan, E_4-ren kasuan bezala, infinitu ε bide daude. Beti bezala, bide horietako bakoitza finitua izango da. Adibidez, $\langle m_0, m_1\rangle$, $\langle m_0, m_1, m_2\rangle$, $\langle m_0, m_1, m_2, m_0\rangle$, $\langle m_0, m_1, m_2, m_0, m_1\rangle$,

$\langle m_1, m_2\rangle$, $\langle m_1, m_2, m_0, m_1\rangle$, $\langle m_2, m_0\rangle$ eta abar. ε trantsizioak zeharkatuz irisgarriak diren egoerei dagokienez, honako hau daukagu: $\textsf{Tr-}\varepsilon(m_0, \lambda_6) = \{m_0, m_1, m_2\}$, $\textsf{Tr-}\varepsilon(m_1, \lambda_6) = \{m_0, m_1, m_2\}$ eta $\textsf{Tr-}\varepsilon(m_2, \lambda_6) = \{m_0, m_1, m_2\}$.

$\lambda^* : 2^Q \times A^* \to 2^Q$ konputazio-funtzioak egoera-multzo bat eta hitz bat emanda, egoera horietatik abiatu eta hitz hori irakurri ondoren zein egoeretan egongo garen urratsez urrats nola kalkulatu dezakegun adierazi beharko digu. Urrats horietako bakoitzean, ε trantsizioak zeharkatuz irisgarriak diren egoerak kontuan hartu beharko dira. Horretarako, konputazio-prozesuan parte hartzen duen q egoera bakoitzeko, $\textsf{Tr-}\varepsilon(q, \lambda)$ egoera-multzoa kalkulatu beharko da. $\textsf{Tr-}\varepsilon(q, \lambda)$ multzoa definitu denean, egoera batek multzo horretakoa izateko bete behar duen baldintza zein den adierazi da baina ez da zehaztu nola kalkulatuko den baldintza hori betetzen duten egoeren multzoa. Hori dela-eta, $\textsf{Tr-}\varepsilon(q, \lambda)$ egoera-multzoa kalkulatuko duen funtzio bat proposatuko da orain. Funtzio horri $\textsf{Irisgarriak-}\varepsilon$ deituko diogu. Hasiera batean, $\textsf{Irisgarriak-}\varepsilon$ funtzioaren mota honako hau izango dela pentsa dezakegu:

$$\textsf{Irisgarriak-}\varepsilon :: Q \times \underbrace{(Q \times (A \cup \{\varepsilon\}) \to 2^Q)}_{\text{hau } \lambda\text{-ren mota da}} \to 2^Q$$

Baita ere pentsa dezakegu $\textsf{Irisgarriak-}\varepsilon$ funtzioak honela definituta egon beharko lukeela:

$$\textsf{Irisgarriak-}\varepsilon(q, \lambda) = \lambda(q, \varepsilon) \cup \left(\bigcup_{q_{j_\ell} \in \lambda(q,\varepsilon)} \textsf{Irisgarriak-}\varepsilon(q_{j_\ell}, \lambda)\right)$$

Beraz, $\textsf{Irisgarriak-}\varepsilon(q, \lambda)$ espresioak q egoeratik abiatu eta λ trantsizio-funtzioak baimendutako ε trantsizioak zeharkatuz irisgarriak diren egoeren multzoa adierazten du. Beste era batera esanda, q' egoera bat $\textsf{Irisgarriak-}\varepsilon(q, \lambda)$ multzokoa izango da q egoeratik q' egoerara ε bideren bat existitzen bada.

$\bigcup_{q_{j_\ell} \in \lambda(q,\varepsilon)} \textsf{Irisgarriak-}\varepsilon(q_{j_\ell}, \lambda)$ espresioaren esanahia honako hau da: $\lambda(q, \varepsilon) = \{q_{j_0}, q_{j_1}, \dots, q_{j_k}\}$, hau da, sinbolorik kontsumitu gabe urrats batean irisgarriak diren egoeren multzoa $\{q_{j_0}, q_{j_1}, \dots, q_{j_k}\}$ baldin bada, kalkulua errekurtsiboki errepikatu beharko da, $\{q_{j_0}, q_{j_1}, \dots, q_{j_k}\}$ egoeretatik ε trantsizioak zeharkatuz irisgarriak diren egoerak zein diren erabakitzeko. Beraz, $\lambda(q, \varepsilon) = \{q_{j_0}, q_{j_1}, \dots, q_{j_k}\}$ baldin bada:

$$\bigcup_{q_{j_\ell} \in \lambda(q,\varepsilon)} \textsf{Irisgarriak-}\varepsilon(q_{j_\ell}, \lambda) =$$
$$\textsf{Irisgarriak-}\varepsilon(q_{j_0}, \lambda) \cup \textsf{Irisgarriak-}\varepsilon(q_{j_1}, \lambda) \cup \cdots \cup \textsf{Irisgarriak-}\varepsilon(q_{j_k}, \lambda)$$

Hala ere, $\textsf{Irisgarriak-}\varepsilon(q, \lambda)$ funtzioaren definizio horrek arazo bat du: kalkulua ez bukatzea gerta daiteke. Beraz, $\textsf{Irisgarriak-}\varepsilon(q, \lambda)$ funtzioaren bidez kalkulatu nahi den $\textsf{Tr-}\varepsilon(q, \lambda)$ egoera-multzoa beti finitua izan arren, $\textsf{Irisgarriak-}\varepsilon(q, \lambda)$ funtzioaren bidezko kalkulua infinitua izan daiteke. Arazoa non dagoen ikusteko, E_4 ε-AFEDa hartu eta $\textsf{Irisgarriak-}\varepsilon(t_0, \lambda_4)$ espresioaren kalkulu-prozesua garatuko dugu.

$$
\begin{aligned}
&\textsf{Irisgarriak-}\varepsilon(t_0, \lambda_4)\\
&^{(1)} = \lambda_4(t_0, \varepsilon) \cup (\textstyle\bigcup_{t \in \lambda_4(t_0,\varepsilon)} \textsf{Irisgarriak-}\varepsilon(t, \lambda_4))\\
&^{(2)} = \{t_0, t_1\} \cup (\textsf{Irisgarriak-}\varepsilon(t_0, \lambda_4) \cup\ \textsf{Irisgarriak-}\varepsilon(t_1, \lambda_4))\\
&^{(3)} = \{t_0, t_1\} \cup \lambda_4(t_0, \varepsilon) \cup (\textstyle\bigcup_{t \in \lambda_4(t_0,\varepsilon)} \textsf{Irisgarriak-}\varepsilon(t, \lambda_4))\\
&\qquad\quad \cup \lambda_4(t_1, \varepsilon) \cup (\textstyle\bigcup_{t \in \lambda_4(t_1,\varepsilon)} \textsf{Irisgarriak-}\varepsilon(t, \lambda_4))\\
&^{(4)} = \{t_0, t_1\} \cup\ \{t_0, t_1\} \cup\ \textsf{Irisgarriak-}\varepsilon(t_0, \lambda_4) \cup\ \textsf{Irisgarriak-}\varepsilon(t_1, \lambda_4)\\
&\qquad\quad \cup \{t_2\} \cup \textsf{Irisgarriak-}\varepsilon(t_2, \lambda_4)\\
&^{(5)} = \{t_0, t_1, t_2\} \cup\ \textsf{Irisgarriak-}\varepsilon(t_0, \lambda_4) \cup\ \textsf{Irisgarriak-}\varepsilon(t_1, \lambda_4)\\
&\qquad\quad \cup \textsf{Irisgarriak-}\varepsilon(t_2, \lambda_4)\\
&^{(6)} = \{t_0, t_1, t_2\} \cup \lambda_4(t_0, \varepsilon) \cup (\textstyle\bigcup_{t \in \lambda_4(t_0,\varepsilon)} \textsf{Irisgarriak-}\varepsilon(t, \lambda_4))\\
&\qquad\quad \cup \lambda_4(t_1, \varepsilon) \cup (\textstyle\bigcup_{t \in \lambda_4(t_1,\varepsilon)} \textsf{Irisgarriak-}\varepsilon(t, \lambda_4))\\
&\qquad\quad \cup \lambda_4(t_2, \varepsilon) \cup (\textstyle\bigcup_{t \in \lambda_4(t_2,\varepsilon)} \textsf{Irisgarriak-}\varepsilon(t, \lambda_4))\\
&^{(7)} = \{t_0, t_1, t_2\} \cup \{t_0, t_1\} \cup \textsf{Irisgarriak-}\varepsilon(t_0, \lambda_4) \cup\ \textsf{Irisgarriak-}\varepsilon(t_1, \lambda_4)\\
&\qquad\quad \cup \{t_2\} \cup \textsf{Irisgarriak-}\varepsilon(t_2, \lambda_4)\\
&\qquad\quad \cup \{t_0\} \cup \textsf{Irisgarriak-}\varepsilon(t_0, \lambda_4)\\
&^{(8)} = \{t_0, t_1, t_2\} \cup\ \textsf{Irisgarriak-}\varepsilon(t_0, \lambda_4) \cup\ \textsf{Irisgarriak-}\varepsilon(t_1, \lambda_4)\\
&\qquad\quad \cup \textsf{Irisgarriak-}\varepsilon(t_2, \lambda_4)\\
&^{(9)} = \ldots
\end{aligned}
$$

Kalkulu-prozesua begizta batean sartu da. Izan ere, (5) eta (8) puntuetan gauza bera daukagu. Era honetako begiztak grafoak zeharkatu behar direnean agertu ohi diren arazoak dira. Azken batean, automata finitu bat norantzadun grafotzat har daiteke. Begizta hori sortzearen arrazoia honako hau da: Irisgarriak-ε funtzioak ez ditu "gogoratzen" aurretik tratatu dituen egoerak eta, horregatik, kalkuluak errepikatu egiten ditu. Oraintxe garatu den adibide horretan kalkulu berak infinitu aldiz egingo lituzke. Kalkulatu beharreko egoera denak kalkulatuta eduki arren, —adibide horretan, $\{t_0, t_1, t_2\}$ egoera-multzoa— ezinezkoa da prozesua geldiaraztea. Arazo hori konpontzeko, hirugarren parametro bat ipini diezaiokegu Irisgarriak-ε funtzioari. Hirugarren parametro hori egoera-multzo bat izango da eta dagoeneko tratatu diren egoerak "gogoratzeko" balioko du.

Beraz, Irisgarriak-ε funtzioaren behin betiko mota honako hau izango da:

$$\textsf{Irisgarriak-}\varepsilon :: Q \times \underbrace{(Q \times (A \cup\ \{\varepsilon\}) \to 2^Q)}_{\text{hau } \lambda\text{-ren mota da}} \times 2^Q \to 2^Q$$

Irisgarriak-ε funtzioaren **behin betiko definizio formala** honako hau da:

$$\textsf{Irisgarriak-}\varepsilon(q, \lambda, C) =$$
$$\begin{cases} \varnothing & q \in C \text{ baldin bada}\\ (\lambda(q,\varepsilon) \setminus C) \cup (\bigcup_{q_{j_\ell} \in (\lambda(q,\varepsilon)\setminus(C \cup \{q\}))} \textsf{Irisgarriak-}\varepsilon(q_{j_\ell}, \lambda, C \cup \{q\})) & \text{bestela} \end{cases}$$

Ondorioz, Irisgarriak-$\varepsilon(q, \lambda, C)$ espresioak q egoeratik abiatu eta λ trantsizio-funtzioak baimendutako ε trantsizioak zeharkatuz irisgarriak diren egoeren multzoa adieraziko du, baina C multzoko egoera batera daramaten ε trantsizioak ez dira kontuan hartu beharko irisgarriak diren egoeren multzo hori kalkulatzean. Beste era batera esanda, q' egoera bat Irisgarriak-$\varepsilon(q, \lambda, C)$

multzokoa izango da q egoeratik q' egoerara C multzokoak diren egoeretatik igarotzen ez den ε bideren bat existitzen bada.

Funtzio hori erabiliz q egoeratik abiatu eta λ trantsizio-funtzioak baimendutako ε trantsizioak zeharkatuz irisgarriak diren egoeren multzoa lortzeko, Irisgarriak-$\varepsilon(q, \lambda, \varnothing)$ kalkulatu beharko da. Irisgarriak-$\varepsilon(q, \lambda, \varnothing)$ kalkulatuz lortuko den multzoa Tr-$\varepsilon(q, \lambda)$ multzoa izango da. Orain, berriro E_4 ε-AFED hartuko dugu eta Irisgarriak-$\varepsilon(t_0, \lambda_4, \varnothing)$ kalkulatuko dugu:

Irisgarriak-$\varepsilon(t_0, \lambda_4, \varnothing)$
$\stackrel{(1)}{=} (\lambda_4(t_0, \varepsilon) \setminus \varnothing) \cup (\bigcup_{t \in (\lambda_4(t_0,\varepsilon) \setminus (\varnothing \cup \{t_0\}))} \text{Irisgarriak-}\varepsilon(t, \lambda_4, \varnothing \cup \{t_0\}))$
$\stackrel{(2)}{=} \{t_0, t_1\} \cup \text{ Irisgarriak-}\varepsilon(t_1, \lambda_4, \{t_0\})$
$\stackrel{(3)}{=} \{t_0, t_1\} \cup (\lambda_4(t_1, \varepsilon) \setminus \{t_0\}) \cup (\bigcup_{t \in (\lambda_4(t_1,\varepsilon) \setminus (\{t_0\} \cup \{t_1\}))} \text{Irisgarriak-}\varepsilon(t, \lambda_4, \{t_0\} \cup \{t_1\}))$
$\stackrel{(4)}{=} \{t_0, t_1, t_2\} \cup \text{ Irisgarriak-}\varepsilon(t_2, \lambda_4, \{t_0, t_1\})$
$\stackrel{(5)}{=} \{t_0, t_1, t_2\} \cup (\lambda_4(t_2, \varepsilon) \setminus \{t_0, t_1\}) \cup$
$(\bigcup_{t \in (\lambda_4(t_2,\varepsilon) \setminus (\{t_0,t_1\} \cup \{t_2\}))} \text{Irisgarriak-}\varepsilon(t, \lambda_4, \{t_0, t_1\} \cup \{t_2\}))$
$\stackrel{(6)}{=} \{t_0, t_1, t_2\} \cup (\{t_0\} \setminus \{t_0, t_1\}) \cup (\bigcup_{t \in (\{t_0\} \setminus (\{t_0,t_1,t_2\}))} \text{Irisgarriak-}\varepsilon(t, \lambda_4, \{t_0, t_1, t_2\}))$
$\stackrel{(7)}{=} \{t_0, t_1, t_2\} \cup \varnothing \cup \varnothing$
$\stackrel{(8)}{=} \{t_0, t_1, t_2\}$

Irisgarriak-ε funtzioaren bigarren definizioa erabiliz, kalkulu-prozesua bukatu egingo da emaitza bezala Tr-$\varepsilon(t_0, \lambda_4)$ multzoa itzuliz, hau da, $\{t_0, t_1, t_2\}$ multzoa itzuliz.

Trantsizio hutsak edo ε trantsizioak daudenean sinbolorik irakurri edo kontsumitu gabe aurrera egiteko aukera dugunez, q egoera batean egotea egoera horretan eta gainera egoera horretatik ε trantsizioen bidez irisgarriak diren egoeretan egotearen baliokidea da. Hori Itxidura-ε izeneko funtzio baten bidez formalizatuko dugu.

Itxidura-ε funtzioaren mota honako hau da:

$$\text{Itxidura-}\varepsilon :: Q \times \underbrace{(Q \times (A \cup \{\varepsilon\}) \to 2^Q)}_{\text{hau } \lambda\text{-ren mota da}} \to 2^Q$$

Itxidura-ε funtzioaren **definizio formala** honako hau da:

$$\text{Itxidura-}\varepsilon(q, \lambda) = \{q\} \cup \text{ Irisgarriak-}\varepsilon(q, \lambda, \varnothing) = \{q\} \cup \text{ Tr-}\varepsilon(q, \lambda)$$

Irisgarriak-$\varepsilon(q, \lambda, \varnothing)$ eta Itxidura-$\varepsilon(q, \lambda)$ espresioen arteko ezberdintasuna honako hau da: Itxidura-$\varepsilon(q, \lambda)$ multzoan q egoera beti agertuko da, nahiz eta q egoeratik q egoerara daraman ε biderik ez egon.

Itxidura-ε funtzioa multzoetara honela hedatzen da:

$$\text{Itxidura-}\varepsilon^{\#}(S, \lambda) = \bigcup_{q \in S} \text{Itxidura-}\varepsilon(q, \lambda)$$

Jarraian, 3.4.1 irudiko eta 3.4.2 irudiko ε-AFEDetara itzuliko gara berriro ere:

E_1-en $\lambda_1(q_0, \varepsilon) = \{q_1\}$ da eta Itxidura-$\varepsilon(q_0, \lambda_1) = \{q_0, q_1, q_2\}$ da. Bestalde, $\lambda_1(q_1, \varepsilon) = \{q_2\}$ da eta Itxidura-$\varepsilon(q_1, \lambda_1) = \{q_1, q_2\}$ da. Azkenik, $\lambda_1(q_2, \varepsilon) = \varnothing$ da eta Itxidura-$\varepsilon(q_2, \lambda_1) =$

$\{q_2\}$ da. Gainera, $\mathsf{Itxidura}\text{-}\varepsilon^{\#}(\{q_1, q_2\}, \lambda_1) = \{q_1, q_2\}$ da eta $\mathsf{Itxidura}\text{-}\varepsilon^{\#}(\{q_0, q_2\}, \lambda_1) = \{q_0, q_1, q_2\}$ da.

E_2 hartzen badugu, $\lambda_2(r_0, \varepsilon) = \varnothing$ da eta $\mathsf{Itxidura}\text{-}\varepsilon(r_0, \lambda_2) = \{r_0\}$ da. Bestalde, $\lambda_2(r_1, \varepsilon) = \{r_2\}$ da eta $\mathsf{Itxidura}\text{-}\varepsilon(r_1, \lambda_2) = \{r_1, r_2\}$ da. Azkenik, $\lambda_2(r_2, \varepsilon) = \varnothing$ da eta $\mathsf{Itxidura}\text{-}\varepsilon(r_2, \lambda_2) = \{r_2\}$. Aipatzekoa da baita ere $\mathsf{Itxidura}\text{-}\varepsilon^{\#}(\{r_1, r_2\}, \lambda) = \{r_1, r_2\}$ dela eta $\mathsf{Itxidura}\text{-}\varepsilon^{\#}(\{r_0, r_2\}, \lambda) = \{r_0, r_2\}$ dela.

E_4 kontsideratuz, $\lambda_4(t_0, \varepsilon) = \{t_0, t_1\}$ da eta $\mathsf{Itxidura}\text{-}\varepsilon(t_0, \lambda_4) = \{t_0, t_1, t_2\}$. Bestalde, $\lambda_4(t_1, \varepsilon) = \{t_2\}$ da eta $\mathsf{Itxidura}\text{-}\varepsilon(t_1, \lambda_4) = \{t_0, t_1, t_2\}$ da. Azkenik, $\lambda_4(t_2, \varepsilon) = \{t_0\}$ da eta $\mathsf{Itxidura}\text{-}\varepsilon(t_2, \lambda_4) = \{t_0, t_1, t_2\}$ da. Kasu honetan ere aipagarria da $\mathsf{Itxidura}\text{-}\varepsilon^{\#}(\{t_1, t_2\}, \lambda_4) = \{t_0, t_1, t_2\}$ dela eta, bere aldetik, $\mathsf{Itxidura}\text{-}\varepsilon^{\#}(\{t_0, t_2\}, \lambda_4) = \{t_0, t_1, t_2\}$ dela.

λ^* **konputazio-funtzioaren definizio formala** honako hau da:

- $\lambda^*(S, \varepsilon) = \mathsf{Itxidura}\text{-}\varepsilon^{\#}(S, \lambda)$
- $\lambda^*(\varnothing, w) = \varnothing$
- $\lambda^*(S, \alpha w) = \lambda^*(\bigcup_{q_j \in \mathsf{Itxidura}\text{-}\varepsilon^{\#}(S,\lambda)} \lambda(q_j, \alpha), w)$

hor, S egoera-multzo bat da, α A-ko sinbolo bat da eta w A^*-ko hitz bat da.

Definizio horren bidez λ^*-ren jokabidea zein den adierazten da. Sarreratzat emandako hitza ε hitz hutsa baldin bada, S-ko egoeretan eta S-ko egoera horietatik sinbolorik gabe irisgarriak diren egoeretan geratuko gara. Sarreratzat emandako S multzoa hutsa baldin bada, hau da, $\varnothing$, orduan ez gaude egoeraren batean eta ondorioz ezingo dugu beste egoeraren batera iritsi, eta hori $\varnothing$ egoera-multzo hutsera iritsiko garela esanez adierazten da. Bestalde, S multzo hutsa ez bada eta sarrerako hitza hutsa ez bada eta bere lehenengo osagaia (A-koa izango den sinboloa) α baldin bada eta gainerako osagaiek w azpihitza osatzen badute, ε trantsizioen bidez S multzotik irisgarriak diren egoerak hartu ($\mathsf{Itxidura}\text{-}\varepsilon^{\#}(S, \lambda)$) eta egoera horietako bakoitzari eta α sinboloari λ aplikatuko zaie. Egoera-multzo berrian kokatu ondoren, w hitzeko gainerako sinboloak irakurtzen jarraitu beharko da.

Adibidez, S multzoa $\{q_2, q_4, q_5, q_7\}$ dela eta $\mathsf{Itxidura}\text{-}\varepsilon^{\#}(S, \lambda)$ multzoa $\{q_2, q_4, q_5, q_6, q_7, q_{10}\}$ dela kontsideratzen badugu,

$$\lambda^*(\{q_2, q_4, q_5, q_7\}, \alpha w) = \lambda^*(\lambda(q_2, \alpha) \cup \lambda(q_4, \alpha) \cup \lambda(q_5, \alpha) \cup \lambda(q_6, \alpha) \cup \lambda(q_7, \alpha) \cup \lambda(q_{10}, \alpha), w)$$

Hor, $\lambda(q_2, \alpha)$ espresioak q_2 egoeratik α sinboloaren bidez irisgarriak diren egoerak zein diren adierazten du, $\lambda(q_4, \alpha)$ espresioak q_4 egoeratik α sinboloaren bidez irisgarriak diren egoerak zein diren adierazten du eta abar.

Konputazio bat **konfigurazio ez-deterministez eratutako sekuentzia** gisa adieraz daiteke, baina baita **konfigurazio deterministez eratutako zuhaitz** gisa ere. Hasierako konfigurazio ez-deterministako egoera-multzoan egoera bat baino gehiago daudenean, konputazio bat konfigurazio deterministen bidez adieraztea nahi izanez gero, egoera horietako bakoitzeko zuhaitz bat garatu beharko da, **konfigurazio deterministez eratutako zuhaitz-multzoa** lortuz.

ε-AFED bati dagokion konputazioa **konfigurazio deterministez eratutako zuhaitz baten bidez (edo zuhaitz-multzo baten bidez)** garatzean, adar bakoitzak gehienez bi adabegitan izan dezake konfigurazio bera. Beraz, adar baten garapena bi arrazoirengatik buka daiteke: λ trantsizio-funtzioa kontuan hartuz beste konfiguraziorik ezin delako sortu edo, bestela, ipini den adabegi berriak duen konfigurazioa adar bereko beste adabegi batean ere agertzen delako. Konputazio bat konfigurazio deterministez osatutako zuhaitz baten bidez edo zuhaitz-multzo baten bidez guztiz garatu ondoren, ε-AFEDak "Bai" erantzungo du baldin eta soilik baldin gutxienez adar batean Y multzokoa den q_j egoera batez eta ε hitzaz eratutako (q_j, ε) konfigurazio bat baldin badu. Ez da beharrezkoa (q_j, ε) konfigurazio determinista hori hostoa izatea. Konfigurazio deterministez osatutako zuhaitz baten bidez konputazio bat garatu ondoren, konputazio hori bukatutakoan zein egoeretara iritsi garen jakiteko irizpidea honako hau da: bigarren osagaitzat ε hitza duten konfigurazioetako egoerez osatutako egoera-multzoak adieraziko du zein egoeretan bukatu dugun.

3.4.1.1 atalean ε-AFED desberdinentzat konputazio batzuk garatu dira konfigurazio deterministez eratutako zuhaitzen bidez. Orain beste adibide bat erabiliz, alde batetik konputazio baten garapen formala landuko dugu eta, bestetik, konputazio hori konfigurazio ez-deterministez osatutako sekuentzia bezala eta konfigurazio deterministez osatutako zuhaitz-multzo bezala adieraziko dugu.

Demagun $A = \{a, b, c\}$ alfabetoa daukagula eta une honetako konfigurazioa $(\{q_1, q_4\}, caab)$ dela eta λ funtzioa honela definituta dagoela jarraian aipatzen diren kasuetarako:

- $\lambda(q_1, c) = \{q_2, q_6\}$
- $\lambda(q_1, \varepsilon) = \{q_8\}$
- $\lambda(q_2, a) = \varnothing$
- $\lambda(q_3, a) = \{q_7\}$
- $\lambda(q_4, c) = \{q_3, q_5, q_6\}$
- $\lambda(q_5, a) = \varnothing$
- $\lambda(q_5, b) = \{q_6\}$
- $\lambda(q_6, a) = \{q_5, q_6\}$
- $\lambda(q_6, b) = \{q_2, q_4\}$
- $\lambda(q_7, a) = \{q_7\}$
- $\lambda(q_7, b) = \{q_1, q_2\}$
- $\lambda(q_8, c) = \{q_{10}, q_{11}\}$
- $\lambda(q_{10}, \varepsilon) = \{q_{12}\}$

- $\lambda(q_{10}, a) = \{q_{10}\}$
- $\lambda(q_{10}, b) = \{q_{10}\}$
- $\lambda(q_{11}, a) = \varnothing$
- $\lambda(q_{12}, a) = \{q_{12}\}$
- $\lambda(q_{12}, b) = \{q_{13}\}$

Urratsez urrats $\lambda^*(\{q_1, q_4\}, caab)$ konputazioa honela kalkulatuko litzateke:

- $caab$ hitza hutsa ez denez eta Itxidura-$\varepsilon^{\#}(\{q_1, q_4\}, \lambda) = \{q_1, q_4, q_8\}$ denez,
  $\lambda^*(\{q_1, q_4\}, caab) = \lambda^*(\lambda(q_1, c) \cup \lambda(q_4, c) \cup \lambda(q_8, c), aab)$
  betetzen da.

- $\lambda(q_1, c) = \{q_2, q_6\}$ denez, $\lambda(q_4, c) = \{q_3, q_5, q_6\}$ eta $\lambda(q_8, c) = \{q_{10}, q_{11}\}$ betetzen direnez,

  $\lambda^*(\lambda(q_1, c) \cup \lambda(q_4, c) \cup \lambda(q_8, c), aab)$

  espresioa honela geldituko da $\lambda^*(\{q_2, q_3, q_5, q_6, q_{10}, q_{11}\}, aab)$.

- $\lambda(q_8, c)$ espresioak urrats bakar bat adierazten duenez, bere balioa $\{q_{10}, q_{11}\}$ da, eta ez $\{q_{10}, q_{11}, q_{12}\}$ nahiz eta q_{12} irisgarria izan q_{10} egoeratik ε trantsizio baten bidez.

- aab hitza hutsa ez denez eta Itxidura-$\varepsilon^{\#}(\{q_2, q_3, q_5, q_6, q_{10}, q_{11}\}, \lambda) = \{q_2, q_3, q_5, q_6, q_{10}, q_{11}, q_{12}\}$ denez, honako hau daukagu:

  $\lambda^*(\{q_2, q_3, q_5, q_6, q_{10}, q_{11}\}, aab) =$
  $= \lambda^*(\lambda(q_2, a) \cup \lambda(q_3, a) \cup \lambda(q_5, a) \cup \lambda(q_6, a) \cup \lambda(q_{10}, a) \cup \lambda(q_{11}, a) \cup \lambda(q_{12}, a), ab)$.

- $\lambda(q_2, a) = \varnothing$, $\lambda(q_3, a) = \{q_7\}$, $\lambda(q_5, a) = \varnothing$, $\lambda(q_6, a) = \{q_5, q_6\}$, $\lambda(q_{10}, a) = \{q_{10}\}$, $\lambda(q_{11}, a) = \varnothing$ eta $\lambda(q_{12}, a) = \{q_{12}\}$ direla kontuan hartuz,

  $\lambda^*(\lambda(q_2, a) \cup \lambda(q_3, a) \cup \lambda(q_5, a) \cup \lambda(q_6, a) \cup \lambda(q_{10}, a) \cup \lambda(q_{11}, a) \cup \lambda(q_{12}, a), ab)$

  espresiotik $\lambda^*(\{q_5, q_6, q_7, q_{10}, q_{12}\}, ab)$ espresioa lortzen da.

- ab hitza hutsa ez denez eta Itxidura-$\varepsilon^{\#}(\{q_5, q_6, q_7, q_{10}, q_{12}\}, \lambda) = \{q_5, q_6, q_7, q_{10}, q_{12}\}$ denez, honako hau daukagu:
  $\lambda^*(\{q_5, q_6, q_7, q_{10}, q_{12}\}, ab) = \lambda^*(\lambda(q_5, a) \cup \lambda(q_6, a) \cup \lambda(q_7, a) \cup \lambda(q_{10}, a) \cup \lambda(q_{12}, a), b)$.

- $\lambda(q_5, a) = \varnothing$, $\lambda(q_6, a) = \{q_5, q_6\}$, $\lambda(q_7, a) = \{q_7\}$, $\lambda(q_{10}, a) = \{q_{10}\}$ eta $\lambda(q_{12}, a) = \{q_{12}\}$ betetzen direnez, $\lambda^*(\lambda(q_5, a) \cup \lambda(q_6, a) \cup \lambda(q_7, a) \cup \lambda(q_{10}, a) \cup \lambda(q_{12}, a), b)$ espresiotik $\lambda^*(\{q_5, q_6, q_7, q_{10}, q_{12}\}, b)$ espresioa lortzen da.

- b hitza hutsa ez denez, eta Itxidura-$\varepsilon^{\#}(\{q_5, q_6, q_7, q_{10}, q_{12}\}, \lambda) = \{q_5, q_6, q_7, q_{10}, q_{12}\}$ denez, honako hau daukagu:
  $\lambda^*(\{q_5, q_6, q_7, q_{10}, q_{12}\}, b) = \lambda^*(\lambda(q_5, b) \cup \lambda(q_6, b) \cup \lambda(q_7, b) \cup \lambda(q_{10}, b) \cup \lambda(q_{12}, ab), \varepsilon)$.

- $\lambda(q_5, b) = \{q_6\}$, $\lambda(q_6, b) = \{q_2, q_4\}$, $\lambda(q_7, b) = \{q_1, q_2\}$, $\lambda(q_{10}, b) = \{q_{10}\}$ eta $\lambda(q_{12}, b) = \{q_{13}\}$ betetzen direnez, $\lambda^*(\lambda(q_5, b) \cup \lambda(q_6, b) \cup \lambda(q_7, b) \cup \lambda(q_{10}, b) \cup \lambda(q_{12}, b), \varepsilon)$ espresiotik $\lambda^*(\{q_1, q_2, q_4, q_6, q_{10}, q_{13}\}, \varepsilon)$ espresioa lortzen da.
- ε hitz hutsa geratzen denez, $\lambda^*(\{q_1, q_2, q_4, q_6, q_{10}, q_{13}\}, \varepsilon)$ espresioaren balioa $\mathsf{Itxidura}\text{-}\varepsilon^{\#}(\{q_1, q_2, q_4, q_6, q_{10}, q_{13}\}, \lambda)$ da, hau da, $\{q_1, q_2, q_4, q_6, q_8, q_{10}, q_{13}\}$.

Beraz, $\lambda^*(\{q_1, q_4\}, caab) = \{q_1, q_2, q_4, q_6, q_8, q_{10}, q_{13}\}$.

Konputazio hori **konfigurazio ez-deterministez osatutako** honako **sekuentzia** honen bidez adieraz daiteke:

$$
\begin{array}{c}
(\{q_1, q_4\}, caab) \equiv_{\mathsf{Itxidura}\text{-}\varepsilon^{\#}} (\{q_1, q_4, q_8\}, caab) \\
| \\
(\{q_2, q_3, q_5, q_6, q_{10}, q_{11}\}, aab) \equiv_{\mathsf{Itxidura}\text{-}\varepsilon^{\#}} (\{q_2, q_3, q_5, q_6, q_{10}, q_{11}, q_{12}\}, aab) \\
| \\
(\{q_5, q_6, q_7, q_{10}, q_{12}\}, ab) \equiv_{\mathsf{Itxidura}\text{-}\varepsilon^{\#}} (\{q_5, q_6, q_7, q_{10}, q_{12}\}, ab) \\
| \\
(\{q_5, q_6, q_7, q_{10}, q_{12}\}, b) \equiv_{\mathsf{Itxidura}\text{-}\varepsilon^{\#}} (\{q_5, q_6, q_7, q_{10}, q_{12}\}, b) \\
| \\
(\{q_1, q_2, q_4, q_6, q_{10}, q_{13}\}, \varepsilon) \equiv_{\mathsf{Itxidura}\text{-}\varepsilon^{\#}} (\{q_1, q_2, q_4, q_6, q_8, q_{10}, q_{13}\}, \varepsilon)
\end{array}
$$

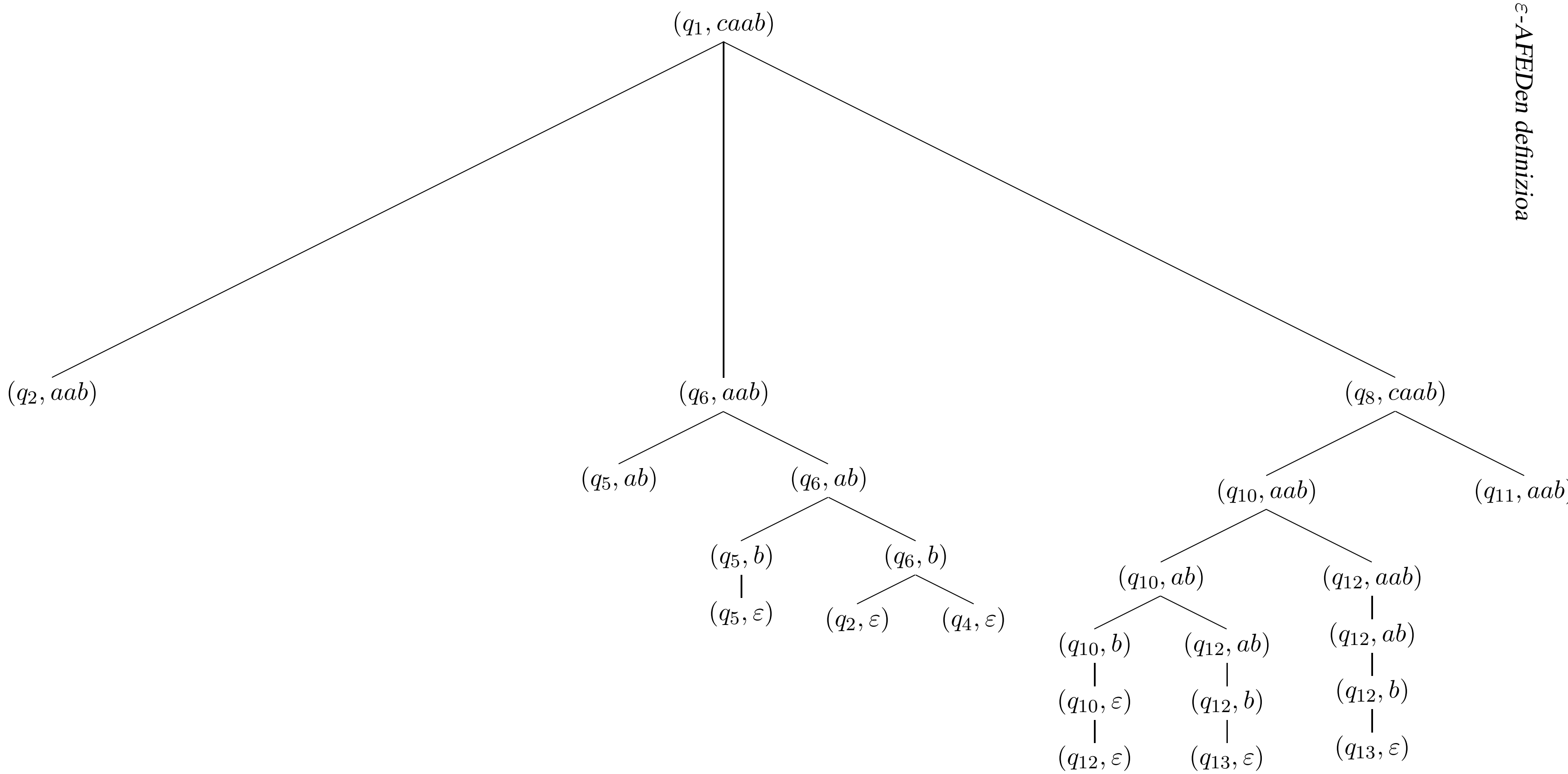
Lerro bakoitzean, $\equiv$ sinboloaren ezkerrean, une horretako egoera-multzoa zein den zehazten da eta $\equiv$ sinboloaren eskuinean, egoera horietatik ε trantsizioen bidez irisgarriak diren egoerak agertzen dira, hau da, $\mathsf{Itxidura}\text{-}\varepsilon^{\#}$.

Konfigurazio deterministak erabiliz ere adieraz daiteke konputazio hori. Abiapuntuko egoera-multzoak multzo bakarra baldin badu, konputazioa konfigurazio deterministez osatutako zuhaitz bakar baten bidez adieraziko da (3.4.1.1 ataleko adibideetan gertatzen den bezala). Aldiz, abiapuntuko egoera-multzoak multzo bat baino gehiago baldin baditu, konputazioa konfigurazio deterministez osatutako zuhaitz-multzo baten bidez adieraziko da: abiapuntuko egoera-multzoko egoera bakoitzeko zuhaitz bat, hain zuzen ere. Une honetan eskuetan darabilkigun adibidean abiapuntuko multzoak bi egoera dituenez, konfigurazio deterministez eratutako bi zuhaitz eraiki beharko dira. Zuhaitz horiek 3.4.3 irudian eta 3.4.4 irudian ikus daitezke.

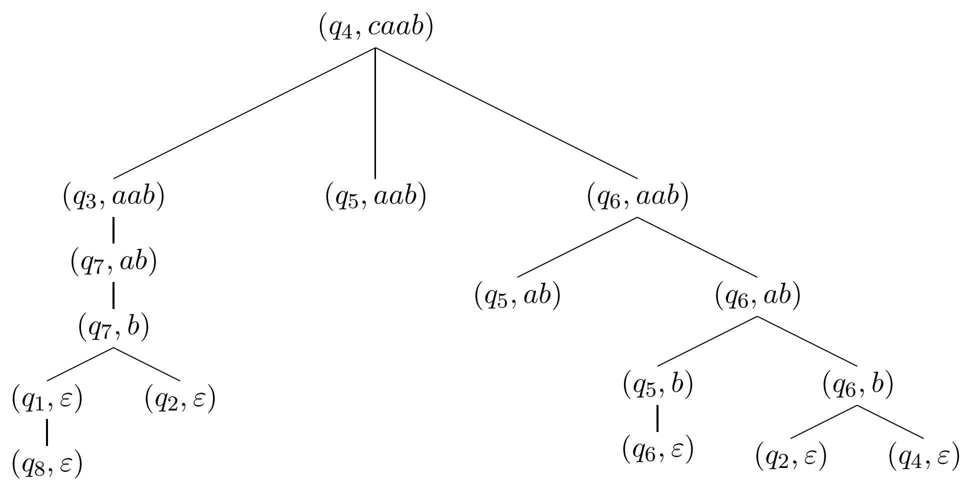
3.4.1.3.5 ε-AFED bati dagokion lengoaia

$E = (Q, A, \lambda, q_0, Y)$ ε-AFED bat emanda, **ε-AFED horri dagokion lengoaia**, ε-AFED horrek "Bai" erantzuten dien hitz guztiez osatuta dago. Beste era batera esanda, q_0 egoeratik abiatuz, osorik irakurri ondoren Y multzokoa den egoeraren bat duen egoera-multzo batean lagatzen gaituzten hitzez osatutako lengoaia da. E-ri dagokion lengoaia hori era formalean honela definituko genuke:

$$L(E) = \{w \mid w \in A^* \wedge (\lambda^*(\{q_0\}, w) \cap Y) \neq \varnothing\}$$



3.4.3 irudia. $\lambda^*(\{q_1, q_4\}, caab)$ konputazioari dagokion konfigurazio deterministen bidezko adierazpeneko lehenengo zuhaitza (3.4.1.3.4 atala).



3.4.4 irudia. $\lambda^*(\{q_1, q_4\}, caab)$ konputazioari dagokion konfigurazio deterministen bidezko adierazpeneko bigarren zuhaitza (3.4.1.3.4 atala).

3.4.1.4 ε-AFEDen adibideak

Jarraian, lengoaia desberdinei dagozkien ε-AFED batzuk aurkeztuko dira. Kasu bakoitzean konputazioak ere garatuko dira, bai definizio formala kontuan hartuz eta baita konfigurazio ez-deterministez eratutako sekuentzien bidez eta konfigurazio deterministez eratutako zuhaitzen bidez ere.

3.4.1.4.1 Adibidea: $\{\varepsilon\}$ lengoaiari dagokion ε-AFED bat

3.4.2 irudian aurkeztutako $E_4 = (T, A, \lambda_4, t_0, Y_4)$ ε-AFEDa hartuko dugu berriro. Automata hori boskote bezala 3.4.1.3 atalean definitu da. ε-AFED horrek a, b eta c sinboloak izan ditzaketen hitzak hartuko ditu sarrerako datu bezala. Datu bezala hartutako hitza hutsa baldin bada "Bai" erantzungo du eta bestela "Ez" erantzungo du. Beraz, hitza hutsa al den ala ez erabakiko du. Lengoaia hori formalki honela defini daiteke:

$$\{w \mid w \in A^* \wedge |w| = 0\}$$

Baina $\{\varepsilon\}$ bezala ere defini daiteke.

ε-AFED honek hiru egoera ditu: t_0, t_1 y t_2. Hasierako egoera t_0 da. Karaktere-kate bat irakurtzen hastean t_0 egoeran egongo gara. Edozein $j \in \{0, 1, 2\}$ eta edozein $\alpha \in A$ hartuz $\lambda(t_j, \alpha) = \varnothing$ betetzen denez, ezin da sinbolorik kontsumitu. Hala ere, ε trantsizioak erabiliz, egoera batetik bestera igaro gaitezke. Horregatik, bi zirkulu dituen t_2 egoerara sinbolo denak irakurrita iristeko, hasierako hitzak hutsa izan beharko du, hau da, ε. ε trantsizioak direla-eta, E_4 ε-AFEDko egoeretan zehar sinbolorik kontsumitu gabe era infinituan, hau da, bukaerarik gabe jarraitu daitekeela dirudiela pentsatu arren, λ^* konputazio-funtzioaren definizio formala kontuan hartuz (3.4.1.3.4 atala) konputazioak beti finituak izango direla ondoriozta dezakegu.

Jarraian $(\{t_0\}, aaa)$ konfigurazio ez-deterministari dagokion konputazioa garatuko da urratsez urrats:

1. urratsa: $\lambda^*(\{t_0\}, aaa) = \lambda^*(\lambda(t_0, a) \cup \lambda(t_1, a) \cup \lambda(t_2, a), aa) = \lambda^*(\varnothing, aa)$. Kasu honetan $\text{Itxidura-}\varepsilon^{\#}(\{t_0\}) = \{t_0, t_1, t_2\}$ dela kontuan hartu da.

2. urratsa: $\lambda^*(\varnothing, aa) = \varnothing$.

Konputazioa $\varnothing$ multzoan bukatu denez eta multzo horretan Y multzoko elementurik ez dagoenez, hau da, $\varnothing \cap Y = \varnothing$, badakigu hitza ez dela lengoaiakoa.

Konputazio hori konfigurazio ez-deterministez eratutako honako sekuentzia honen bidez adieraz daiteke grafikoki:

$$\begin{array}{c} (\{t_0\}, aaa) \equiv_{\text{Itxidura-}\varepsilon^{\#}} (\{t_0, t_1, t_2\}, aaa) \\ | \\ (\varnothing, aa) \equiv_{\text{Itxidura-}\varepsilon^{\#}} (\varnothing, aa) \\ \\ \text{Ez} \end{array}$$

Lerro bakoitzean, $\equiv$ sinboloaren ezkerrean, une horretako egoera-multzoa zein den zehazten da eta $\equiv$ sinboloaren eskuinean, egoera horietatik ε trantsizioen bidez irisgarriak diren egoerak agertzen dira, hau da, Itxidura-$\varepsilon^{\#}$. Konputazioa egoera-multzo hutsa duen konfigurazio batera iritsi denez, bukatu egin da. Erantzuna ezezkoa izango da azkenenko konfigurazioko egoera-multzoa hutsa delako eta bertan Y-ko elementurik ez dagoelako.

Konputazio hori konfigurazio determinististez eratutako honako zuhaitz honen bidez ere adieraz daiteke:

$$(t_0, aaa)$$

(t_0, aaa) $\quad$ (t_1, aaa)

Ez $\quad$ (t_2, aaa)

(t_0, aaa)

Ez

3.4.1.3.4 atalean zehaztu denari jarraituz, (Q, A, λ, q_0, Y) ε-AFED bati dagokion konputazioa konfigurazio determinististez eratutako zuhaitz baten bidez (edo zuhaitz-multzo baten bidez) garatzean, adar bakoitzak gehienez bi adabegitan izan dezake konfigurazio bera. Beraz, adar baten garapena bi arrazoirengatik buka daiteke: λ trantsizio-funtzioa kontuan hartuz beste konfiguraziorik ezin delako sortu edo, bestela, ipini den adabegi berriak duen konfigurazioa adar bereko beste adabegi batean ere agertzen delako. Konputazio bat konfigurazio determinististez osatutako zuhaitz baten bidez edo zuhaitz-multzo baten bidez guztiz garatu ondoren, ε-AFEDak "Bai" erantzungo du baldin eta soilik baldin gutxienez adar batean Y multzokoa den q_j egoera batez eta ε hitzaz eratutako (q_j, ε) konfigurazio bat baldin badu. Ez da beharrezkoa (q_j, ε) konfigurazio determinista hori hostoa izatea. Garatu den konputazioan ez da lortu baldintza hori betetzen duen konfiguraziorik. Gehiago zehaztuz, ez da lortu (t_2, ε) konfigurazioa eta, ondorioz, ε-AFEDak "Ez" erantzungo du. Bestalde, konfigurazio determinististez osatutako zuhaitz baten bidez konputazio bat garatu ondoren, konputazio hori bukatutakoan zein egoeretara iritsi garen jakiteko irizpidea honako hau da: bigarren osagaitzat ε hitza duten konfigurazioetako egoerez osatutako egoera-multzoak adieraziko du zein egoeretan bukatu dugun. Kasu honetan, multzo hutsean bukatu dugu.

Jarraian $(\{t_0\}, \varepsilon)$ konfigurazio ez-deterministari dagokion konputazioa garatuko da urratsez urrats:

1. urratsa: $\lambda^*(\{t_0\}, \varepsilon) = \mathsf{Itxidura}\text{-}\varepsilon^{\#}(\{t_0\}) = \{t_0, t_1, t_2\}$.

Lortutako egoera-multzoan Y_4-ko egoera bat badagoenez, ε hitza E_4 ε-AFEDaren bidez definitutako lengoaiakoa dela ondoriozta dezakegu.

Konputazio hori konfigurazio ez-determinististez eratutako honako sekuentzia honen bidez adieraz daiteke grafikoki:

$$(\{t_0\}, \varepsilon) \equiv_{\text{Itxidura-}\varepsilon^{\#}} (\{t_0, t_1, t_2\}, \varepsilon)$$

Bai

Lortu den adabegi bakarrean, $\equiv$ sinboloaren ezkerrean, une horretako egoera-multzoa zein den zehazten da eta $\equiv$ sinboloaren eskuinean, egoera horietatik ε trantsizioen bidez irisgarriak diren egoerak agertzen dira, hau da, Itxidura-$\varepsilon^{\#}$. Sekuentzia hori hitz hutsa duen konfigurazio bat lortu delako bukatu da. ε-AFEDak baiezko erantzuna emango du azkeneko konfigurazioan (kasu honetan, lortu den konfigurazio bakarrean) egoera-multzoa $\{t_0, t_1, t_2\}$ delako eta multzo horretan t_2 dagoelako.

Konputazio hori konfigurazio deterministez eratutako honako zuhaitz honen bidez ere adieraz daiteke:

$$\begin{array}{ccc}
 & (t_0, \varepsilon) & \\
(t_0, \varepsilon) & & (t_1, \varepsilon) \\
 & & | \\
\text{Ez} & & (t_2, \varepsilon) \\
 & & | \\
 & & (t_0, \varepsilon)
\end{array}$$

Bai

Zuhaitzak bi adar ditu eta adar biak adabegi errepikatuak agertu direlako bukatu dira. Ezkerreko adarrak ez du Y_4 multzokoa den egoera batez eta ε hitz hutsaz osatutako konfiguraziorik. Aldiz, eskuineko adarrak Y_4 multzokoa den egoera batez eta ε hitz hutsaz osatutako konfigurazio bat du. Konfigurazio hori (t_2, ε) da. Adibide honetak ikus daiteke ez dela beharrezkoa konfigurazio hori hostoa izatea.

3.4.1.4.2 Adibidea: A^* lengoaia unibertsala definitzen duen ε-AFEDa

Har dezagu berriro ere 3.4.2 irudian aurkeztutako $E_6 = (M, A, \lambda_6, m_0, Y_6)$ ε-AFEDa. Automata hori boskote gisa 3.4.1.3 atalean formalizatu da. a, b eta c sinboloez osatutako hitz bat jasoko du adibide honetako ε-AFEDak. Edozein hitzentzat erantzuna baiezkoa izango da, ε-AFED hori A^* lengoaia unibertsalari dagokiona baita. Formalki, lengoaia era desberdinetan adieraz daiteke. Adibidez:

$$\{w \mid w \in A^* \wedge w = w\}$$

Beste aukera bat honako hau izango litzateke:

$$\{w \mid w \in A^* \wedge |w| = |w|\}$$

Hirugarren aukera bat honako hau izango litzateke:

$$(\{a\} \cup \{b\} \cup \{c\})^*$$

$\{a, b, c\}^*$ eran edo A^* eran ere defini daiteke lengoaia unibertsala.

E_6 ε-AFEDak hiru egoera ditu: m_0, m_1 eta m_2. Hasierako egoera m_0 da eta onarpen egoera bakarra m_2 da. Hitz bat irakurtzen hasterakoan m_0 egoeran egongo gara. Irakurri beharreko hitzeko lehenengo sinboloa a baldin bada, bi aukera daude: a irakurri (kontsumitu) eta m_0 egoeran gelditu edo a kontsumitu gabe m_1 egoerara joan. Hitzaren lehenengo sinboloa b edo c baldin bada, aukera bakarra m_1 egoerara igarotzea da. m_1 egoeran gaudenean, b baten aurrean b hori kontsumitu eta m_1-en gera gaitezke edo b kontsumitu gabe m_2 egoerara joan gaitezke. m_1 egoeran a eta c sinboloekin aukera bakarra sinboloa irakurri gabe m_2 egoerara igarotzea da. m_2 egoeran antzekoa gertatzen da baina alde batetik c eta bestetik a eta b-rekin: zehazki, c baldin badugu, c hori kontsumitu eta m_2 egoeran gera gaitezke edo$c$ hori kontsumitu gabe m_0-ra igaro gaitezke; a-rekin eta b-rekin aukera bakarra sinboloa kontsumitu gabe m_0-ra igarotzea da.

w hitz bat emanda, E_6 ε-AFEDak "Bai" erantzungo du azkeneko konfigurazioa (S, ε) erakoa baldin bada eta $m_2 \in S$ betetzen bada. Edozein hitz hartuta, hitzari dagokion konputazioak baldintza hori beteko du.

Jarraian $(\{m_0\}, aaa)$ konfigurazio ez-deterministari dagokion konputazioa garatuko da urratsez urrats:

1. urratsa: $\lambda^*(\{m_0\}, aaa) = \lambda^*(\lambda(m_0, a) \cup \lambda(m_1, a) \cup \lambda(m_2, a), aa) = \lambda^*(\{m_0\}, aa)$. Kalkulu hori egitean Itxidura-$\varepsilon^\#(\{m_0\}) = \{m_0, m_1, m_2\}$ dela kontuan izan da.

2. urratsa: $\lambda^*(\{m_0\}, aa) = \lambda^*(\lambda(m_0, a) \cup \lambda(m_1, a) \cup \lambda(m_2, a), a) = \lambda^*(\{m_0\}, a)$. Kasu honetan ere Itxidura-$\varepsilon^\#(\{m_0\}) = \{m_0, m_1, m_2\}$ dela kontuan hartu da.

3. urratsa: $\lambda^*(\{m_0\}, a) = \lambda^*(\lambda(m_0, a) \cup \lambda(m_1, a) \cup \lambda(m_2, a), \varepsilon) = \lambda^*(\{m_0\}, \varepsilon)$. Hirugarrengo aldiz, Itxidura-$\varepsilon^\#(\{m_0\}) = \{m_0, m_1, m_2\}$ dela kontuan hartu da.

4. urratsa: $\lambda^*(\{m_0\}, \varepsilon) = \{m_0, m_1, m_2\}$. Izan ere, Itxidura-$\varepsilon^\#(\{m_0\}) = \{m_0, m_1, m_2\}$.

Konputazioa $\{m_0, m_1, m_2\}$ multzoan bukatu da eta multzo horrek badu Y_6-koa den elementu bat, hau da, $\{m_0, m_1, m_2\} \cap Y_6 \neq \varnothing$ betetzen da eta, ondorioz, hitza lengoaiakoa da.

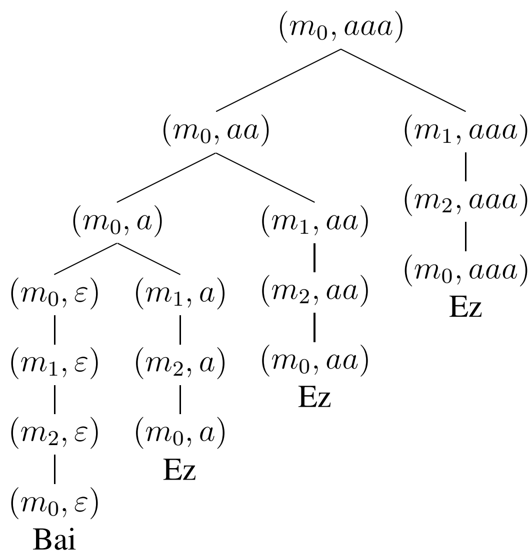
Konputazio hori konfigurazio ez-deterministez eratutako honako sekuentzia honen bidez adieraz daiteke grafikoki:

$$\begin{array}{rcl}
(\{m_0\}, aaa) \equiv_{\text{Itxidura-}\varepsilon^\#} & (\{m_0, m_1, m_2\}, aaa) \\
 & | \\
(\{m_0\}, aa) \equiv_{\text{Itxidura-}\varepsilon^\#} & (\{m_0, m_1, m_2\}, aa) \\
 & | \\
(\{m_0\}, a) \equiv_{\text{Itxidura-}\varepsilon^\#} & (\{m_0, m_1, m_2\}, a) \\
 & | \\
(\{m_0\}, \varepsilon) \equiv_{\text{Itxidura-}\varepsilon^\#} & (\{m_0, m_1, m_2\}, \varepsilon)
\end{array}$$

Bai

Lerro bakoitzean, $\equiv$ sinboloaren ezkerrean, une horretako egoera-multzoa zein den zehazten da eta $\equiv$ sinboloaren eskuinean, egoera horietatik ε trantsizioen bidez irisgarriak diren egoerak agertzen dira, hau da, Itxidura-$\varepsilon^{\#}$. ε hitz hutsa agertu delako bukatu da sekuentzia. Azkeneko konfigurazioan m_2 egoera agertzen denez, erantzuna baiezkoa izango dela ondorioztа dezakegu.

Konputazio hori konfigurazio deterministez eratutako honako zuhaitz honen bidez ere adieraz daiteke:



Adar bakoitza bere azkeneko adabegia adarrean bertan errepikatuta dagoelako bukatu da. Ezkerreko adarrak hitz hutsaz eta Y_6-koa den egoera batez osatutako konfigurazio bat duenez, ε-AFEDak baiezko erantzuna emango du. Konfigurazio hori (m_2, ε) da. Aurretik ere esan denez, ez da beharrezkoa konfigurazio hori hostoa izatea.

Jarraian $(\{m_0\}, abac)$ konfigurazio ez-deterministari dagokion konputazioa garatuko da urratsez urrats:

1. urratsa: $\lambda^*(\{m_0\}, abac) = \lambda^*(\lambda(m_0, a) \cup \lambda(m_1, a) \cup \lambda(m_2, a), bac) = \lambda^*(\{m_0\}, bca)$. Urrats honetan Itxidura-$\varepsilon^{\#}(\{m_0\}) = \{m_0, m_1, m_2\}$ dela hartu da kontuan.

2. urratsa: $\lambda^*(\{m_0\}, bac) = \lambda^*(\lambda(m_0, b) \cup \lambda(m_1, b) \cup \lambda(m_2, b), ac) = \lambda^*(\{m_1\}, ac)$. Kalkulu hori egitean Itxidura-$\varepsilon^{\#}(\{m_0\}) = \{m_0, m_1, m_2\}$ dela izan da kontuan.

3. urratsa: $\lambda^*(\{m_1\}, ac) = \lambda^*(\lambda(m_0, a) \cup \lambda(m_1, a) \cup \lambda(m_2, a), c) = \lambda^*(\{m_0\}, c)$. Urrats honetan Itxidura-$\varepsilon^{\#}(\{m_1\}) = \{m_0, m_1, m_2\}$ dela kontuan hartu da.

4. urratsa: $\lambda^*(\{m_0\}, c) = \lambda^*(\lambda(m_0, c) \cup \lambda(m_1, c) \cup \lambda(m_2, c), \varepsilon) = \lambda^*(\{m_2\}, \varepsilon)$. Kasu honetan Itxidura-$\varepsilon^{\#}(\{m_0\}) = \{m_0, m_1, m_2\}$ daukagu.

5. urratsa: $\lambda^*(\{m_2\}, \varepsilon) = \{m_0, m_1, m_2\}$. Izan ere, Itxidura-$\varepsilon^{\#}(\{m_2\}) = \{m_0, m_1, m_2\}$ da.

Lortutako egoera-multzoan Y_6-koa den egoera bat badagoenez, hitza E_6 ε-AFED dagokion lengoaiakoa dela ondorioztа dezakegu.

Konputazio hori konfigurazio ez-deterministez eratutako honako sekuentzia honen bidez adieraz daiteke grafikoki:

$$\begin{array}{rcl}
(\{m_0\}, abac) & \equiv_{\mathsf{Itxidura}\text{-}\varepsilon^{\#}} & (\{m_0, m_1, m_2\}, abac) \\
& | & \\
(\{m_0\}, bac) & \equiv_{\mathsf{Itxidura}\text{-}\varepsilon^{\#}} & (\{m_0, m_1, m_2\}, bac) \\
& | & \\
(\{m_1\}, ac) & \equiv_{\mathsf{Itxidura}\text{-}\varepsilon^{\#}} & (\{m_0, m_1, m_2\}, ac) \\
& | & \\
(\{m_0\}, c) & \equiv_{\mathsf{Itxidura}\text{-}\varepsilon^{\#}} & (\{m_0, m_1, m_2\}, c) \\
& | & \\
(\{m_2\}, \varepsilon) & \equiv_{\mathsf{Itxidura}\text{-}\varepsilon^{\#}} & (\{m_0, m_1, m_2\}, \varepsilon) \\
& \text{Bai} &
\end{array}$$

Lerro bakoitzean, $\equiv$ sinboloaren ezkerrean, une horretako egoera-multzoa zein den zehazten da eta $\equiv$ sinboloaren eskuinean, egoera horietatik ε trantsizioen bidez irisgarriak diren egoerak agertzen dira, hau da, $\mathsf{Itxidura}$-$\varepsilon^{\#}$. Konputazioa ε hitza duen konfigurazio ez-determinista batera iritsi delako bukatu da.

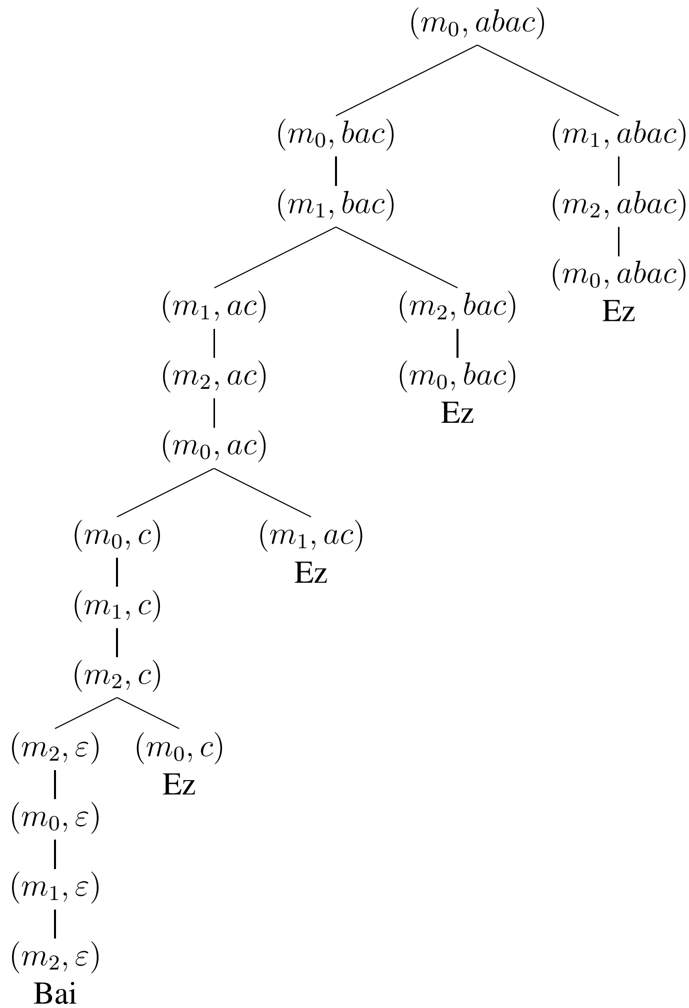
Konputazio hori 3.4.5 irudian erakusten den konfigurazio deterministez eratutako zuhaitzaren bidez ere adieraz daiteke.

adar bakoitza, adarreko azkeneko konfigurazioa adarrean bertan errepikatuta dagoelako bukatu da. ε-AFEDak “Bai” erantzungo du adar batean (m_2, ε) konfigurazioa agertzen delako. Kasu honetan konfigurazio hori hosto gisa eta barneko adabegi gisa agertzen da.

3.4.1.4.3 $(\{a\}^+\{b\}^*\{c\}^*)^+$ lengoaiari dagokion ε-AFEDa

Har dezagun $E_5 = (P, A, \lambda_5, p_0, Y_5)$ ε-AFEDa. Automata horren trantsizio-diagrama 3.4.2 irudian dugu eta boskote gisa 3.4.1.3 atalean formalizatu da. E_5 ε-AFEDak a, b eta c sinboloez osatutako hitz bat hartuko du sarrerako datutzat. Lengoaia formaliki honela adieraz daiteke: $(\{a\}^+\{b\}^*\{c\}^*)^+$.

E_5 ε-AFEDak hiru egoera ditu: p_0, p_1 eta p_2. Hasierako egoera p_0 da eta onarpen egoera bakarra p_2 da. Hitz bat irakurtzen hasterakoan p_0 egoeran egongo gara. p_0 egoeran a baten aurrean bi aukera ditugu: sinbolo hori kontsumitu eta p_0 egoeran gelditu edo p_1 egoerara igaro a irakurri gabe. p_0 egoeran b eta c-rekin ezin da ezer egin. p_1 egoeran, b baldin badugu, bi aukera ditugu: sinbolo hori kontsumitu eta p_1-en gelditu edo pasar sinboloa irakurri gabe p_2-ra igaro. a eta c sinboloekin aukera bakarra p_2 egoerara igarotzea da. p_2 egoeran c-rekin bi aukera daude: c kontsumitu eta p_2 egoerara igaro edo p_0-ra joan c kontsumitu gabe. a eta b sinboloekin aukera bakarra sinboloa kontsumitu gabe p_0-ra joatea da.



3.4.5 irudia. $\lambda^*(\{m_0\}, abac)$ konputazioari dagokion konfigurazio deterministen bidezko zuhaitza (3.4.1.4.2 atala).

w hitz bat emanda, E_5 ε-AFEDak "Bai" erantzungo du bukaerako konfigurazioa (S, ε) erakoa baldin bada eta $p_2 \in S$ betetzen bada.

Jarraian $(\{p_0\}, aaa)$ konfigurazio ez-deterministari dagokion konputazioa garatuko da urratsez urrats:

1. urratsa: $\lambda^*(\{p_0\}, aaa) = \lambda^*(\lambda(p_0, a), aa) = \lambda^*(\{p_0, p_1\}, aa)$. Kalkulu horretan $\mathsf{Itxidura}\text{-}\varepsilon^{\#}(\{p_0\}) = \{p_0\}$ dela kontuan hartu da.

2. urratsa: $\lambda^*(\{p_0, p_1\}, aa) = \lambda^*(\lambda(p_0, a) \cup \lambda(p_1, a) \cup \lambda(p_2, a), a) = \lambda^*(\{p_0, p_1\}, a)$. Urrats honetan $\mathsf{Itxidura}\text{-}\varepsilon^{\#}(\{p_0, p_1\}) = \{p_0, p_1, p_2\}$ betetzen dela kontuan izan da.

3. urratsa: $\lambda^*(\{p_0, p_1\}, a) = \lambda^*(\lambda(p_0, a) \cup \lambda(p_1, a) \cup \lambda(p_2, a), \varepsilon) = \lambda^*(\{p_0, p_1\}, \varepsilon)$.

Urrats honetan ere Cierre-$\varepsilon^{\#}(\{p_0, p_1\}) = \{p_0, p_1, p_2\}$ daukagu.

4. urratsa: $\lambda^*(\{p_0, p_1\}, \varepsilon) = \{p_0, p_1, p_2\}$. Berriro ere Itxidura-$\varepsilon^{\#}(\{p_0, p_1\}) = \{p_0, p_1, p_2\}$ dela kontuan hartu da.

Konputazioa $\{p_0, p_1, p_2\}$ multzoan bukatu denez eta gutxienez multzo horretako elementu bat Y_5-ekoa denez, hau da, $\{p_0, p_1, p_2\} \cap Y_5 \neq \varnothing$ denez, hitza lengoaiakoa dela ondoriozta dezakegu.

Konputazio hori konfigurazio ez-deterministez eratutako honako sekuentzia honen bidez adieraz daiteke grafikoki:

$$\begin{array}{c}
(\{p_0\}, aaa) \equiv_{\text{Itxidura-}\varepsilon^{\#}} (\{p_0\}, aaa) \\
| \\
(\{p_0, p_1\}, aa) \equiv_{\text{Itxidura-}\varepsilon^{\#}} (\{p_0, p_1, p_2\}, aa) \\
| \\
(\{p_0, p_1\}, a) \equiv_{\text{Itxidura-}\varepsilon^{\#}} (\{p_0, p_1, p_2\}, a) \\
| \\
(\{p_0, p_1\}, \varepsilon) \equiv_{\text{Itxidura-}\varepsilon^{\#}} (\{p_0, p_1, p_2\}, \varepsilon)
\end{array}$$

Bai

Lerro bakoitzean, $\equiv$ sinboloaren ezkerrean, une horretako egoera-multzoa zein den zehazten da eta $\equiv$ sinboloaren eskuinean, egoera horietatik ε trantsizioen bidez irisgarriak diren egoerak agertzen dira, hau da, Itxidura-$\varepsilon^{\#}$. Konputazioa ε hitza duen konfigurazio ez-determinista batera iritsi delako bukatu da. Lortu den azkeneko egoera-multzoan Y_5-eko egoera bat badagoenez, p_2 egoera hain zuzen ere, ε-AFEDak baiezko erantzuna emango duela ondoriozta dezakegu.

Konputazio hori 3.4.6 irudian erakusten den konfigurazio deterministez eratutako zuhaitzaren bidez ere adieraz daiteke.

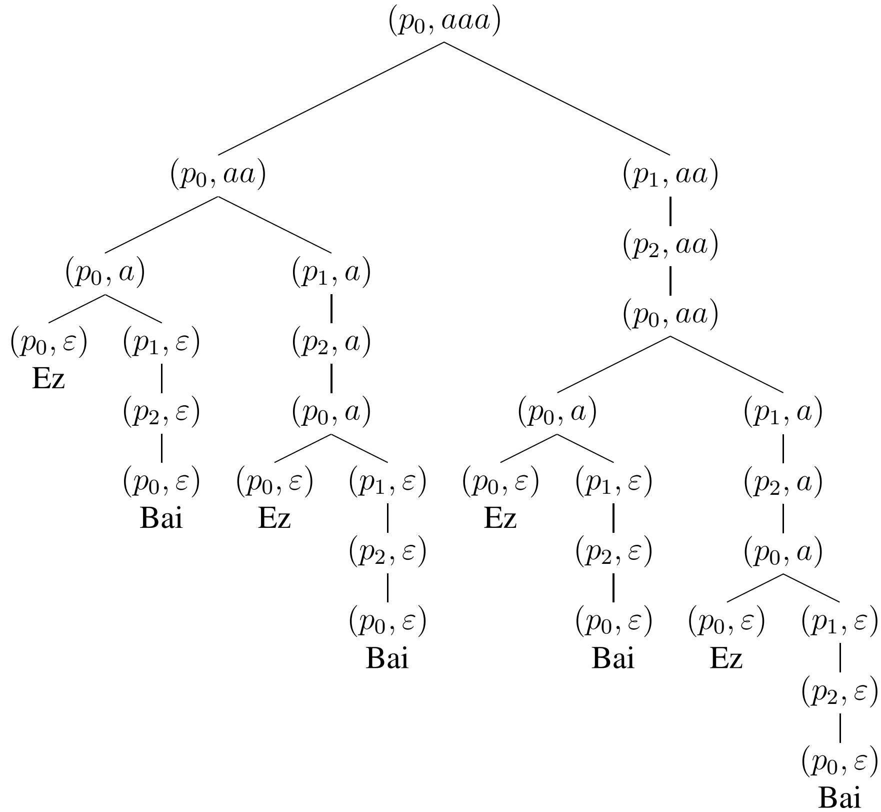
Kasu honetan, λ_5 kontuan hartuz beste konfiguraziorik ezin delako lortu bukatu da adar bakoitza. Beraz, adar bat bera ere ez da bukatu konfigurazio bat adarrean bertan errepikatu delako. ε-AFEDak baiezko erantzuna emango du hitz hutsaz eta Y_5-ekoa den egoera batez osatutako konfigurazio bat agertzen delako: (p_2, ε) konfigurazioa hain zuzen ere. Konfigurazio hori ezkerretik hasita bigarren, laugarren, seigarren eta zortzigarren adarretan agertzen da.

Jarraian $(\{p_0\}, abac)$ konfigurazio ez-deterministari dagokion konputazioa garatuko da urratsez urrats:

1. urratsa: $\lambda^*(\{p_0\}, abac) = \lambda^*(\lambda(p_0, a), bac) = \lambda^*(\{p_0, p_1\}, bac)$. Urrats honetan Itxidura-$\varepsilon^{\#}(\{p_0\}) = \{p_0\}$ dela kontuan izan da.

2. urratsa: $\lambda^*(\{p_0, p_1\}, bac) = \lambda^*(\lambda(p_0, b) \cup \lambda(p_1, b) \cup \lambda(p_2, b), ac) = \lambda^*(\{p_1\}, ac)$. Kalkulu horretan Itxidura-$\varepsilon^{\#}(\{p_0, p_1\}) = \{p_0, p_1, p_2\}$ dela kontuan hartu da.

3. urratsa: $\lambda^*(\{p_1\}, ac) = \lambda^*(\lambda(p_0, a) \cup \lambda(p_1, a) \cup \lambda(p_2, a), c) = \lambda^*(\{p_0, p_1\}, c)$. Kalkulu hori egitean Itxidura-$\varepsilon^{\#}(\{p_1\}) = \{p_0, p_1, p_2\}$ daukagu.



3.4.6 irudia. $\lambda^*(\{p_0\}, aaa)$ konputazioari dagokion konfigurazio deterministen bidezko zuhaitza (3.4.1.4.3 atala).

4. urratsa: $\lambda^*(\{p_0, p_1\}, c) = \lambda^*(\lambda(p_0, c) \cup \lambda(p_1, c) \cup \lambda(p_2, c), \varepsilon) = \lambda^*(\{p_2\}, \varepsilon)$. Hemen Itxidura-$\varepsilon^{\#}(\{p_0, p_1\}) = \{p_0, p_1, p_2\}$ daukagu.

5. urratsa: $\lambda^*(\{p_2\}, \varepsilon) = \{p_0, p_2\}$. Izan ere, Itxidura-$\varepsilon^{\#}(\{p_2\}) = \{p_0, p_2\}$ da.

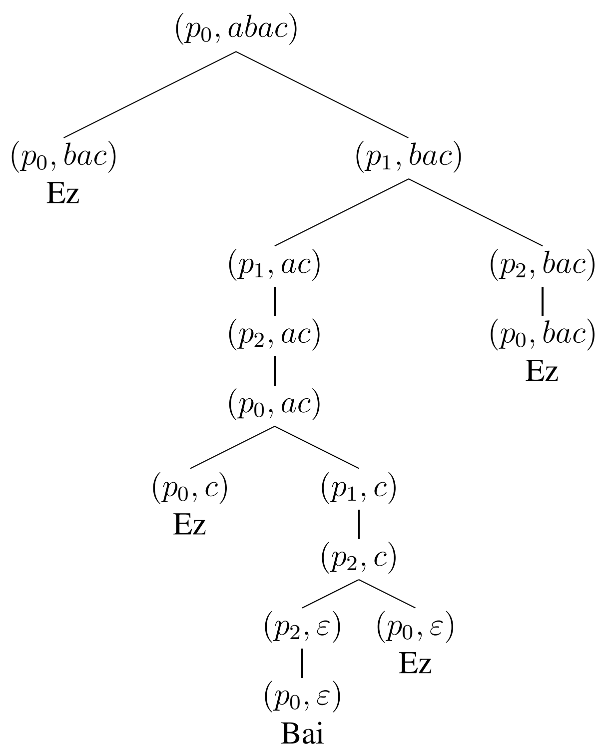
Lortu den egoera-multzoan badagoenez Y_5-ekoa den elementu bat, hitza E_5 ε-AFEDari dagokion lengoaiakoa dela ondorioztа dezakegu.

Konputazio hori konfigurazio ez-deterministez eratutako honako sekuentzia honen bidez adieraz daiteke grafikoki:

$$\begin{array}{c}
(\{p_0\}, abac) \equiv_{\mathsf{Itxidura}\text{-}\varepsilon^{\#}} (\{p_0\}, abac) \\
| \\
(\{p_0, p_1\}, bac) \equiv_{\mathsf{Itxidura}\text{-}\varepsilon^{\#}} (\{p_0, p_1, p_2\}, bac) \\
| \\
(\{p_1\}, ac) \equiv_{\mathsf{Itxidura}\text{-}\varepsilon^{\#}} (\{p_0, p_1, p_2\}, ac) \\
| \\
(\{p_0, p_1\}, a) \equiv_{\mathsf{Itxidura}\text{-}\varepsilon^{\#}} (\{p_0, p_1, p_2\}, c) \\
| \\
(\{p_2\}, \varepsilon) \equiv_{\mathsf{Itxidura}\text{-}\varepsilon^{\#}} (\{p_0, p_1\}, \varepsilon) \\
\text{Bai}
\end{array}$$

Lerro bakoitzean, $\equiv$ sinboloaren ezkerrean, une horretako egoera-multzoa zein den zehazten da eta $\equiv$ sinboloaren eskuinean, egoera horietatik ε trantsizioen bidez irisgarriak diren egoerak agertzen dira, hau da, Itxidura-$\varepsilon^{\#}$. Konputazioa ε hitza duen konfigurazio ez-determinista batera iritsi delako bukatu da. Azkeneko konfigurazioko egoera-multzoan Y_5-eko egoera bat dagoenez, p_2 egoera hain zuzen ere, erantzuna baiezkoa izango da.

Konputazio hori konfigurazio deterministez eratutako honako zuhaitz honen bidez ere adieraz daiteke:



Ezkerretik hasita, lehenengo, bigarrengo eta bosgarrengo adarretan ezin da aurrera egin λ_5 kontuan hartuz. Adar horietan ez da lortu hitz osoa irakurtzea. Laugarren adarrean ezin da aurrera egin λ_5 kontuan hartuz baina hitz osoa irakurtzea lortu da. Hala ere ez da agertzen (p_2, ε) konfigurazioa. Hirugarren adarrean ere ezin da gehiago aurrera egin λ_5 kontuan hartuz. Dena den, hirugarrengo adarrean (p_2, ε) agertzen da, nahiz eta hostoa ez izan, eta, ondorioz,

ε-AFEDak baiezko erantzuna emango du. Adarretako bat bera ere ez da bukatu adabegi bat adarrean bertan errepikatu delako.

Jarraian $(\{p_0\}, acba)$ konfigurazio ez-deterministari dagokion konputazioa garatuko da urratsez urrats:

1. urratsa: $\lambda^*(\{p_0\}, acba) = \lambda^*(\lambda(p_0, a), bca) = \lambda^*(\{p_0, p_1\}, cba)$. Kalkulu horretan $\text{Itxidura-}\varepsilon^{\#}(\{p_0\}) = \{p_0\}$ dela kontuan hartu da.

2. urratsa: $\lambda^*(\{p_0, p_1\}, cba) = \lambda^*(\lambda(p_0, c) \cup \lambda(p_1, c) \cup \lambda(p_2, c), ba) = \lambda^*(\{p_2\}, ba)$. Urrats honetan $\text{Itxidura-}\varepsilon^{\#}(\{p_0, p_1\}) = \{p_0, p_1, p_2\}$ daukagu.

3. urratsa: $\lambda^*(\{p_2\}, ba) = \lambda^*(\lambda(p_0, b) \cup \lambda(p_2, b), a) = \lambda^*(\varnothing, a)$. Izan ere, $\text{Itxidura-}\varepsilon^{\#}(\{p_2\}) = \{p_0, p_2\}$ da.

4. urratsa: $\lambda^*(\varnothing, a) = \varnothing$.

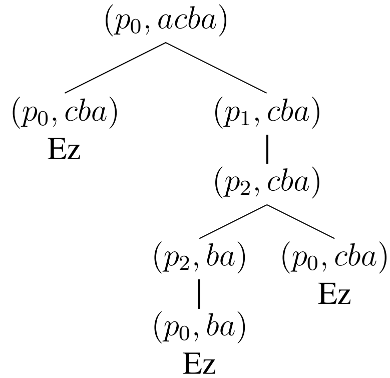
Lortutako egoera-multzoan Y_5-eko elementurik ez dagoenez, hitza ez da E_5 ε-AFEDari dagokion lengoaiakoa.

Konputazio hori konfigurazio ez-deterministez eratutako honako sekuentzia honen bidez adieraz daiteke grafikoki:

$$
\begin{array}{c}
(\{p_0\}, acba) \equiv_{\text{Itxidura-}\varepsilon^{\#}} (\{p_0\}, acba) \\
| \\
(\{p_0, p_1\}, cba) \equiv_{\text{Itxidura-}\varepsilon^{\#}} (\{p_0, p_1, p_2\}, cba) \\
| \\
(\{p_2\}, ba) \equiv_{\text{Itxidura-}\varepsilon^{\#}} (\{p_0, p_2\}, ba) \\
| \\
(\varnothing, a) \equiv_{\text{Itxidura-}\varepsilon^{\#}} (\varnothing, a) \\
\text{Ez}
\end{array}
$$

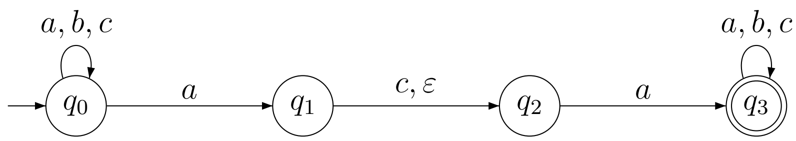
Lerro bakoitzean, $\equiv$ sinboloaren ezkerrean, une horretako egoera-multzoa zein den zehazten da eta $\equiv$ sinboloaren eskuinean, egoera horietatik ε trantsizioen bidez irisgarriak diren egoerak agertzen dira, hau da, $\text{Itxidura-}\varepsilon^{\#}$. Konputazioa egoera-multzo hutsa duen konfigurazio ez-determinista batera iritsi delako bukatu da. ε-AFEDak ezezko erantzuna emango duela ondoriozta dezakegu azkeneko konfigurazioan egora-multzoa hutsa delako eta Y_5-eko elementurik ez dagoelako multzo hutsean.

Konputazio hori konfigurazio deterministez eratutako honako zuhaitz honen bidez ere adieraz daiteke:



Adar horietan ezin da aurrera egin λ_5 kontuan hartuz. Gainera, adar horietan ez da lortu hitza bukatzea. Eta horregatik ε-AFEDak ezezko erantzuna emango du.

3.4.1.4.4 aa katea edo aca katea duten hitzez osatutako lengoaiari dagokion ε-AFEDa



3.4.7 irudia. aa katea edo aca katea duten hitzez osatutako lengoaiari dagokion ε-AFEDa.

Adibide honetako ε-AFEDak (begiratu 3.4.7 irudia) a, b eta c sinboloak izan ditzaketen hitzak hartuko ditu sarrerako datu bezala. ε-AFEDak, datu gisa jaso duen hitzak aa edo aca katea ba al duen erabakiko du. Baiezko kasuan "Bai" eta ezezko kasuan "Ez" erantzunez. Lengoaia horren definizio formala honako hau da:

$$\{w \mid w \in A^* \wedge \exists u, v(u \in A^* \wedge v \in A^* \wedge (w = uaav \vee w = uacav))\}$$

ε-AFED honek lau egoera ditu. Hasierako egoera q_0 da. Karaktere-kate bat irakurtzen hastean q_0 egoeran egongo gara. Lehenengo karakterea a baldin bada, ε-AFED hau q_0 eta q_1 egoeretara igaroko da, hau da, $\lambda(q_0, a) = \{q_0, q_1\}$. Lehenengo karakterea b edo c baldin bada, q_0 egoeran geldituko da, hau da, $\lambda(q_0, b) = \{q_0\}$ eta $\lambda(q_0, c) = \{q_0\}$. ε-AFED hau q_1 egoeran baldin badago, c-rekin q_2-ra igaroko da, hau da, $\lambda(q_1, c) = \{q_2\}$. Bestalde, a edo b-rekin ezingo du inora joan baina q_2 egoerara igaro daiteke sinbolorik irakurri gabe. Beraz, $\lambda(q_1, a) = \lambda(q_1, b) = \varnothing$ eta $\lambda(q_1, \varepsilon) = \{q_2\}$. ε-AFED hau q_2 egoeran baldin badago, a-rekin q_3 egoerara igaroko da, hau da, $\lambda(q_2, a) = \{q_3\}$. Aldiz, b edo c-rekin inora ezingo du joan. Beraz, $\lambda(q_2, b) = \lambda(q_2, c) = \varnothing$. Azkenik, ε-AFED hau q_3 egoeran baldin badago, bai a-rekin, bai b-rekin eta bai c-rekin, q_3 egoeran geldituko da, hau da, $\lambda(q_3, a) = \lambda(q_3, b) = \lambda(q_3, c) = \{q_3\}$. Sarrerako datutzat w hitza emanda, q_0-tik abiatuz eta hitz hori irakurriz lortzen den azken konfigurazioa (S, ε) baldin bada, ε-AFED honek “Bai” erantzungo du $q_3 \in S$ betetzen bada. Hasierako w hitza ε baldin bada, ε-AFED honek q_0 egoeran bukatuko du, hau da, $\lambda^*(\{q_0\}, \varepsilon) = \{q_0\}$. Hor $\{q_0\} \cap Y = \varnothing$ betetzen denez, hau da, $\{q_0\} \cap \{q_3\} = \varnothing$ betetzen denez, ε-AFED honek “Ez” erantzungo du. ε-AFED honek hasierako datutzat emandako w hitza irakurri ezinda geratzen bada, inora ezingo du joan. Hori $\lambda^*(\{q_0\}, w) = \varnothing$ betetzen dela esanez adierazten da. $\varnothing \cap Y = \varnothing$ denez, hau da, $\varnothing \cap \{q_3\} = \varnothing$ denez, ε-AFEDak “Ez” erantzungo du. Hasierako hitz osoa irakurtzea lortu arren, q_3 egoerara ez iritsiz gero “Ez” erantzungo da.

ε-AFED honen osagaiak honako boskote honetan agertzen dira:

$$(Q, A, \lambda, q_0, Y)$$

Hor:

- $Q = \{q_0, q_1, q_2, q_3\}$
- $A = \{a, b, c\}$
- $\lambda : Q \times (A \cup \{\varepsilon\}) \to 2^Q$ jarraian erakusten den taularen bidez definitutako trantsizio-funtzioa da:

λ	a	b	c	ε
q_0	$\{q_0, q_1\}$	$\{q_0\}$	$\{q_0\}$	$\varnothing$
q_1	$\varnothing$	$\varnothing$	$\{q_2\}$	$\{q_2\}$
q_2	$\{q_3\}$	$\varnothing$	$\varnothing$	$\varnothing$
q_3	$\{q_3\}$	$\{q_3\}$	$\{q_3\}$	$\varnothing$

  Taula horren bidez $\lambda(q_0, a) = \{q_0, q_1\}$ dela, $\lambda(q_0, b) = \{q_0\}$ dela, eta abar adierazten da.

- q_0 hasierako egoera da. Hasierako egoera beti bakarra izan ohi da.
- $Y = \{q_3\}$.

Jarraian $(\{q_0\}, aaa)$ konfigurazio ez-deterministari dagokion konputazioa garatuko da urratsez urrats:

1. urratsa: $\lambda^*(\{q_0\}, aaa) = \lambda^*(\lambda(q_0, a), aa) = \lambda^*(\{q_0, q_1\}, aa)$. Kasu honetan Itxidura-$\varepsilon^*(\{q_0\}) = \{q_0\}$ dela kontuan hartu da.

2. urratsa: $\lambda^*(\{q_0, q_1\}, aa) = \lambda^*(\lambda(q_0, a) \cup \lambda(q_1, a) \cup \lambda(q_2, a), a) = \lambda^*(\{q_0, q_1, q_3\}, a)$. Hor Itxidura-$\varepsilon^*(\{q_0, q_1\}) = \{q_0, q_1, q_2\}$ dela kontuan izan da.

3. urratsa: $\lambda^*(\{q_0,q_1,q_3\},a) = \lambda^*(\lambda(q_0,a)\cup\lambda(q_1,a)\cup\lambda(q_2,a)\cup\lambda(q_3,a),\varepsilon) = \lambda^*(\{q_0,q_1,q_3\},\varepsilon)$. Kasu honetan $\text{Itxidura-}\varepsilon^*(\{q_0,q_1,q_3\}) = \{q_0,q_1,q_2,q_3\}$ daukagu.

4. urratsa: $\lambda^*(\{q_0,q_1,q_3\},\varepsilon) = \{q_0,q_1,q_2,q_3\}$. Izan ere, $\text{Itxidura-}\varepsilon^*(\{q_0,q_1,q_3\}) = \{q_0,q_1,q_2,q_3\}$.

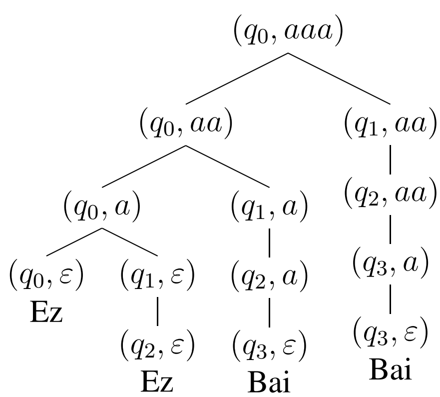
Konputazioa $\{q_0,q_1,q_2,q_3\}$ egoera-multzoan bukatzen denez, eta multzo horretan gutxienez egoera bat Y-koa denez, hau da, $\{q_0,q_1,q_2,q_3\}\cap Y \neq \varnothing$ betetzen denez, aaa hitza lengoaiakoa da (aa katea edo aca katea du).

Konputazio hori konfigurazio ez-deterministez osatutako honako sekuentzia honen bidez adieraz daiteke:

$$\begin{array}{c}
(\{q_0\},aaa) \equiv_{\text{Itxidura-}\varepsilon^*} (\{q_0\},aaa)\\
|\\
(\{q_0,q_1\},aa) \equiv_{\text{Itxidura-}\varepsilon^*} (\{q_0,q_1,q_2\},aa)\\
|\\
(\{q_0,q_1,q_3\},a) \equiv_{\text{Itxidura-}\varepsilon^*} (\{q_0,q_1,q_2,q_3\},a)\\
|\\
(\{q_0,q_1,q_3\},\varepsilon) \equiv_{\text{Itxidura-}\varepsilon^*} (\{q_0,q_1,q_2,q_3\},\varepsilon)\\
\text{Bai}
\end{array}$$

Lerro bakoitzean, $\equiv$ sinboloaren ezkerrean, une horretako egoera-multzoa zein den zehazten da eta $\equiv$ sinboloaren eskuinean, egoera horietatik ε trantsizioen bidez irisgarriak diren egoerak agertzen dira, hau da, $\text{Itxidura-}\varepsilon^{\#}$. Konputazioa ε hitza duen konfigurazio ez-determinista batera iritsi delako bukatu da.

Konputazio hori konfigurazio deterministez eratutako honako zuhaitz honen bidez ere adieraz daiteke:



Adar horietatik bat bera ere ez da bukatu konfigurazio bat adarrean bertan errepikatu delako. Adar bakoitza λ kontuan hartuz aurrera egiterik ez dagoelako bukatu da. Ezkerretik hasita, lehenengo adarrak bigarren aa azpikatea atzematen du, hau da, $a\underline{aa}$. Laugarren adarrak lehenengo aa azpikatea atzematen du, hau da, $\underline{aa}a$. Bi adar horietan (q_3,ε) konfigurazioa dugunez, ε-AFEDak "Bai" erantzungo duela ondoriozta dezakegu.

Jarraian $(\{q_0\}, abca)$ konfigurazio ez-deterministari dagokion konputazioa garatuko da urratsez urrats:

1. urratsa: $\lambda^*(\{q_0\}, abca) = \lambda^*(\lambda(q_0, a), bca) = \lambda^*(\{q_0, q_1\}, bca)$. Hor $\textsf{Itxidura-}\varepsilon^*(\{q_0\}) = \{q_0\}$ dela kontuan izan da.

2. urratsa: $\lambda^*(\{q_0, q_1\}, bca) = \lambda^*(\lambda(q_0, b) \cup \lambda(q_1, b) \cup \lambda(q_2, b), ca) = \lambda^*(\{q_0\}, ca)$. Kasu honetan $\textsf{Itxidura-}\varepsilon^*(\{q_0, q_1\}) = \{q_0, q_1, q_2\}$ betetzen da.

3. urratsa: $\lambda^*(\{q_0\}, ca) = \lambda^*(\lambda(q_0, c), a) = \lambda^*(\{q_0\}, a)$. Kasu honetan $\textsf{Itxidura-}\varepsilon^*(\{q_0\}) = \{q_0\}$ daukagu.

4. urratsa: $\lambda^*(\{q_0\}, a) = \lambda^*(\lambda(q_0, a), \varepsilon) = \lambda^*(\{q_0, q_1\}, \varepsilon)$. Urrats honetan $\textsf{Itxidura-}\varepsilon^*(\{q_0\}) = \{q_0\}$ dela kontuan izan behar da.

5. urratsa: $\lambda^*(\{q_0, q_1\}, \varepsilon) = \{q_0, q_1, q_2\}$. Izan ere, $\textsf{Itxidura-}\varepsilon^*(\{q_0, q_1\}) = \{q_0, q_1, q_2\}$ baita.

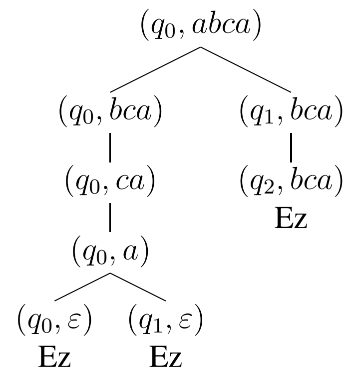
Bukaeran lortutako egoera-multzoan Y-koa den egoerarik ez dagoenez, $abca$ hitza ez da ε-AFED honi dagokion lengoaiakoa.

Konputazio hori deterministak ez diren konfigurazioz osatutako honako sekuentziaren bidez adieraz daiteke:

$$\begin{array}{c}
(\{q_0\}, abca) \equiv_{\textsf{Itxidura-}\varepsilon^*} (\{q_0\}, abca) \\
| \\
(\{q_0, q_1\}, bca) \equiv_{\textsf{Itxidura-}\varepsilon^*} (\{q_0, q_1, q_2\}, bca) \\
| \\
(\{q_0\}, ca) \equiv_{\textsf{Itxidura-}\varepsilon^*} (\{q_0\}, ca) \\
| \\
(\{q_0\}, a) \equiv_{\textsf{Itxidura-}\varepsilon^*} (\{q_0\}, a) \\
| \\
(\{q_0, q_1\}, \varepsilon) \equiv_{\textsf{Itxidura-}\varepsilon^*} (\{q_0, q_1, q_2\}, \varepsilon) \\
\text{Ez}
\end{array}$$

Lerro bakoitzean, $\equiv$ sinboloaren ezkerrean, une horretako egoera-multzoa zein den zehazten da eta $\equiv$ sinboloaren eskuinean, egoera horietatik ε trantsizioen bidez irisgarriak diren egoerak agertzen dira, hau da, $\textsf{Itxidura-}\varepsilon^{\#}$. Konputazioa ε hitza duen konfigurazio ez-determinista batera iritsi delako bukatu da.

Konputazio hori trantsizio deterministez eratutako honako zuhaitz honen bidez ere adieraz daiteke:



Adar horietako bat bera ere ez da bukatu azkeneko konfigurazioa adarrean bertan errepikatuta dagoelako. Adar denak λ kontuan hartuz aurrera egiterik ez dagoelako bukatu dira. (q_3, ε) konfigurazioa ez denez agertzen, ε-AFEDak “Ez” erantzungo duela ondoriozta dezakegu.

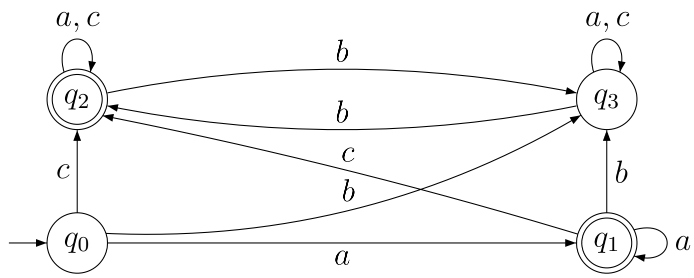
3.4.1.4.5 Adibidea: Bakarrik a sinboloaren errepikapenez osatuta dauden edo b-ren agerpen-kopuru bikoitia duten hitz ez-hutsez eratutako lengoaiari dagokion ε-AFEDa

Adibide honetako ε-AFEDak a, b eta c sinboloak izan ditzaketen hitzak hartuko ditu sarrerako datu gisa. ε-AFEDak, datu gisa jaso duen hitza bakarrik a sinboloaren errepikapenez osatuta hitz ez-hutsa al den edo b-ren agerpen-kopuru bikoitia duen hitz ez-hutsa al den erabakiko du. Baiezko kasuan “Bai” eta ezezko kasuan “Ez” erantzunez. Lengoaia horren definizio formala honako hau da:

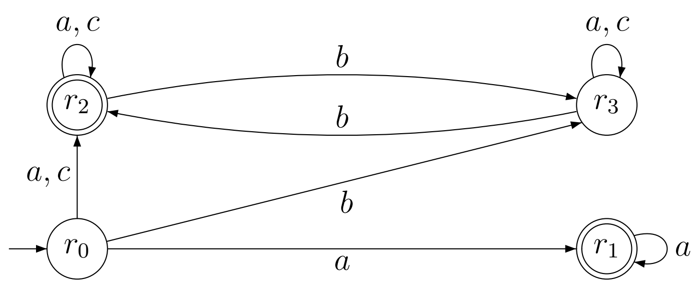
$$\{w \mid w \in A^* \wedge |w| \geq 1 \wedge (|w|_a = |w| \vee |w|_b \bmod 2 = 0)\}$$

Baina ε-AFED hori aurkeztu aurretik, lengoaia bera definitzen duten AFD bat eta AFED bat erakutsiko dira. Horrela hiru formalismo horiek alderatu ahal izango ditugu eta gehienetan ε-AFEDak AFEDak eta AFD-ak baino ulerterrazagoak direla ikusiko dugu. Era berean, AFEDak AFDak baino ulerterrazagoak direla ere erakutsiko da. Izan ere, adibide honetan kontrolatu beharreko baldintzak bakoitza bere aldetik aztertu ahal izango dira AFEDa erabiliz, diseinua erraztuz.

Hasteko, 3.4.8 irudian adibide honetako lengoaiari dagokion AFD bat erakusten da. q_1 egoerak orain arte irakurri diren sinbolo denak a izan direla eta gainera gutxienez bat irakurri dela eta b-rik ez dela irakurri (eta ondorioz b-ren agerpen-kopurua bikoitia dela) gogoratzen du. Bestalde, q_2 egoeran gaudenean badakigu gutxienez a ez den sinboloren bat irakurri dela eta b kopurua bikoitia dela. Bukatzeko, q_3 egoeran gaudenean badakigu gutxienez a ez den sinboloren bat irakurri dela eta b kopurua bakoitia dela. Hor q_1-etik q_2-ra eta q_1-etik q_3-ra doazen trantsizioak direla eta, aztertu beharreko bi baldintzen egiaztapena (bakarrik a agertzea edo b kopurua bikoitia izatea) nahastuta dago. Bakarrik a sinboloaren errepikapenez osatuta dauden hitz ez hutsak atzemateko q_0 eta q_1 egoerak erabiltzen dira eta b kopuru bikoitia duten hitz hutsak atzemateko lau egoerak behar dira (q_0, q_1, q_2 eta q_3).



3.4.8 irudia. Bakarrik a sinboloaren errepikapenez osatuta dauden edo b-ren agerpen-kopuru bikoitia duten hitz ez-hutsez eratutako lengoaiari dagokion AFDa.



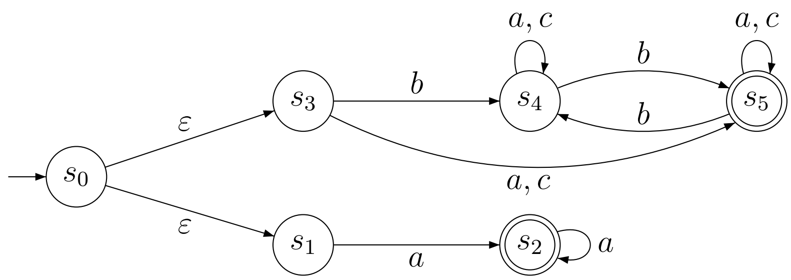
3.4.9 irudia. Bakarrik a sinboloaren errepikapenez osatuta dauden edo b-ren agerpen-kopuru bikoitia duten hitz ez-hutsez eratutako lengoaiari dagokion AFEDa.

Lengoaia bera definitzen duen AFED bat 3.4.9 irudian ikus dezakegu. r_1, r_2 eta r_3 egoerek 3.4.8 irudiko AFDko q_1, q_2 eta q_3 egoeren helburu bera dute. Baina AFED horretan bakarrik a-ren errepikapenez osatutako hitza izatea (r_0 eta r_1 egoeren bidez) eta b kopurua bikoitia izatea (r_0, r_2 eta r_3 egoeren bidez) bakoitza bere aldetik aztertu daiteke. Beraz, AFED horren bidez disjuntzioaren esanahia 3.4.8 irudiko AFDan baino era garbiagoan adierazten da.

3.4.10 irudian, adibide honetako lengoaia definitzen duen ε-AFED bat daukagu. Hor s_0 egoeratik ateratzen diren ε trantsizioei esker, baldintza biak (hitz hutsa ez izanda, bakarrik a-ren errepikapenez osatuta egotea edo b kopurua bikoitia izatea) 3.4.9 irudiko AFEDan baino hobeto bereizten dira. s_0-tik abiatzerakoan, zuzenean s_1 eta s_3 egoeretara igarotzen da ezer irakurri gabe. Alde batetik, s_1 eta s_2 egoeren bidez hitza hutsa ez izatea eta a-ren errepikapenez bakarrik osatuta egotea kontrolatzen da. Bestetik, s_3, s_4 eta s_5 egoeren bidez hitza hutsa ez izatea eta b kopurua bikoitia izatea kontrolatzen da. Datu gisa emandako hitzak bi baldintza horietakoren bat betetzen badu, erantzuna “Bai” izango da eta bi baldintza horietatik bat bera ere ez badu betetzen, erantzuna “Ez” izango da.

Adibide honetako ε-AFEDa honako boskote honen bidez defini daiteke:

$$(S, A, \lambda, s_0, Y)$$



3.4.10 irudia. Bakarrik a sinboloaren errepikapenez osatuta dauden edo b-ren agerpen-kopuru bikoitia duten hitz ez-hutsez eratutako lengoaiari dagokion ε-AFEDa.

Hor:

- $S = \{s_0, s_1, s_2, s_3, s_4, s_5\}$
- $A = \{a, b, c\}$
- $\lambda : S \times (A \cup \{\varepsilon\}) \to 2^S$ honako taula honen bidez definitutako trantsizio-funtzioa da:

λ	a	b	c	ε
s_0	$\varnothing$	$\varnothing$	$\varnothing$	$\{s_1, s_3\}$
s_1	$\{s_2\}$	$\varnothing$	$\varnothing$	$\varnothing$
s_2	$\{s_2\}$	$\varnothing$	$\varnothing$	$\varnothing$
s_3	$\{s_5\}$	$\{s_4\}$	$\{s_5\}$	$\varnothing$
s_4	$\{s_4\}$	$\{s_5\}$	$\{s_4\}$	$\varnothing$
s_5	$\{s_5\}$	$\{s_4\}$	$\{s_5\}$	$\varnothing$

  Taula horretan $\lambda(s_0, a) = \varnothing$ dela, $\lambda(q_0, \varepsilon) = \{s_1, s_3\}$ dela, eta abar adierazten da.

- s_0 hasierako egoera da. Hasierako egoera bakarra izan ohi da beti.
- $Y = \{s_2, s_5\}$.

Orain $(\{q_0\}, aaa)$ konfigurazio ez-deterministari dagokion konputazioa garatuko da urratsez urrats:

1. urratsa: $\lambda^*(\{s_0\}, aaa) = \lambda^*(\lambda(s_0, a) \cup \lambda(s_1, a) \cup \lambda(s_3, a), aa) = \lambda^*(\{s_2, s_5\}, aa)$.
Hor, Itxidura-$\varepsilon^*(\{s_0\}, \lambda) = \{s_0, s_1, s_3\}$ dela kontuan hartu da.

2. urratsa: $\lambda^*(\{s_2, s_5\}, aa) = \lambda^*(\lambda(s_2, a) \cup \lambda(s_5, a) = \lambda^*(\{s_2, s_5\}, a)$.
Kasu honetan, Itxidura-$\varepsilon^*(\{s_2, s_5\}, \lambda) = \{s_2, s_5\}$ dela kontuan izan behar da.

3. urratsa: $\lambda^*(\{s_2, s_5\}, a) = \lambda^*(\{s_2, s_5\}, a\varepsilon) = \lambda^*(\lambda(s_2, a) \cup \lambda(s_5, a), \varepsilon) = \lambda^*(\{s_2, s_5\}, \varepsilon)$.
Kasu honetan ere Itxidura-$\varepsilon^*(\{s_2, s_5\}, \lambda) = \{s_2, s_5\}$ dela kontuan izan behar da.

4. urratsa: $\lambda^*(\{s_2, s_5\}, \varepsilon) = \{s_2, s_5\}$. Izan ere, Itxidura-$\varepsilon^*(\{s_2, s_5\}, \lambda) = \{s_2, s_5\}$ da.

Konputazioa $\{s_2, s_5\}$ egoera-multzoan bukatu denez eta $\{s_2, s_5\}$ multzoa eta Y multzoaren arteko ebakidura hutsa ez denez, badakigu hitza lengoaiakoa dela.

Konputazio hori era grafikoagoan ere adieraz daiteke konfigurazio ez-deterministez osatutako honako sekuentzia honen bidez:

$$\begin{array}{c}
(\{s_0\}, aaa) \equiv_{\mathsf{Itxidura}\text{-}\varepsilon^*} (\{s_0, s_1, s_3\}, aaa) \\
| \\
(\{s_2, s_5\}, aa) \equiv_{\mathsf{Itxidura}\text{-}\varepsilon^*} (\{s_2, s_5\}, aa) \\
| \\
(\{s_2, s_5\}, a) \equiv_{\mathsf{Itxidura}\text{-}\varepsilon^*} (\{s_2, s_5\}, a) \\
| \\
(\{s_2, s_5\}, \varepsilon) \equiv_{\mathsf{Itxidura}\text{-}\varepsilon^*} (\{s_2, s_5\}, \varepsilon) \\
\text{Bai}
\end{array}$$

Lerro bakoitzean, $\equiv$ sinboloaren ezkerrean, une horretako egoera-multzoa zein den zehazten da eta $\equiv$ sinboloaren eskuinean, egoera horietatik ε trantsizioen bidez irisgarriak diren egoerak agertzen dira, hau da, Itxidura-$\varepsilon^{\#}$. Konputazioa ε hitza duen konfigurazio ez-determinista batera iritsi delako bukatu da.

Konputazio hori konfigurazio deterministez eratutako honako zuhaitz honen bidez ere adieraz daiteke:

$$\begin{array}{ccc}
 & (s_0, aaa) & \\
\swarrow & & \searrow \\
(s_1, aaa) & & (s_3, aaa) \\
| & & | \\
(s_2, aa) & & (s_5, aa) \\
| & & | \\
(s_2, a) & & (s_5, a) \\
| & & | \\
(s_2, \varepsilon) & & (s_5, \varepsilon) \\
\text{Bai} & & \text{Bai}
\end{array}$$

Zuhaitzak bi adar ditu. Lehenengo adarrean (ezkerretik hasita) (s_2, ε) konfigurazioa daukagu. Konfigurazio horretan egoera Y-koa da eta hitza hutsa da. Beraz, ε-AFEDak baiezkoa erantzungo duela ondoriozta dezakegu. Lehenengo adar horren bidez hitza bakarrik a sinboloaren errepikapenez osatutako hitz ez-hutsa dela detektatu da. Bigarren adarrean (s_5, ε) konfigurazioa daukagu. Konfigurazio horretan egoera Y-koa da eta hitza hutsa da. Beraz, bigarren adar hori kontuan hartuz ere ε-AFEDak baiezkoa erantzungo duela ondoriozta dezakegu. Bigarren adar horren bidez hitza b kopurua bikoitia duen hitz ez-hutsa dela detektatu da.

Orain $(\{q_0\}, abca)$ konfigurazio ez-deterministari dagokion konputazioa garatuko da urratsez urrats:

1. urratsa: $\lambda^*(\{s_0\}, abca) = \lambda^*(\lambda(s_0, a) \cup \lambda(s_1, a) \cup \lambda(s_3, a), bca) = \lambda^*(\{s_2, s_5\}, bca)$.
Hor, Itxidura-$\varepsilon^*(\{s_0\}, \lambda) = \{s_0, s_1, s_3\}$ dela kontuan hartu da.

2. urratsa: $\lambda^*(\{s_2, s_5\}, bca) = \lambda^*(\lambda(s_2, b) \cup \lambda(s_5, b), ca) = \lambda^*(\{s_4\}, ca)$. Kasu honetan Itxidura-$\varepsilon^*(\{s_2, s_5\}, \lambda) = \{s_2, s_5\}$ dela kontuan hartu behar da.

3. urratsa: $\lambda^*(\{s_4\}, ca) = \lambda^*(\lambda(s_4, c), a) = \lambda^*(\{s_4\}, a)$. Kasu honetan Itxidura-$\varepsilon^*(\{s_4\}, \lambda) = \{s_4\}$ da.

4. urratsa: $\lambda^*(\{s_4\}, a) = \lambda^*(\{s_4\}, a\varepsilon) = \lambda^*(\lambda(s_4, a), \varepsilon) = \lambda^*(\{s_4\}, \varepsilon)$. Izan ere, Itxidura-$\varepsilon^*(\{s_4\}, \lambda) = \{s_4\}$.

5. urratsa: $\lambda^*(\{s_4\}, \varepsilon) = \{s_4\}$. Urrats honetan ere Itxidura-$\varepsilon^*(\{s_4\}, \lambda) = \{s_4\}$ dela kontuan hartu behar da.

Konputazioa $\{s_4\}$ egoera-multzoan bukatu denez eta $\{s_4\}$ multzoa eta Y multzoaren arteko ebakidura hutsa denez, badakigu hitza ez dela ε-AFEDak definitutako lengoaiakoa.

Konputazio hori era grafikoagoan ere adieraz daiteke konfigurazio ez-deterministez osatutako honako sekuentzia honen bidez:

$$
\begin{array}{c}
(\{s_0\}, abca) \equiv_{\text{Itxidura-}\varepsilon^*} (\{s_0, s_1, s_3\}, abca) \\
| \\
(\{s_2, s_5\}, bca) \equiv_{\text{Itxidura-}\varepsilon^*} (\{s_2, s_5\}, bca) \\
| \\
(\{s_4\}, ca) \equiv_{\text{Itxidura-}\varepsilon^*} (\{s_4\}, ca) \\
| \\
(\{s_4\}, a) \equiv_{\text{Itxidura-}\varepsilon^*} (\{s_4\}, a) \\
| \\
(\{s_4\}, \varepsilon) \equiv_{\text{Itxidura-}\varepsilon^*} (\{s_4\}, \varepsilon) \\
\text{Ez}
\end{array}
$$

Lerro bakoitzean, $\equiv$ sinboloaren ezkerrean, une horretako egoera-multzoa zein den zehazten da eta $\equiv$ sinboloaren eskuinean, egoera horietatik ε trantsizioen bidez irisgarriak diren egoerak agertzen dira, hau da, Itxidura-$\varepsilon^\#$. Konputazioa ε hitza duen konfigurazio ez-determinista batera iritsi delako bukatu da. ε-AFEDak ezezko erantzuna emango du azkeneko egoera-multzoan ez dagoelako Y-ko egoerarik.

Konputazio hori trantsizio deterministez eratutako honako zuhaitz honen bidez ere adieraz daiteke:

- $(s_0, abca)$
  - $(s_1, abca)$
    - (s_2, bca) Ez
  - $(s_3, abca)$
    - (s_5, bca)
      - (s_4, ca)
        - (s_4, a)
          - (s_4, ε) Ez

Lehenengo adarrean ez da lortu hitz osoa irakurtzea. Bigarrengo adarrean hitz hutsa duen konfigurazio bat lortu da baina konfigurazio horretako egoera ez da Y-koa. Beraz, ε-AFEDak $abca$ hitzarentzat ezezkoa erantzungo duela ondoriozta dezakegu.

3.4.1.4.6 Adibidea: bakarrik a-ren errepikapenak edo b kopuru bakoitia duten hitz ez-hutsez osatutako legoaiari dagokion ε-AFEDa

Adibide honetako ε-AFEDak a, b eta c sinboloak izan ditzaketen hitzak hartuko ditu sarrerako datu gisa. ε-AFEDak, datu gisa jaso duen hitza bakarrik a sinboloaren errepikapenez osatuta dagoen hitz ez-hutsa al den edo b-ren agerpen-kopuru bakoitia duen hitz ez-hutsa al den erabakiko du. Baiezko kasuan “Bai” eta ezezko kasuan “Ez” erantzunez. Lengoaia horren definizio formala honako hau da:

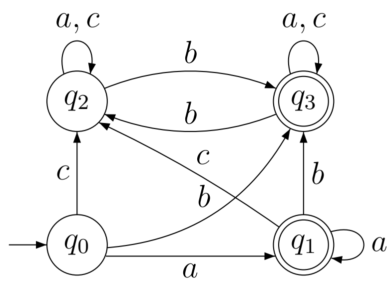
$$\{w \mid w \in A^* \wedge |w| \geq 1 \wedge (|w|_a = |w| \vee |w|_b \bmod 2 \neq 0)\}$$

Baina ε-AFED hori aurkeztu aurretik, lengoaia bera definitzen duten AFD bat eta AFED bat erakutsiko dira. Horrela hiru formalismo horiek alderatu ahal izango ditugu eta gehienetan ε-AFEDak AFEDak eta AFDak baino ulerterrazagoak direla ikusiko dugu. Era berean, AFEDak AFDak baino ulerterrazagoak direla ere erakutsiko da. Izan ere, adibide honetan kontrolatu beharreko baldintzak bakoitza bere aldetik aztertu ahal izango dira AFEDa erabiliz, diseinua erraztuz.

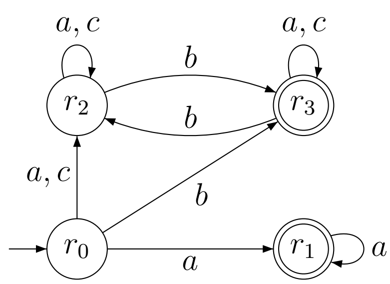
Hasteko, 3.4.11 irudian adibide honetako lengoaia definitzen duen AFD bat daukagu. Aurreko ataleko adibidean gertatzen den bezala, hitzak bakarrik a-ren errepikapenak al dituen detektatzen duen q_1 egoera eta b kopurua bakoitia izatearen propietatea kontrolatzen duten q_2 eta q_3 egoerak konektatuta daude.

3.4.12 irudian lengoaia bera definitzen duen AFED bat daukagu. AFED horretan ez dago r_1-etik r_2-ra eta r_1-etik r_3-ra doan trantsiziorik. Beraz, bi propietateen azterketa bereiztea lortu da.

3.4.13 irudian, lengoaia bera definitzen duen ε-AFED bat daukagu. s_0 egoeratik ateratzen diren ε trantsizioak aztertu beharreko baldintza biak (bakarrik a-ren errepikapenak edukitzea



3.4.11 irudia. Bakarrik a-ren errepikapenak edo b kopuru bakoitia duten hitz ez-hutsez osatutako legoaiari dagokion AFDa.



3.4.12 irudia. Bakarrik a-ren errepikapenak edo b kopuru bakoitia duten hitz ez-hutsez osatutako legoaiari dagokion AFEDa.

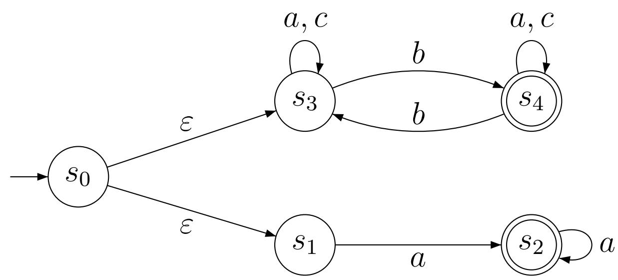
edo b kopuru bakoitia edukitzea) guztiz bereiztea ahalbidetzen dute. s_0 egoeratik abiatzen bagara, zuzenean s_1 eta s_3 egoeretara igaroko gara. Alde batetik, s_1 eta s_2 egoeren bidez, hitzak b-rik eta c-rik ez edukitzea baina gutxienez a bat edukitzea kontrolatzen da. Bestalde, s_3 eta s_4 egoeren bidez hitzak b kopuru bakoitia izatearen propietatea kontrolatzen da. Hitzak propietate horietakoren bat betetzen badu, erantzuna “Bai” izango da baina hitzak bi propietate horietatik bat bera ere ez badu betetzen, orduan erantzuna “Ez” izango da. Adibide honetako automatak aurreko adibideko automaten antzekoak dira baina xehetasun garratzitsu batzuk desberdinak dira, esate baterako, zirkulu bikoitza duten egoerak.

Jarraian 3.4.13 irudiko ε-AFEDaren osagaien definizio formala emango da. Adibide honetako ε-AFEDa honako boskote hau da:

$$(S, A, \lambda, s_0, Y)$$

Hor

- $S = \{s_0, s_1, s_2, s_3, s_4\}$
- $A = \{a, b, c\}$



3.4.13 irudia. Bakarrik a-ren errepikapenak edo b kopuru bakoitia duten hitz ez-hutsez osatutako legoaiari dagokion ε-AFEDa.

- $\lambda : S \times (A \cup \{\varepsilon\}) \to 2^S$ honako taula honen bidez definitutako trantsizio-funtzioa da:

λ	a	b	c	ε
s_0	$\varnothing$	$\varnothing$	$\varnothing$	$\{s_1, s_3\}$
s_1	$\{s_2\}$	$\varnothing$	$\varnothing$	$\varnothing$
s_2	$\{s_2\}$	$\varnothing$	$\varnothing$	$\varnothing$
s_3	$\{s_3\}$	$\{s_4\}$	$\{s_3\}$	$\varnothing$
s_4	$\{s_4\}$	$\{s_3\}$	$\{s_4\}$	$\varnothing$

Taula horren bidez $\lambda(s_0, a) = \varnothing$ dela, $\lambda(s_0, \varepsilon) = \{s_1, s_3\}$ dela eta abar adierazten da.

- q_0 hasierako egoera da. Hasierako egoera beti bakarra izango da.
- $Y = \{s_2, s_4\}$.

Orain $(\{q_0\}, aca)$ konfigurazio ez-deterministari dagokion konputazioa garatuko da urratsez urrats:

1. urratsa: $\lambda^*(\{s_0\}, aca) = \lambda^*(\lambda(s_0, a) \cup \lambda(s_1, a) \cup \lambda(s_3, a), ca) = \lambda^*(\{s_2, s_3\}, ca)$. Hor, Itxidura-$\varepsilon^*(\{s_0\}, \lambda) = \{s_0, s_1, s_3\}$ dela kontuan hartu da.

2. urratsa: $\lambda^*(\{s_2, s_3\}, ca) = \lambda^*(\lambda(s_2, c) \cup \lambda(s_3, c), a) = \lambda^*(\{s_3\}, a)$. Kasu honetan Itxidura-$\varepsilon^*(\{s_2, s_3\}, \lambda) = \{s_2, s_3\}$ dela kontuan hartu behar da.

3. urratsa: $\lambda^*(\{s_3\}, a) = \lambda^*(\{s_3\}, a\varepsilon) = \lambda^*(\lambda(s_3, a), \varepsilon) = \lambda^*(\{s_3\}, \varepsilon)$. Kalkulu hori egitean Itxidura-$\varepsilon^*(\{s_3\}, \lambda) = \{s_3\}$ dela kontuan izan behar da.

4. urratsa: $\lambda^*(\{s_3\}, \varepsilon) = \{s_3\}$. Izan ere, Itxidura-$\varepsilon^*(\{s_3\}, \lambda) = \{s_3\}$ da.

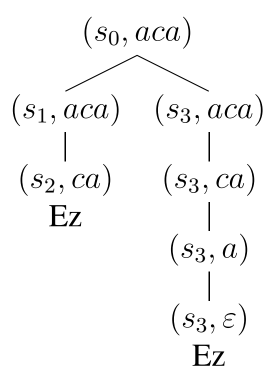
Konputazioa $\{s_3\}$ egoera-multzoan bukatu denez eta $\{s_3\}$ multzoa eta Y multzoaren arteko ebakidura hutsa denez, badakigu hitza ez dela ε-AFEDak definitutako lengoaiakoa.

Konputazio hori era grafikoagoan ere adieraz daiteke konfigurazio ez-deterministez osatutako honako sekuentzia honen bidez:

$$
\begin{array}{c}
(\{s_0\}, aca) \equiv_{\mathsf{Itxidura}\text{-}\varepsilon^*} (\{s_0, s_1, s_3\}, aca) \\
| \\
(\{s_2, s_3\}, ca) \equiv_{\mathsf{Itxidura}\text{-}\varepsilon^*} (\{s_2, s_3\}, ca) \\
| \\
(\{s_3\}, a) \equiv_{\mathsf{Itxidura}\text{-}\varepsilon^*} (\{s_3\}, a) \\
| \\
(\{s_3\}, \varepsilon) \equiv_{\mathsf{Itxidura}\text{-}\varepsilon^*} (\{s_3\}, \varepsilon) \\
\text{Ez}
\end{array}
$$

Lerro bakoitzean, $\equiv$ sinboloaren ezkerrean, une horretako egoera-multzoa zein den zehazten da eta $\equiv$ sinboloaren eskuinean, egoera horietatik ε trantsizioen bidez irisgarriak diren egoerak agertzen dira, hau da, Itxidura-$\varepsilon^{\#}$. Konputazioa ε hitza duen konfigurazio ez-determinista batera iritsi delako bukatu da.

Konputazio hori konfigurazio deterministez eratutako honako zuhaitz honen bidez ere adieraz daiteke:



Ezkerretik hasita, lehenengo adarrean Y-koa de s_2 egoerara iristea lortu da baina ez da lortu hitz osoa irakurtzea. Bigarren adarrean hitz osoa irakurtzea lortu da baina s_3 ez da onarpen egoera bat. Guztira, ez da lortu hitz hutsaz eta onarpen egoera batez osatutako konfigurazio bat eta, ondorioz, ε-AFEDak ezezko erantzuna emango du aca hitzarentzat.

Orain $(\{q_0\}, abca)$ konfigurazio ez-deterministari dagokion konputazioa garatuko da urratsez urrats:

1. urratsa: $\lambda^*(\{s_0\}, abca) = \lambda^*(\lambda(s_0, a) \cup \lambda(s_1, a) \cup \lambda(s_3, a), bca) = \lambda^*(\{s_2, s_3\}, bca)$.
Hor, Itxidura-$\varepsilon^*(\{s_0\}, \lambda) = \{s_0, s_1, s_3\}$ dela kontuan hartu behar da.

2. urratsa: $\lambda^*(\{s_2, s_3\}, bca) = \lambda^*(\lambda(s_2, b) \cup \lambda(s_3, b), ca) = \lambda^*(\{s_4\}, a)$.
Kasu honetan Itxidura-$\varepsilon^*(\{s_2, s_3\}, \lambda) = \{s_2, s_3\}$ dela kontuan izan da.

3. urratsa: $\lambda^*(\{s_4\}, ca) = \lambda^*(\lambda(s_4, c), a) = \lambda^*(\{s_4\}, a)$.
Kalkulu hori egitean Itxidura-$\varepsilon^*(\{s_4\}, \lambda) = \{s_4\}$ dela kontuan hartu da.

4. urratsa: $\lambda^*(\{s_4\}, a) = \lambda^*(\{s_4\}, a\varepsilon) = \lambda^*(\lambda(s_4, a), \varepsilon) = \lambda^*(\{s_4\}, \varepsilon)$. Izan ere, Itxidura-$\varepsilon^*(\{s_4\}, \lambda) = \{s_4\}$ da.

5. urratsa: $\lambda^*(\{s_4\}, \varepsilon) = \{s_4\}$. Izan ere, Itxidura-$\varepsilon^*(\{s_4\}, \lambda) = \{s_4\}$ da.

Konputazioa $\{s_4\}$ egoera-multzoan bukatu denez eta $\{s_4\}$ multzoa eta Y multzoaren arteko ebakidura hutsa ez denez, badakigu hitza ε-AFEDak definitutako lengoaiakoa dela.

Konputazio hori era grafikoagoan ere adieraz daiteke konfigurazio ez-deterministez osatutako honako sekuentzia honen bidez:

$$\begin{array}{c}
(\{s_0\}, abca) \equiv_{\text{Itxidura-}\varepsilon^*} (\{s_0, s_1, s_3\}, abca) \\
| \\
(\{s_2, s_3\}, bca) \equiv_{\text{Itxidura-}\varepsilon^*} (\{s_2, s_3\}, bca) \\
| \\
(\{s_4\}, ca) \equiv_{\text{Itxidura-}\varepsilon^*} (\{s_4\}, ca) \\
| \\
(\{s_4\}, a) \equiv_{\text{Itxidura-}\varepsilon^*} (\{s_4\}, a) \\
| \\
(\{s_4\}, \varepsilon) \equiv_{\text{Itxidura-}\varepsilon^*} (\{s_4\}, \varepsilon) \\
\text{Bai}
\end{array}$$

Lerro bakoitzean, $\equiv$ sinboloaren ezkerrean, une horretako egoera-multzoa zein den zehazten da eta $\equiv$ sinboloaren eskuinean, egoera horietatik ε trantsizioen bidez irisgarriak diren egoerak agertzen dira, hau da, Itxidura-$\varepsilon^{\#}$. Konputazioa ε hitza duen konfigurazio ez-determinista batera iritsi delako bukatu da.

Konputazio hori konfigurazio deterministez eratutako honako zuhaitz honen bidez ere adieraz daiteke:

$$\begin{array}{cc}
\multicolumn{2}{c}{(s_0, abca)} \\
(s_1, abca) & (s_3, abca) \\
| & | \\
(s_2, bca) & (s_3, bca) \\
\text{Ez} & | \\
 & (s_4, ca) \\
 & | \\
 & (s_4, a) \\
 & | \\
 & (s_4, \varepsilon) \\
 & \text{Bai}
\end{array}$$

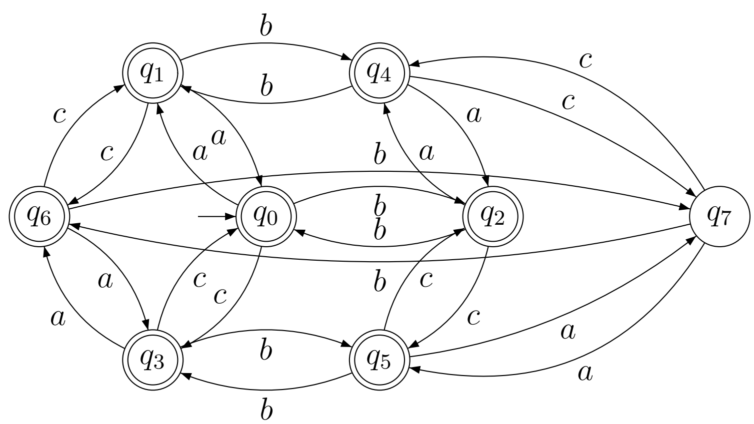
Ezkerretik hasita, lehenengo adarrean Y-koa de s_2 egoerara iristea lortu arren ez da lortu hitz osoa irakurtzea. Bigarren adarrean hitz hutsaz eta Y-koa den egoera batez osatutako konfugurazio bat lortu daosoa irakurtzea lortu da: (s_4, ε). Ondorioz, ε-AFEDak baiezko erantzuna emango du $abca$ hitzarentzat. Kasu honetan bigarren adarrak b kopurua bakoitia dela detektatu du.

3.4.1.4.7 Adibidea: a, b edo c kopuru bikoitian duten hitzez osatutako lengoaiari dagokion ε-AFEDa

Adibide honetako ε-AFEDak a, b eta c sinboloak izan ditzaketen hitzak hartuko ditu sarrerako datutzat. ε-AFEDak, datu gisa jaso duen hitzean a sinboloa, b sinboloa edo c sinboloa kopuru bikoitian agertzen al den erabakiko du. Sinbolo horietako bat kopuru bikoitian agertzearekin nahikoa izango da "Bai" erantzuteko. ε hitz hutsa lengoaiakoa da. Lengoaia horren definizio formala honako hau da:

$$\{w \mid w \in A^* \wedge (|w|_a \bmod 2 = 0 \vee |w|_b \bmod 2 = 0 \vee |w|_c \bmod 2 = 0)\}$$

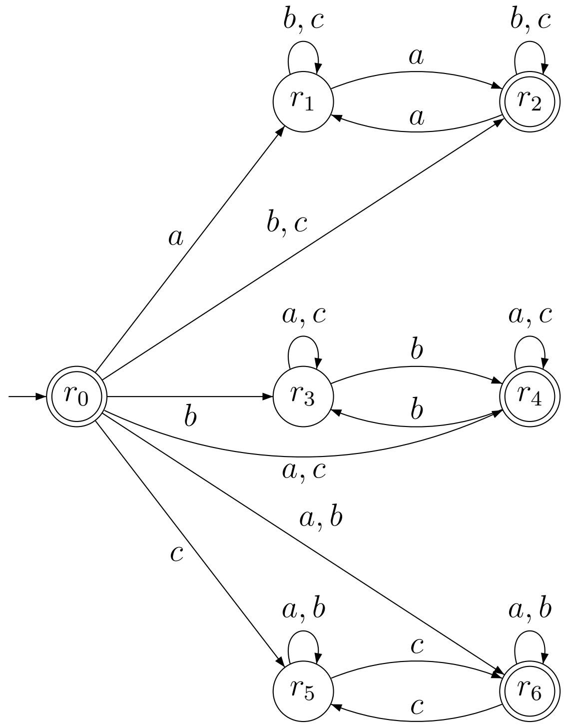
Baina ε-AFED hori aurkeztu aurretik, lengoaia bera definitzen duten AFD bat eta AFED bat erakutsiko dira. Horrela hiru formalismo horiek alderatu ahal izango ditugu eta gehienetan ε-AFEDak AFEDak eta AFDak baino ulerterrazagoak direla ikusiko dugu. Era berean, AFEDak AFDak baino ulerterrazagoak direla ere erakutsiko da. Izan ere, adibide honetan kontrolatu beharreko baldintzak bakoitza bere aldetik aztertu ahal izango dira AFEDa erabiliz, eta horrela diseinua erraztuz.



3.4.14 irudia. a, b edo c kopuru bikoitian duten hitzez osatutako lengoaiari dagokion AFDa.

Adibide honetako lehenengo irudian, hau da, 3.4.14 irudian, adibide honetako lengoaia definitzen duen AFD bat erakusten da. Kontrolatu beharreko hiru propietateak (a kopurua bikoitia izatea edo b kopurua bikoitia izatea edo c kopurua bikoitia izatea) elkarrekin erabat lotuta edo konektatuta agertzen dira. AFADa oso konplikatua ez izan arren, egoera bakoitzak gogoratzen duen propietatea zein den jakitea ez da berehalakoa. Adibide hau nahiko txikia da baina, hala ere, egiaztatu edo kontrolatu beharreko propietateak zailagoak baldin badira, AFDa eraikitzea eta ulertzea nahiko zaila izan daitekeela erakusteko balio digu.

Adibide honetako bigarren irudian, hau da, 3.4.15 irudian, lengoaia bera definitzeko balio duen AFED bat daukagu. AFED horren bidez, kontrolatu beharreko hiru propietateen azterketa

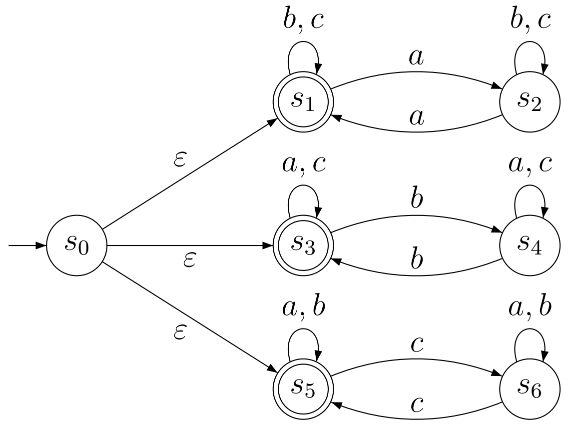


3.4.15 irudia. a, b edo c kopuru bikoitian duten hitzez osatutako lengoaiari dagokion AFEDa.

bereiztea lortzen da: r_0, r_1 eta r_2 egoeren bidez a kopurua bikoitia izatearen propietatea egiaztatzen da; r_0, r_3 eta r_4 egoeren bidez b kopurua bikoitia izatearen propietatea kontrolatzen da; azkenik, r_0, r_5 eta r_6 egoeren bidez c kopurua bikoitia al den aztertzen da. AFED hau aurretik eman den AFDa baino ulerterrazagoa da baina r_0 egoera hiru propietateetan behar izateak eta, ondorioz, r_0 egoera eta r_1, r_3 eta r_5 egoeren arteko loturak oraindik nahasmena edo argitasun eza sor dezake.

Adibide honetako hirugarren irudian, hau da, 3.4.16 irudian, lengoaia bera definitzen duen ε-AFED bat erakusten da. Hor, s_0 egoeratik ateratzen diren ε trantsizioek aztertu beharreko hiru propietateak (a kopurua bikoitia izatea edo b kopurua bikoitia izatea edo c kopurua bikoitia izatea) guztiz bereiztea ahalbidetzen dute. Horretarako, s_0 egoeratik abiatzen bagara, zuzenean s_1, s_3 eta s_5 egoeretara igaroko gara. Alde batetik, s_1 eta s_2 egoeren bidez hitzean a kopurua bikoitia izatearen propietatea kontrolatuko da. Beste aldetik, s_3 eta s_4 egoeren bidez hitzean b kopurua bikoitia izatearen propietatea kontrolatuko da. Eta azkenik, s_5 eta s_6 egoeren bidez hitzean c kopurua bikoitia izatearen propietatea aztertu edo kontrolatuko da. Hitzak propietate horietakoren bat betetzen badu, “Bai” erantzungo du automatak, baina hitzak propietate horietatik bat bera ere ez badu betetzen, orduan automatak “Ez” erantzungo du. Adibide honek garbi erakusten du ε-AFEDak erabiliz automaten diseinua azko sinplifika daitekeela.

Jarraian 3.4.16 irudiko ε-AFEDaren definizio formala emango da. ε-AFED hori honako boskote hau da:



3.4.16 irudia. a, b edo c kopuru bikoitian duten hitzez osatutako lengoaiari dagokion ε-AFEDa.

$$(S, A, \lambda, s_0, Y)$$

Boskote horretan,

- $S = \{s_0, s_1, s_2, s_3, s_4, s_5, s_6\}$
- $A = \{a, b, c\}$
- $\lambda : S \times (A \cup \{\varepsilon\}) \to 2^S$ honako taula honen bidez definitutako trantsizio-funtzioa da:

λ	a	b	c	ε
s_0	$\varnothing$	$\varnothing$	$\varnothing$	$\{s_1, s_3, s_5\}$
s_1	$\{s_2\}$	$\{s_1\}$	$\{s_1\}$	$\varnothing$
s_2	$\{s_1\}$	$\{s_2\}$	$\{s_2\}$	$\varnothing$
s_3	$\{s_3\}$	$\{s_4\}$	$\{s_3\}$	$\varnothing$
s_4	$\{s_4\}$	$\{s_3\}$	$\{s_4\}$	$\varnothing$
s_5	$\{s_5\}$	$\{s_5\}$	$\{s_6\}$	$\varnothing$
s_6	$\{s_6\}$	$\{s_6\}$	$\{s_5\}$	$\varnothing$

Taula horren bidez $\lambda(s_0, a) = \varnothing$ dela, $\lambda(s_0, \varepsilon) = \{s_1, s_3, s_5\}$ dela eta abar adierazten da.

- s_0 hasierako egoera da. Hasierako egoera beti bakarra da.
- $Y = \{s_1, s_3, s_5\}$. Kasu honetan s_0 egoera Y multzoan sar daiteke baina ez da beharrezkoa, izan ere s_0 egoeratik s_1, s_3 eta s_5 egoeretara igaro gaitezke sinbolorik kontsumitu gabe.

Orain $(\{q_0\}, aca)$ konfigurazio ez-deterministari dagokion konputazioa garatuko da urratsez urrats:

1. urratsa: $\lambda^*(\{s_0\}, aca) = \lambda^*(\lambda(s_0, a) \cup \lambda(s_1, a) \cup \lambda(s_3, a) \cup \lambda(s_5, a), ca) = \lambda^*(\{s_2, s_3, s_5\}, ca)$. Hor, Itxidura-$\varepsilon^*(\{s_0\}, \lambda) = \{s_0, s_2, s_3, s_5\}$ dela kontuan hartu behar izan da.

2. urratsa: $\lambda^*(\{s_2, s_3, s_5\}, ca) = \lambda^*(\lambda(s_2, c) \cup \lambda(s_3, c) \cup \lambda(s_5, c), a) = \lambda^*(\{s_2, s_3, s_6\}, a)$. Kasu honetan Itxidura-$\varepsilon^*(\{s_2, s_3, s_5\}, \lambda) = \{s_2, s_3, s_5\}$ dela kontuan hartu da.

3. urratsa: $\lambda^*(\{s_2, s_3, s_6\}, a) = \lambda^*(\{s_2, s_3, s_6\}, a\varepsilon) = \lambda^*(\lambda(s_2, a) \cup \lambda(s_3, a) \cup \lambda(s_6, a), \varepsilon) = \lambda^*(\{s_1, s_3, s_6\}, \varepsilon)$. Kalkulu hori egitean Itxidura-$\varepsilon^*(\{s_2, s_3, s_6\}, \lambda) = \{s_2, s_3, s_6\}$ dela kontuan hartu da.

4. urratsa: $\lambda^*(\{s_1, s_3, s_6\}, \varepsilon) = \{s_1, s_3, s_6\}$. Izan ere, Itxidura-$\varepsilon^*(\{s_1, s_3, s_6\}, \lambda) = \{s_1, s_3, s_6\}$.

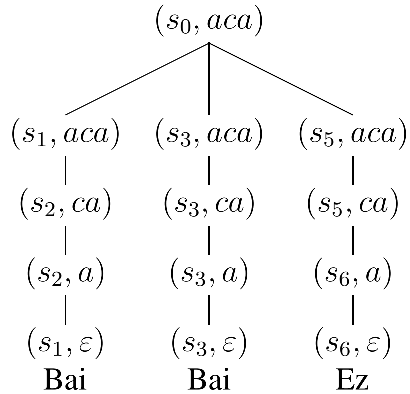
Konputazioa $\{s_1, s_3, s_6\}$ egoera-multzoan bukatu denez eta $\{s_1, s_3, s_6\}$ multzoaren eta Y multzoaren arteko ebakidura hutsa ez denez, badakigu hitza ε-AFEDak definitutako lengoaiakoa dela (a, b edo c kopuru bikoitian agertzen da).

Konputazio hori era grafikoagoan ere adieraz daiteke konfigurazio ez-deterministez osatutako honako sekuentzia honen bidez:

$$
\begin{array}{c}
(\{s_0\}, aca) \equiv_{\text{Itxidura-}\varepsilon^*} (\{s_0, s_1, s_3, s_5\}, aca) \\
| \\
(\{s_2, s_3, s_5\}, ca) \equiv_{\text{Itxidura-}\varepsilon^*} (\{s_2, s_3, s_5\}, ca) \\
| \\
(\{s_2, s_3, s_6\}, a) \equiv_{\text{Itxidura-}\varepsilon^*} (\{s_2, s_3, s_6\}, a) \\
| \\
(\{s_1, s_3, s_6\}, \varepsilon) \equiv_{\text{Itxidura-}\varepsilon^*} (\{s_1, s_3, s_6\}, \varepsilon) \\
\text{Bai}
\end{array}
$$

Lerro bakoitzean, $\equiv$ sinboloaren ezkerrean, une horretako egoera-multzoa zein den zehazten da eta $\equiv$ sinboloaren eskuinean, egoera horietatik ε trantsizioen bidez irisgarriak diren egoerak agertzen dira, hau da, Itxidura-$\varepsilon^{\#}$. Konputazioa ε hitza duen konfigurazio ez-determinista batera iritsi delako bukatu da.

Konputazio hori konfigurazio deterministez eratutako honako zuhaitz honen bidez ere adieraz daiteke:



Ezkerretik hasita, lehenengo eta bigarren adarretan Y multzokoa den egoera batez eta hitz hutsaz osatutako konfigurazio bat edukitzea lortu da: (s_1, ε) batean eta (s_3, ε) bestean. Lehenengo adarraren bidez, hitzak a kopuru bikoitia duela detektatu da eta bigarren adarraren bidez, hitzak b kopuru bikoitia duela detektatu da. Hirugarrengo adarrean hitz hutsa duen (s_6, ε) konfigurazioa lortu da baina s_6 ez da Y-koa. Hirugarren adarraren bidez c kopurua bikoitia al den aztertu da baina kasu horretan ezezkoa atera da, izan ere, c kopurua bakoitia da. Guztira, Y multzokoa den egoera batez eta hitz hutsaz osatutako konfigurazio bat gutxienez badagoenez, ε-AFEDak baiezko erantzuna emango du.

Orain $(\{q_0\}, abc)$ konfigurazio ez-deterministari dagokion konputazioa garatuko da urratsez urrats:

1. urratsa: $\lambda^*(\{s_0\}, abc) = \lambda^*(\lambda(s_0, a) \cup \lambda(s_1, a) \cup \lambda(s_3, a) \cup \lambda(s_5, a), bc) = \lambda^*(\{s_2, s_3, s_5\}, bc)$. Kaluku hori egitean Itxidura-$\varepsilon^*(\{s_0\}, \lambda) = \{s_0, s_2, s_3, s_5\}$ dela kontuan hartu behar da.

2. urratsa: $\lambda^*(\{s_2, s_3, s_5\}, bc) = \lambda^*(\lambda(s_2, b) \cup \lambda(s_3, b) \cup \lambda(s_5, b), c) = \lambda^*(\{s_2, s_4, s_5\}, c)$. Kasu honetan Itxidura-$\varepsilon^*(\{s_2, s_3, s_5\}, \lambda) = \{s_2, s_3, s_5\}$ dela kontuan hartu behar da.

3. urratsa: $\lambda^*(\{s_2, s_4, s_5\}, c) = \lambda^*(\{s_2, s_4, s_5\}, c\varepsilon) = \lambda^*(\lambda(s_2, c) \cup \lambda(s_4, c) \cup \lambda(s_5, c), \varepsilon) = \lambda^*(\{s_2, s_4, s_6\}, \varepsilon)$. Hor, Itxidura-$\varepsilon^*(\{s_2, s_4, s_5\}, \lambda) = \{s_2, s_4, s_5\}$ dela kontuan hartu da.

4. urratsa: $\lambda^*(\{s_3, s_4, s_6\}, \varepsilon) = \{s_2, s_4, s_6\}$. Izan ere, Itxidura-$\varepsilon^*(\{s_2, s_4, s_6\}, \lambda) = \{s_2, s_4, s_6\}$.

Konputazioa $\{s_2, s_4, s_6\}$ egoera-multzoan bukatu denez eta $\{s_2, s_4, s_6\}$ multzoa eta Y multzoaren arteko ebakidura hutsa denez, badakigu hitza ez dela ε-AFEDak definitutako lengoaiakoa (ez a, ez b eta ez c ez dira agertzen kopuru bikoitian).

Konputazio hori era grafikoagoan ere adieraz daiteke konfigurazio ez-deterministez osatutako honako sekuentzia honen bidez:

$$
\begin{array}{c}
(\{s_0\}, abc) \equiv_{\mathsf{Itxidura}\text{-}\varepsilon} (\{s_0, s_1, s_3, s_5\}, abc) \\
| \\
(\{s_2, s_3, s_5\}, bc) \equiv_{\mathsf{Itxidura}\text{-}\varepsilon} (\{s_2, s_3, s_5\}, bc) \\
| \\
(\{s_2, s_4, s_5\}, c) \equiv_{\mathsf{Itxidura}\text{-}\varepsilon} (\{s_2, s_4, s_5\}, c) \\
| \\
(\{s_2, s_4, s_6\}, \varepsilon) \equiv_{\mathsf{Itxidura}\text{-}\varepsilon} (\{s_2, s_4, s_6\}, \varepsilon) \\
\text{Ez}
\end{array}
$$

Lerro bakoitzean, $\equiv$ sinboloaren ezkerrean, une horretako egoera-multzoa zein den zehazten da eta $\equiv$ sinboloaren eskuinean, egoera horietatik ε trantsizioen bidez irisgarriak diren egoerak agertzen dira, hau da, $\mathsf{Itxidura}\text{-}\varepsilon^{\#}$. Konputazioa ε hitza duen konfigurazio ez-determinista batera iritsi delako bukatu da.

Konputazio hori trantsizio deterministez eratutako honako zuhaitz honen bidez ere adieraz daiteke:

$$
\begin{array}{ccc}
 & (s_0, abc) & \\
\diagup & | & \diagdown \\
(s_1, abc) & (s_3, abc) & (s_5, abc) \\
| & | & | \\
(s_2, bc) & (s_3, bc) & (s_5, bc) \\
| & | & | \\
(s_2, c) & (s_4, c) & (s_5, c) \\
| & | & | \\
(s_2, \varepsilon) & (s_4, \varepsilon) & (s_6, \varepsilon) \\
\text{Ez} & \text{Ez} & \text{Ez}
\end{array}
$$

Adar guztietan hitz osoa irakurtzea lortu da, baina hitz hutsa duten konfigurazioetan agertzen diren egoerak (s_2, s_4 eta s_6) ez dira Y-koak. Beraz, ε-AFEDak ezetz erantzungo duela ondoriozta dezakegu.

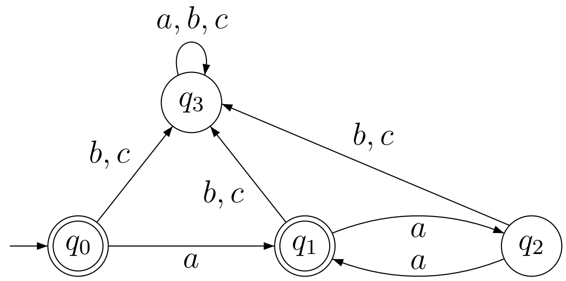
Adar denak Y multzokoa ez den egoera batez eta ε hitz hutsaz eratutako konfigurazio determinista batean bukatzen direnez, abc hitza ez da adibide honetako lengoaiakoa.

3.4.1.4.8 ε hitzaz eta a sinboloaren errepikapenez osatutako luzera bakoitiko hitzez osatutako lengoaiari dagokion ε-AFEDa

Adibide honetako ε-AFEDak a, b eta c sinboloak izan ditzaketen hitzak hartuko ditu sarrerako datutzat. ε-AFEDak, datu gisa jaso duen hitza ε hitza edo a sinboloaren errepikapenez osatutako luzera bakoitiko hitz bat al den erabakiko du. Lengoaia horren definizio formala honako hau da:

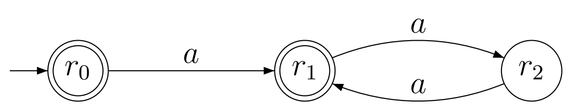
$$\{w \mid w \in A^* \wedge (w = \varepsilon \vee (|w| = |w|_a \wedge |w| \bmod 2 \neq 0))\}$$

Baina ε-AFED hori aurkeztu aurretik, lengoaia bera definitzen duten AFD bat eta AFED bat erakutsiko dira. Horrela hiru formalismo horiek alderatu ahal izango ditugu eta gehienetan ε-AFEDak AFEDak eta AFDak baino ulerterrazagoak direla ikusiko dugu. Era berean, AFEDak AFDak baino ulerterrazagoak direla ere erakutsiko da. Izan ere, adibide honetan kontrolatu beharreko baldintzak bakoitza bere aldetik aztertu ahal izango dira AFEDa erabiliz, eta horrela diseinua erraztuz.



3.4.17 irudia. ε hitzaz eta a sinboloaren errepikapenez osatutako luzera bakoitiko hitzez osatutako lengoaiari dagokion AFDa.

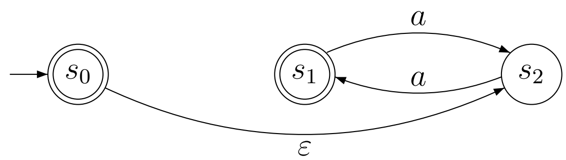
Hasteko, 3.4.17 irudian adibide honetako lengoaiari dagokion AFD bat dugu.



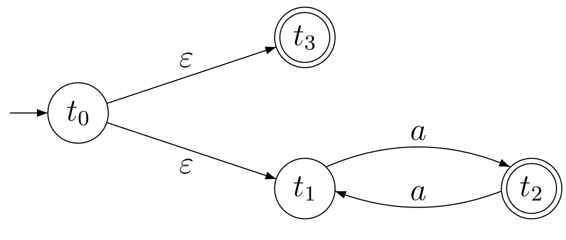
3.4.18 irudia. ε hitzaz eta a sinboloaren errepikapenez osatutako luzera bakoitiko hitzez osatutako lengoaiari dagokion AFEDa.

Bigarren lekuan, 3.4.18 irudian, adibide honetako lengoaiari dagokion AFED bat dugu. AFED horren bidez diseinua sinplifikatu egin da aurreko AFDarekiko, erantzuna baiezkoa izateko bete beharrekoa bakarrik kontrolatu behar baita.

Hirugarren lekuan, 3.4.19 irudian, lengoaia berari dagokion ε-AFED bat dugu. La transición ε que sale desde el estado s_0 egoeratik ateratzen den ε trantsizio hutsak aztertu beharreko propietate biak banantzeko balio digu: hitz hutsa izatea edo a-ren errepikapenez osatutako luzera bakoitiko hitza izatea. Oraindik ere argiagoa den beste ε-AFED bat erakusten da 3.4.20 irudian.



3.4.19 irudia. ε hitzaz eta a sinboloaren errepikapenez osatutako luzera bakoitiko hitzez osatutako lengoaiari dagokion ε-AFED bat.



3.4.20 irudia. ε hitzaz eta a sinboloaren errepikapenez osatutako luzera bakoitiko hitzez osatutako lengoaiari dagokion beste ε-AFED bat.

Orain, 3.4.19 irudia erakusten den ε-AFEDaren osagaiak definituko dira. ε-AFED hori honako boskote honen bidez formaliza daiteke:

$$(S, A, \lambda, s_0, Y)$$

Boskote horretako osagaiak honako hauek dira:

- $S = \{s_0, s_1, s_2\}$
- $A = \{a, b, c\}$
- $\lambda : S \times (A \cup \{\varepsilon\}) \rightarrow 2^S$ honako taula honen bidez definitutako trantsizio-funtzioa da:

λ	a	b	c	ε
s_0	$\varnothing$	$\varnothing$	$\varnothing$	$\{s_2\}$
s_1	$\{s_2\}$	$\varnothing$	$\varnothing$	$\varnothing$
s_2	$\{s_1\}$	$\varnothing$	$\varnothing$	$\varnothing$

Taula horretan $\lambda(s_0, a) = \varnothing$ dela, $\lambda(s_0, \varepsilon) = \{s_2\}$ dela eta abar adierazten da.

- s_0 hasierako egoera da. Hasierako egoera beti bakarra izango da.

- $Y = \{s_0, s_1\}$. Hau onarpen egoeren multzoa da.

Jarraian, $(\{s_0\}, aca)$ konfigurazio ez-deterministari dagokion konputazioa garatuko da urratsez urrats:

1. urratsa: $\lambda^*(\{s_0\}, aca) = \lambda^*(\lambda(s_0, a) \cup \lambda(s_2, a), ca) = \lambda^*(\{s_1\}, ca)$. Urrats honetan Itxidura-$\varepsilon^{\#}(\{s_0\}, \lambda) = \{s_0, s_2\}$ dela hartu da kontuan.

2. urratsa: $\lambda^*(\{s_1\}, ca) = \lambda^*(\lambda(s_1, c), a) = \lambda^*(\varnothing, a)$. Kalkulu horretan Itxidura-$\varepsilon^{\#}(\{s_1\}, \lambda) = \{s_1\}$ dela kontuan izan da.

3. urratsa: $\lambda^*(\varnothing, a) = \varnothing$.

Konputazioa $\varnothing$ multzoa itzuliz bukatzen denez eta $\varnothing$ eta Y-ren arteko ebakidura hutsa denez, aca hitza ez da lengoaiakoa.

Konputazio hori konfigurazio ez-deterministez eratutako honako sekuentzia honen bidez adieraz daiteke grafikoki:

$$
\begin{array}{c}
(\{s_0\}, aca) \equiv_{\text{Itxidura-}\varepsilon^{\#}} (\{s_0, s_2\}, aca) \\
| \\
(\{s_1\}, ca) \equiv_{\text{Itxidura-}\varepsilon^{\#}} (\{s_1\}, ca) \\
| \\
(\varnothing, a) \equiv_{\text{Itxidura-}\varepsilon^{\#}} (\varnothing, a) \\
\text{Ez}
\end{array}
$$

Lerro bakoitzean, $\equiv$ sinboloaren ezkerrean, une horretako egoera-multzoa zein den zehazten da eta $\equiv$ sinboloaren eskuinean, egoera horietatik ε trantsizioen bidez irisgarriak diren egoerak agertzen dira, hau da, Itxidura-$\varepsilon^{\#}$. Konputazioa $\varnothing$ egoera-multzo hutsa duen konfigurazio ez-determinista batera iritsi delako bukatu da. Erantzuna ezezkoa da $\varnothing$ multzoan ez dagoelako Y-ko elementurik.

Konputazio hori konfigurazio deterministez eratutako honako zuhaitz honen bidez ere adieraz daiteke:

$$
\begin{array}{c}
(s_0, aca) \\
| \\
(s_2, aca) \\
| \\
(s_1, ca) \\
\text{Ez}
\end{array}
$$

Zuhaitzak adar bakarra du. Konfigurazio deterministez osatutako zuhaitzetan, hitz ez-hutsa baldin badugu baina irakurtzen jarraitzerik ez badago, bukatu egin behar da. ε-AFEDak ezezkoa erantzungo du ez delako lortu hitz hutsaz eta Y-ko egoera batez eratutako konfigurazio bat edukitzea.

Jarraian $(\{s_0\}, \varepsilon)$ konfigurazio ez-deterministari dagokion konputazioa garatuko da urratsez urrats:

1. urratsa: $\lambda^*(\{s_0\}, \varepsilon) = \{s_0, s_2\}$.

Konputazioa $\{s_0, s_2\}$ multzoa itzuliz bukatzen da. $\{s_0, s_2\}$ multzoa eta Y-ren arteko ebakidura hutsa ez denez, ε hitza lengoaiakoa dela ondorioztа dezakegu.

Konputazio hori konfigurazio ez-deterministez eratutako honako sekuentzia honen bidez adieraz daiteke grafikoki:

$$(\{s_0\}, \varepsilon) \equiv_{\mathsf{Itxidura}\text{-}\varepsilon^{\#}} (\{s_0, s_2\}, \varepsilon)$$

Bai

Adabegi bakar horretan, $\equiv$ sinboloaren ezkerrean, une horretako egoera-multzoa zein den zehazten da eta $\equiv$ sinboloaren eskuinean, egoera horietatik ε trantsizioen bidez irisgarriak diren egoerak agertzen dira, hau da, Itxidura-$\varepsilon^{\#}$. Konputazioa ε hitza duen konfigurazio ez-determinista batera iritsi delako bukatu da. Erantzuna baiezkoa da $\{s_0, s_2\}$ multzoan Y-ko elementu bat dagoelako.

Konputazio hori konfigurazio deterministez eratutako honako zuhaitz honen bidez ere adieraz daiteke:

$$\begin{array}{c} (s_0, \varepsilon) \\ | \\ (s_2, \varepsilon) \end{array}$$

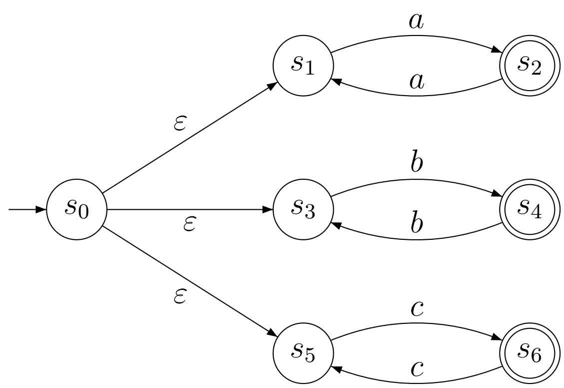
Bai

Zuhaitzak adar bakarra du. λ kontuan hartuz aurrera egiterik ez dagoelako bukatu da adar hori. Ez da bukatu adabegi bat errepikatu delako. Adar horretan, ε hitz hutsaz eta Y-koa den egoera batez osatutako konfigurazio bat dugunez, (s_0, ε) konfigurazioa hain zuzen ere, ε-AFEDaren erantzuna baiezkoa izango dela ondorioztа dezakegu. Adibide honetan ikusten da konfigurazio batean hitz hutsa lortu arren aurrera egin behar dela λ funtzioak hori egitea ahalbidetzen duen bitartean edo errepikatutako adabegi bat agertu arte. Kasu honetan λ funtzioak ez du ahalbidetzen (s_2, ε) konfiguraziotik beste konfiguraziorik eraikitzea.

3.4.1.4.9 a sinboloaren errepikapenez osatutako luzera bakoitiko hitzez, b sinboloaren errepikapenez osatutako luzera bakoitiko hitzez eta c sinboloaren errepikapenez osatutako luzera bakoitiko hitzez eratutako lengoaiari dagokion ε-AFEDa

Adibide honetako ε-AFEDak a, b eta c sinboloak izan ditzaketen hitzak hartuko ditu sarrerako datutzat. ε-AFEDak, datu gisa jaso duen hitza a sinboloaren errepikapenez osatutako luzera bakoitiko hitz bat al den, b sinboloaren errepikapenez osatutako luzera bakoitiko hitz bat al den edo c sinboloaren errepikapenez osatutako luzera bakoitiko hitz bat al den erabakiko du. Lengoaia horren definizio formala honako hau da:

$$\{w \mid w \in A^* \wedge |w| \bmod 2 \neq 0 \wedge (|w| = |w|_a \vee |w| = |w|_b \vee |w| = |w|_c)\}$$



3.4.21 irudia. a sinboloaren errepikapenez osatutako luzera bakoitiko hitzez, b sinboloaren errepikapenez osatutako luzera bakoitiko hitzez eta c sinboloaren errepikapenez osatutako luzera bakoitiko hitzez eratutako lengoaiari dagokion ε-AFEDa.

Adibide honetako ε-AFEDaren trantsizio-diagrama 3.4.21 irudian dugu. Las transiciones que salen desde el estado s_0 egoeratik ateratzen diren ε trantsizioak direla-eta, aztertu beharreko hiru propietateak guztiz banantzea lortzen da: a sinboloaren errepikapenez osatutako luzera bakoitiko hitza izatea, b sinboloaren errepikapenez osatutako luzera bakoitiko hitza izatea eta c sinboloaren errepikapenez osatutako luzera bakoitiko hitza izatea. s_0 egoeratik abiatzen bagara, zuzenean s_1, s_3 eta s_5 egoeretara joango gara sinbolorik kontsumitu gabe. s_1 eta s_2 egoeren bidez a sinboloaren errepikapenez osatutako luzera bakoitiko hitza izatearen propietatea kontrolatzen da. s_3 eta s_4 egoeren bidez b sinboloaren errepikapenez osatutako luzera bakoitiko hitza izatearen propietatea kontrolatzen da. Bukatzeko, s_5 eta s_6 egoeren bidez c sinboloaren errepikapenez osatutako luzera bakoitiko hitza izatearen propietatea kontrolatzen da.

Orain, 3.4.21 irudiko ε-AFEDaren osagaiak definituko dira. ε-AFED honako boskote honen bidez formaliza daiteke:

$$(S, A, \lambda, s_0, Y)$$

Boskote horretako osagaiak honako hauek dira:

- $S = \{s_0, s_1, s_2, s_3, s_4, s_5, s_6\}$
- $A = \{a, b, c\}$
- $\lambda : S \times (A \cup \{\varepsilon\}) \to 2^S$ honako taularen bidez definitutako trantsizio-funtzioa da:

λ	a	b	c	ε
s_0	$\varnothing$	$\varnothing$	$\varnothing$	$\{s_1, s_3, s_5\}$
s_1	$\{s_2\}$	$\varnothing$	$\varnothing$	$\varnothing$
s_2	$\{s_1\}$	$\varnothing$	$\varnothing$	$\varnothing$
s_3	$\varnothing$	$\{s_4\}$	$\varnothing$	$\varnothing$
s_4	$\varnothing$	$\{s_3\}$	$\varnothing$	$\varnothing$
s_5	$\varnothing$	$\varnothing$	$\{s_6\}$	$\varnothing$
s_6	$\varnothing$	$\varnothing$	$\{s_5\}$	$\varnothing$

Taula horretan $\lambda(s_0, a) = \varnothing$ dela, $\lambda(s_0, \varepsilon) = \{s_1, s_3, s_5\}$ dela eta abar adierazten da.

- s_0 hasierako egoera da. Hasierako egoera bakarra da beti.
- $Y = \{s_2, s_4, s_6\}$. Hau onarpen egoeren multzoa da.

Jarraian $(\{q_0\}, \varepsilon)$ konfigurazio ez-deterministari dagokion konputazioa garatuko da urratsez urrats:

1. urratsa: $\lambda^*(\{s_0\}, \varepsilon) = \{s_0, s_1, s_3, s_5\}$.

Konputazioak $\{s_0, s_1, s_3, s_5\}$ multzoa itzultzen duenez eta $\{s_0, s_1, s_3, s_5\}$ multzoaren eta Y multzoaren arteko ebakidura hutsa denez, hitza ez dela lengoaiakoa esango du ε-AFEDak.

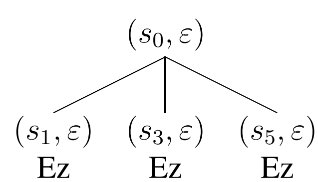
Konputazio hori konfigurazio ez-deterministez eratutako honako sekuentzia honen bidez adieraz daiteke grafikoki:

$$(\{s_0\}, \varepsilon) \equiv_{\mathsf{Itxidura}\text{-}\varepsilon^{\#}} (\{s_0, s_1, s_3, s_5\}, \varepsilon)$$

Ez

Adabegi bakar horretan, $\equiv$ sinboloaren ezkerrean, une horretako egoera-multzoa zein den zehazten da eta $\equiv$ sinboloaren eskuinean, egoera horietatik ε trantsizioen bidez irisgarriak diren egoerak agertzen dira, hau da, Itxidura-$\varepsilon^{\#}$. Konputazioa ε hitza duen konfigurazio ez-determinista batera iritsi delako bukatu da. Erantzuna ezezkoa dela ondoriozta dezakegu, azkeneko konfigurazioko egoera-multzoaren eta Y-ren arteko ebakidura hutsa delako.

Konputazio hori konfigurazio deterministez eratutako honako zuhaitz honen bidez ere adieraz daiteke:



λ kontuan hartuz aurrera egiterik ez dagoelako bukatu dira adar denak. Bat bera ere ez da bukatu adabegia errepikatu delako. Adar guztietan hitz osoa irakurtzea lortu da, baina hitz hutsa duten konfigurazioetan agertzen diren egoerak (s_0, s_1, s_3 eta s_5) ez dira Y-koak. Beraz, ε-AFEDak ezetz erantzungo duela ondorioztа dezakegu.

Jarraian $(\{q_0\}, ccc)$ konfigurazio ez-deterministari dagokion konputazioa garatuko da urratsez urrats:

1. urratsa: $\lambda^*(\{s_0\}, ccc) = \lambda^*(\lambda(s_0, c) \cup \lambda(s_1, c) \cup \lambda(s_3, c) \cup \lambda(s_5, c), cc) = \lambda^*(\{s_6\}, cc)$. Urrats honeta Itxidura-$\varepsilon^{\#}(\{s_0\}, \lambda) = \{s_0, s_1, s_3, s_5\}$ dela kontuan hartu da.

2. urratsa: $\lambda^*(\{s_6\}, cc) = \lambda^*(\lambda(s_6, c), c) = \lambda^*(\{s_5\}, c)$. Kalkulu hori egitean Itxidura-$\varepsilon^{\#}(\{s_6\}, \lambda) = \{s_6\}$ dela kontuan izan da.

3. urratsa: $\lambda^*(\{s_5\}, c) = \lambda^*(\lambda(s_5, c), \varepsilon) = \lambda^*(\{s_6\}, \varepsilon)$. Urrats honetan Itxidura-$\varepsilon^{\#}(\{s_5\}, \lambda) = \{s_5\}$ daukagu.

4. urratsa: $\lambda^*(\{s_6\}, \varepsilon) = \{s_6\}$. Izan ere, Itxidura-$\varepsilon^{\#}(\{s_6\}, \lambda) = \{s_6\}$.

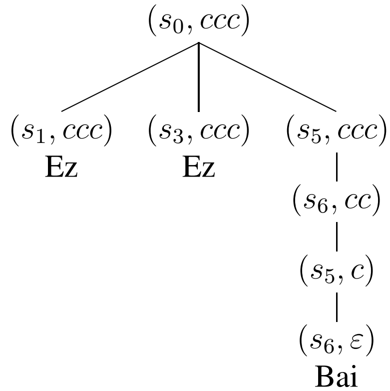
Konputazioak $\{s_6\}$ egoera-multzoa itzultzen duenez eta $\{s_6\}$ eta Y-ren arteko ebakidura hutsa ez denez, ccc hitza lengoaiakoa dela ondorioztа dezakegu.

Konputazio hori konfigurazio ez-deterministez eratutako honako sekuentzia honen bidez adieraz daiteke grafikoki:

$$
\begin{array}{rcl}
(\{s_0\}, ccc) & \equiv_{\text{Itxidura-}\varepsilon^{\#}} & (\{s_0, s_1, s_3, s_5\}, ccc) \\
 & | & \\
(\{s_6\}, cc) & \equiv_{\text{Itxidura-}\varepsilon^{\#}} & (\{s_6\}, cc) \\
 & | & \\
(\{s_5\}, c) & \equiv_{\text{Itxidura-}\varepsilon^{\#}} & (\{s_5\}, c) \\
 & | & \\
(\{s_6\}, \varepsilon) & \equiv_{\text{Itxidura-}\varepsilon^{\#}} & (\{s_6\}, \varepsilon) \\
 & \text{Bai} &
\end{array}
$$

Lerro bakoitzean, $\equiv$ sinboloaren ezkerrean, une horretako egoera-multzoa zein den zehazten da eta $\equiv$ sinboloaren eskuinean, egoera horietatik ε trantsizioen bidez irisgarriak diren egoerak agertzen dira, hau da, Itxidura-$\varepsilon^{\#}$. Konputazioa ε hitza duen konfigurazio ez-determinista batera iritsi delako bukatu da. Erantzuna baiezkoa da azkeneko konfigurazio horretako egoera multzoan Y-ko elementu bat dugulako.

Konputazio hori konfigurazio deterministez eratutako honako zuhaitz honen bidez ere adieraz daiteke:



λ kontuan hartuz aurrera egiterik ez dagoelako bukatu dira adar denak. Bat bera ere ez da bukatu adabegia errepikatu delako. Ezkerretik hasita, lehenengo bi adarretan ez da lortu hitza osorik irakurtzea. Hirugarren adarrean hitz osoa irakurri da eta hitz hutsa duen konfigurazioan agertzen den egoera (s_6) Y-koa da. Beraz, ε-AFEDak ccc hitzarentzat baietz erantzungo duela ondoriozta dezakegu.

3.4.2 ε-AFEDen diseinua

AFDekin eta AFEDekin gertatzen den bezala, ε-AFED baten zeregina hitz bat lengoaia jakin batekoa al den erabakitzea da. Beste era batera esanda, ε-AFED baten zeregina AFED horri sarrerako datutzat emango zaion hitz bakoitzak propietate jakin bat betetzen al duen erabakitzea da. Beraz, AFDetan eta AFEDetan bezala, ε-AFEDa sinboloz sinbolo hitza irakurriz joango da eta irakurriz doan hitz horrek betetzen duen propietatea zein den "gogoratzeko" baliabide gisa egoerak erabiliko ditu. Egoera bakoitzak propietate bat "gogoratzen" edo adierazten du. Une bakoitzean, une horretara arte irakurri den hitz zatiak (edo azpihitzak) zein propietate betetzen duen jakiteko ε-AFEDa zein egoeratan dagoen begiratu beharko da. ε-AFEDak hitza sinboloz sinbolo irakurriko du eta ezingo du atzeraka egin. Beraz, une bakoitzean ε-AFEDa zein egoeratan dagoen jakinda, une horretara arte irakurri den hitz zatiak zein propietate betetzen duen jakingo da eta hitza osatzen duten gainerako sinboloak irakurriz jarraitu beharko da. **Automaten diseinu metodikoan edo sistematikoan**, funtsezko oinarria hitzek bete ditzaketen propietate garrantzitsuak edo esanguratsuak "gogoratzeko" beharrezkoak diren egoerak identifikatzea da. Beharrezkoak diren egoerak zein izango diren erabaki ondoren, egoeren arteko trantsizioak finkatu beharko dira. AFDen eta AFEDen diseinurako azaldutako teknika batzuk ε-AFEDen diseinurako ere balio dute. Hain zuzen ere, egoerak identifikatzean oinarritzen den teknikak eta lehenengo egoera-multzoak identifikatu eta gero egoera-multzo horietako egoerak zehaztean oinarritzen den teknikak ε-AFEDentzat ere balio dute. Era berean, konjuntzioa dugunerako AFDentzat 3.2.2.4 atalean eta AFEDentzat 3.3.3.4 atalean azaldu den teknika ε-AFEDentzat ere erabilgarria da baina ε-AFEDetan q egoera batean egotea $\mathsf{Itxidura}\text{-}\varepsilon(q, \lambda)$ multzoko egoera guztietan egotearen baliokidea dela kontuan izan behar da. Aldiz, AFDen kasuan 3.2.2.3 atalean propietateen ukapenarentzat azaldutako teknika eta 3.2.2.5 atalean disjuntzioarentzat azaldutako teknika ezingo dira erabili ε-AFEDen kasuan. Disjuntzioaren kasuan, ε-AFEDak elkartzeko metodo sistematiko bat azalduko da (AFEDentzat 3.3.3.5.2 atalean azaldutakoaren antzekoa). Gogora dezagun propietateen konjuntzioa lengoaien ebakiduraren parekoa dela eta propietateen disjuntzioa lengoaien arteko bilduraren parekoa dela. Diseinatu beharreko automatari dagokion lengoaia lengoaien arteko bildura gisa edo lengoaien kateaketa gisa edo lengoaia baten itxidura unibertsal gisa edo lengoaia baten itxidura positibo gisa edo lengoaia baten alderantzizkoa gisa defini baldin badaiteke, orduan ε-AFEDak oso egokiak dira, diseinua era sistematikoan egiteko aukera ematen baitute.

3.4.2.1 ε-AFEDen diseinua lengoaien kateaketa dugunerako

L_1 eta L_2 lengoaiei dagozkien E_1 eta E_2 ε-AFEDak baldin baditugu, L_1L_2 lengoaiari dagokion ε-AFED bat diseinatzeko nahikoa da E_1-eko onarpen egoera bakoitzetik trantsizio huts bat (ε trantsizio bat) ipintzea E_2-ko hasierako egoerara. Gainera, horrela eraikitako ε-AFED berrian E_1-eko egoerak ez dira onarpen egoerak izango.

Formalki, demagun L_1 eta L_2 lengoaien kateadura gisa defini daitekeen L lengoaiarentzat ε-AFED bat diseinatu nahi dugula. Beraz, $L = L_1L_2$. Demagun baita ere, L_1 eta L_2 lengoaientzat $E_1 = (Q, A, \lambda_1, q_0, Y_1)$ eta $E_2 = (R, A, \lambda_2, r_0, Y_2)$ ε-AFEDak diseinatuta ditugula.

L lengoaiarentzat $E = (S, A, \lambda, s_0, Y)$ ε-AFED bat honela defini daiteke:

- $S = Q \cup R$.
- A alfabetoa E_1-eko eta E_2-ko alfabeto bera izango da.
- s_0 hasierako egoera q_0 izango da.
- Onarpen egoerei dagokienez, $Y = Y_2$.
- Bukatzeko, λ trantsizio-funtzioa $S \times (A \cup \{\varepsilon\}) \to 2^S$ motakoa izango da eta honako hau beteko da:
  - $\lambda(q_i, \alpha) = \lambda_1(q_i, \alpha)$ izango da $q_i \in Q$ egoera bakoitzarentzat eta $\alpha \in A$ sinbolo bakoitzarentzat.
  - $\lambda(q_i, \varepsilon) = \lambda_1(q_i, \varepsilon)$ izango da $q_i \in Q \setminus Y_1$ egoera bakoitzarentzat.
  - $\lambda(q_i, \varepsilon) = \lambda_1(q_i, \varepsilon) \cup \{r_0\}$ izango da $q_i \in Y_1$ egoera bakoitzarentzat.
  - $\lambda(r_i, \alpha) = \lambda_2(r_i, \alpha)$ izango da $r_i \in R$ egoera bakoitzarentzat eta $\alpha \in A$ sinbolo bakoitzarentzat.
  - $\lambda(r_i, \varepsilon) = \lambda_2(r_i, \varepsilon)$ izango da $r_i \in R$ egoera bakoitzarentzat.

Orain bi adibide emango dira teknika hori nola aplikatzen den erakusteko.

3.4.2.1.1 a-ren agerpen denak b-ren lehenengo agerpena baino lehenago dituzten eta gutxienez a bat eta b bat dituzten hitzen lengoaiari dagokion ε-AFEDa

Har dezagun $A = \{a, b, c\}$ alfabetoa eta a-ren agerpen denak b-ren lehenengo agerpena baino lehenago dituzten eta gutxienez a bat eta b bat dituzten hitzez osatutako L lengoaia. Lengoaia hori honela formaliza daiteke:

$$L = \{w | w \in A^* \wedge \exists u, v (u \in A^* \wedge v \in A^* \wedge |u|_b = 0 \wedge |v|_a = 0 \wedge w = uv) \wedge |w|_a \geq 1 \wedge |w|_b \geq 1\}$$

Har ditzagun baita ere L_1 eta L_2 lengoaiak:

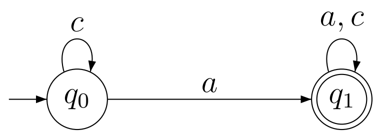
$$L_1 = \{w | w \in A^* \wedge |w|_b = 0 \wedge |w|_a \geq 1\}$$
$$L_2 = \{w | w \in A^* \wedge |w|_a = 0 \wedge |w|_b \geq 1\}$$

L lengoaia L_1 eta L_2 lengoaien kateadura gisa defini daiteke: $L = L_1 L_2$.

3.4.22 irudian L_1 lengoaiari dagokion E_1 ε-AFEDaren trantsizio-diagrama erakusten da. E_1 ε-AFEDa honako boskote honen bidez formaliza daiteke:

$$(Q, A, \lambda_1, q_0, Y_1)$$

Boskote horretako osagaiak honako hauek dira:



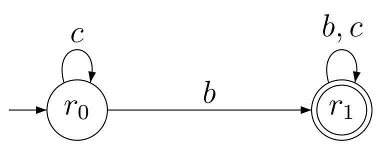
3.4.22 irudia. Gutxienez a bat duten eta b-rik ez duten hitzen lengoaiari dagokion E_1 ε-AFEDa.

- $Q = \{q_0, q_1\}$
- $A = \{a, b, c\}$
- $\lambda_1 : Q \times (A \cup \{\varepsilon\}) \rightarrow 2^Q$ honako taula honen bidez definitutako trantsizio-funtzioa da:

λ_1	a	b	c	ε
q_0	$\{q_1\}$	$\varnothing$	$\{q_0\}$	$\varnothing$
q_1	$\{q_1\}$	$\varnothing$	$\{q_1\}$	$\varnothing$

- q_0 hasierako egoera da.
- $Y_1 = \{q_1\}$.

Ohartu gaitezen E_1-en $\lambda_1(q_i, \varepsilon) = \varnothing$ betetzen dela $q_i \in Q$ egoera denentzat eta, ondorioz, 3.4.22 irudiko trantsizio-diagraman ez dago ε etiketa duen trantsiziorik.



3.4.23 irudia. Gutxienez b bat duten eta a-rik ez duten hitzen lengoaiari dagokion E_2 ε-AFEDa.

3.4.23 irudian L_2 lengoaiari dagokion E_2 ε-AFEDaren trantsizio-diagrama erakusten da. E_2 ε-AFEDa honako boskote honen bidez formaliza daiteke:

$$(R, A, \lambda_2, r_0, Y_2)$$

Boskote horretako osagaiak honako hauek dira:

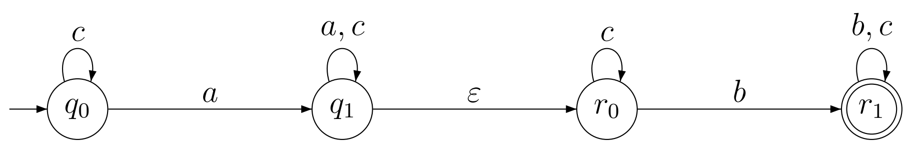
- $R = \{r_0, r_1\}$
- $A = \{a, b, c\}$

- $\lambda_2 : R \times (A \cup \{\varepsilon\}) \to 2^R$ honako taula honen bidez definitutako trantsizio-funtzioa da:

λ_2	a	b	c	ε
r_0	$\varnothing$	$\{r_1\}$	$\{r_0\}$	$\varnothing$
r_1	$\varnothing$	$\{r_1\}$	$\{r_1\}$	$\varnothing$

- r_0 hasierako egoera da.
- $Y_2 = \{r_1\}$.

Ohartu gaitezen E_2-n $\lambda_2(r_j, \varepsilon) = \varnothing$ betetzen dela $r_j \in R$ egoera denentzat eta, ondorioz, 3.4.23 irudiko trantsizio-diagraman ez dago ε etiketa duen trantsiziorik.



3.4.24 irudia. a-ren agerpen denak b-ren lehenengo agerpena baino lehenago dituzten eta gutxienez a bat eta b bat dituzten hitzen lengoaiari dagokion E ε-AFEDa.

3.4.24 irudian L_1L_2 lengoaiari dagokion E ε-AFEDaren trantsizio-diagrama dugu. E ε-AFEDa (S, A, λ, s_0, Y) erako boskote bat gisa formaliza dezakegu eta boskote horretako osagaiak honako hauek izango dira:

- $S = Q \cup R = \{q_0, q_1, r_0, r_1\}$.
- A alfabetoa E_1-eko eta E_2-ko alfabeto bera da.
- s_0 hasierako egoera q_0 da.
- Onarpen egoeren multzoari dagokionez, $Y = Y_2 = \{r_1\}$.
- Azkenik, λ trantsizio-funtzioa $S \times (A \cup \{\varepsilon\}) \to 2^S$ motakoa da eta honako taula honen bidez defini dezakegu:

λ_1	a	b	c	ε
q_0	$\{q_1\}$	$\varnothing$	$\{q_0\}$	$\varnothing$
q_1	$\{q_1\}$	$\varnothing$	$\{q_1\}$	$\{r_0\}$
r_0	$\varnothing$	$\{r_1\}$	$\{r_0\}$	$\varnothing$
r_1	$\varnothing$	$\{r_1\}$	$\{r_1\}$	$\varnothing$

E ε-AFEDan $\lambda(q_1, \varepsilon) = \{r_0\}$ betetzen denez, 3.4.24 irudiko trantsizio-diagraman trantsizio hori ε etiketarekin agertzen da.

3.4.2.1.2 $\{a\}\{b\}^* \cup \{b\}^+\{a\}$ lengoaia eta $\{c\}^+$ lengoaiaren kateadurarentzat ε-AFED bat

Har ditzagun $A = \{a, b, c\}$ alfabetoa eta jarraian adierazten den eran bereizi daitezkeen bi zati dituzten hitzen L lengoaia:

- Lengoaia horretako hitzetan lehenengo zatiak honako bi egitura hauetako bat izan dezake:
  - a bakar batekin hasi eta gero b sinboloaren errepikapenez osatutako bloke bat. Bloke hori hutsa izan daiteke; edo
  - hasteko, b sinboloaren errepikapenez osatutako bloke ez-huts bat eta, bukatzeko, a bakar bat.
- Lengoaia horretako hitzetan bigarren zatia c sinboloaren errepikapenez osatutako bloke ez-huts bat izango da.

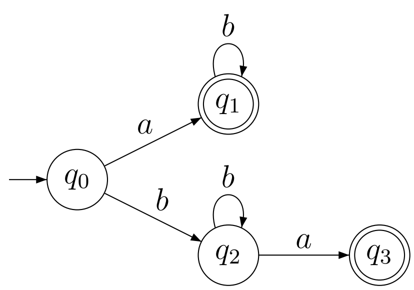
L lengoaia honela formaliza daiteke:

$$L = \{w | w \in A^* \wedge \exists u, v(u \in A^* \wedge v \in A^* \wedge |u|_b = |u| \wedge |v|_c = |v| \wedge (w = aucv \vee w = buacv))\}$$

Har ditzagun orain L_1 eta L_2 lengoaiak:

$$L_1 = \{w | w \in A^* \wedge \exists u(u \in A^* \wedge |u|_b = |u| \wedge (w = au \vee w = bua))\}$$
$$L_2 = \{w | w \in A^* \wedge \exists v(v \in A^* \wedge |v|_c = |v| \wedge w = cv)\}$$

L lengoaia L_1L_2 gisa formula daiteke. Beraz, L-ri dagokion ε-AFED bat eraikitzeko, L_1 eta L_2 lengoaiei dagozkien bi ε-AFED erabil ditzakegu oinarritzat.



3.4.25 irudia. $L_1 = \{a\}\{b\}^* \cup \{b\}^+\{a\}$ lengoaiari dagokion E_1 ε-AFEDa.

3.4.25 irudian L_1 lengoaiari dagokion E_1 ε-AFEDaren trantsizio-diagrama dugu. E_1 honako erako boskote gisa formaliza daiteke:

$$(Q, A, \lambda_1, q_0, Y_1)$$

Boskote horretako osagaien definizioa honako hau da:

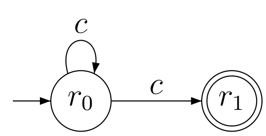
- $Q = \{q_0, q_1, q_2, q_3\}$

- $A = \{a, b, c\}$
- $\lambda_1 : Q \times (A \cup \{\varepsilon\}) \to 2^Q$ honako taula honen bidez definitutako trantsizio-funtzioa da:

λ_1	a	b	c	ε
q_0	$\{q_1\}$	$\{q_2\}$	$\varnothing$	$\varnothing$
q_1	$\varnothing$	$\{q_1\}$	$\varnothing$	$\varnothing$
q_2	$\{q_3\}$	$\{q_2\}$	$\varnothing$	$\varnothing$
q_3	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$

- q_0 hasierako egoera da.
- $Y_1 = \{q_1, q_3\}$.

E_1-en $\lambda_1(q_i, \varepsilon) = \varnothing$ betetzen da $q_i \in Q$ egoera denentzat eta, ondorioz, 3.4.25 irudiko trantsizio-diagraman ez dago ε etiketa duen trantsiziorik.



3.4.26 irudia. $L_2 = \{c\}^+$ lengoaiari dagokion E_2 ε-AFEDa.

3.4.26 irudian L_2 lengoaiari dagokion E_2 ε-AFEDa dugu. E_2 boskote gisa honela formaliza daiteke:

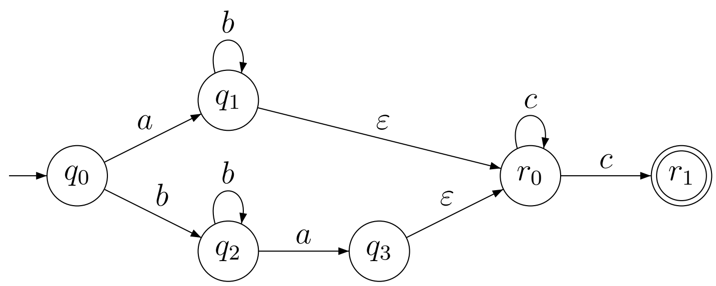
$$(R, A, \lambda_2, r_0, Y_2)$$

Boskote horretako osagaien definizioa honako hau da:

- $R = \{r_0, r_1\}$
- $A = \{a, b, c\}$
- $\lambda_2 : R \times (A \cup \{\varepsilon\}) \to 2^R$ honako taula honen bidez definitutako trantsizio-funtzioa da:

λ_2	a	b	c	ε
r_0	$\varnothing$	$\varnothing$	$\{r_0, r_1\}$	$\varnothing$
r_1	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$

- r_0 hasierako egoera da.
- $Y_2 = \{r_1\}$.



3.4.27 irudia. $L = (\{a\}\{b\}^* \cup \{b\}^+\{a\})\{c\}^+$ lengoaiari dagokion E ε-AFEDa.

E_2-n ere $\lambda_2(r_j, \varepsilon) = \varnothing$ betetzen da $r_j \in R$ egoera denentzat eta, ondorioz, 3.4.26 irudiko trantsizio-diagraman ez dago ε etiketa duen trantsiziorik.

3.4.27 irudian L_1L_2 lengoaiari dagokion E ε-AFEDaren trantsizio-diagrama dugu. E ε-AFEDa (S, A, λ, s_0, Y) boskote gisa formaliza daiteke eta boskote horren osagaiak honako hauek izango lirateke:

- $S = Q \cup R = \{q_0, q_1, q_2, q_3, r_0, r_1\}$.
- A alfabetoa E_1-eko eta E_2-ko alfabeto bera izango da.
- s_0 hasierako egoera q_0 izango da.
- Onarpen egoeren multzoari dagokionez, $Y = Y_2 = \{r_1\}$.
- Azkenik, λ trantsizio-funtzioa $S \times (A \cup \{\varepsilon\}) \to 2^S$ motakoa da eta honako taula honen bidez definituko genuke:

λ	a	b	c	ε
q_0	$\{q_1\}$	$\{q_2\}$	$\varnothing$	$\varnothing$
q_1	$\varnothing$	$\{q_1\}$	$\varnothing$	$\{r_0\}$
q_2	$\{q_3\}$	$\{q_2\}$	$\varnothing$	$\varnothing$
q_3	$\varnothing$	$\varnothing$	$\varnothing$	$\{r_0\}$
r_0	$\varnothing$	$\varnothing$	$\{r_0, r_1\}$	$\varnothing$
r_1	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$

E ε-AFEDan $\lambda(q_1, \varepsilon) = \{r_0\}$ da eta $\lambda(q_3, \varepsilon) = \{r_0\}$ da eta, ondorioz, 3.4.27 irudiko trantsizio-diagraman ε etiketa duten bi trantsizio ditugu.

3.4.2.2 ε-AFEDen diseinua lengoaien itxidura unibertsalaren kasurako

L lengoaia batentzat ε-AFED bat baldin badugu, L^* lengoaiari dagokion ε-AFED bat diseinatzeko nahikoa da onarpen egoera bakoitzetik hasierako egoerara ε trantsizio bat ipintzearekin. Horrela eraikitako ε-AFED berrian onarpen egoera bakarra hasierako egoera izango da. Kasu berezi gisa, abiapuntuko automatan hasierako egoera onarpen egoera baldin bada, ez da beharrezkoa hasierako egoeratik hasierako egoerara doan ε trantsizio bat ipintzea.

Formalki, demagun L_1 lengoaiaren itxidudura unibertsal gisa defini daitekeen L lengoaiarentzat ε-AFED bat diseinatu nahi dugula. Beraz, $L = (L_1)^*$. Demagun baita ere, L_1 lengoaiarentzat $E_1 = (Q, A, \lambda_1, q_0, Y_1)$ ε-AFEDa diseinatuta dugula. L lengoaiarentzat $E = (S, A, \lambda, s_0, Y)$ ε-AFED bat honela defini daiteke:

- $S = Q$.
- A alfabetoa E_1-eko alfabeto bera da.
- s_0 hasierako egoera q_0 da.
- Onarpen egoeren multzoari dagokionez, $Y = \{q_0\}$.
- λ trantsizio-funtzioa $S \times (A \cup \{\varepsilon\}) \rightarrow 2^S$ motakoa izango da eta honako hau beteko da:
  - $\lambda(q_i, \alpha) = \lambda_1(q_i, \alpha)$ izango da $q_i \in Q$ egoera denentzat eta $\alpha \in A$ sinbolo denentzat.
  - $\lambda(q_i, \varepsilon) = \lambda_1(q_i, \varepsilon)$ izango da $q_i \in \{q_0\} \cup (Q \setminus Y_1)$ egoera denentzat.
  - $\lambda(q_i, \varepsilon) = \lambda_1(q_i, \varepsilon) \cup \{q_0\}$ izango da $q_i \in Y_1 \setminus \{q_0\}$ egoera denentzat.

Orain bi adibide emango dira teknika hori nola aplikatzen den erakusteko.

3.4.2.2.1 a kopuru bikoitia duten hitzen lengoaiari dagokion ε-AFED baten diseinua

Har dezagun $A = \{a, b, c\}$ alfabetoa eta a kopuru bikoitia duten hitzen L lengoaia:

$$L = \{w | w \in A^* \wedge |w|_a \bmod 2 = 0\}$$

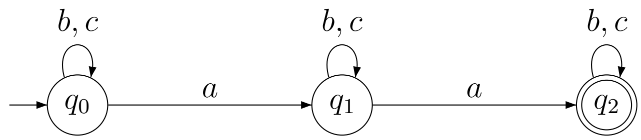
Har dezagun, baita ere, bi a eta nahi adina b eta c dituzten hitzen L_1 lengoaia:

$$L_1 = \{w | w \in A^* \wedge |w|_a = 2\}$$

L lengoaia L_1 lengoaiaren itxidura unibertsala da. Izan ere, a kopuru bikoitia edukitzea bi a zero aldiz edo gehiagotan edukitzea da.

3.4.28 irudian L_1 lengoaiarentzat E_1 ε-AFEDa dugu. ε-AFED hori honako boskote honen bidez formaliza daiteke:

$$(Q, A, \lambda_1, q_0, Y_1)$$



3.4.28 irudia. Bi a dituzten hitzen lengoaiari dagokion E_1 ε-AFEDa.

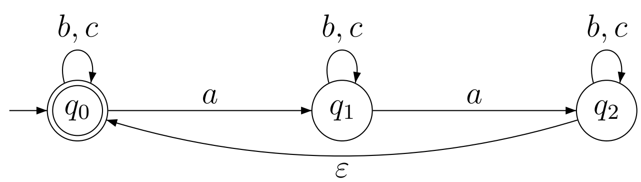
Boskote horretako osagaiak honela defini daitezke:

- $Q = \{q_0, q_1, q_2\}$
- $A = \{a, b, c\}$
- $\lambda_1 : Q \times (A \cup \{\varepsilon\}) \to 2^Q$ honako taula honen bidez definitutako trantsizio-funtzioa da:

λ_1	a	b	c	ε
q_0	$\{q_1\}$	$\{q_0\}$	$\{q_0\}$	$\varnothing$
q_1	$\{q_2\}$	$\{q_1\}$	$\{q_1\}$	$\varnothing$
q_2	$\varnothing$	$\{q_2\}$	$\{q_2\}$	$\varnothing$

- q_0 hasierako egoera da.
- $Y_1 = \{q_2\}$.

E_1-en $\lambda_1(q_i, \varepsilon) = \varnothing$ betetzen da $q_i \in Q$ egoera denentzat eta, ondorioz, 3.4.28 irudiko trantsizio-diagraman ez dago ε etiketa duen trantsiziorik.



3.4.29 irudia. a kopuru bikoitia duten hitzen lengoaiari dagokion E ε-AFEDa.

3.4.29 irudian $L = (L_1)^*$ lengoaiari dagokion E ε-AFEDa dugu. E ε-AFEDa (S, A, λ, s_0, Y) erako boskote gisa formaliza dezakegu eta boskote horretako osagaiak honako hauek izango dira:

- $S = Q = \{q_0, q_1, q_2\}$.
- A alfabetoa E_1-eko alfabeto bera izango da.

- s_0 hasierako egoera q_0 da.
- Onarpen egoeren multzoari dagokionez, $Y = \{q_0\}$.
- Bukatzeko, λ trantsizio-funtzioa $S \times (A \cup \{\varepsilon\}) \to 2^S$ motakoa izango da eta honako taula honen bidez defini daiteke:

λ	a	b	c	ε
q_0	$\{q_1\}$	$\{q_0\}$	$\{q_0\}$	$\varnothing$
q_1	$\{q_2\}$	$\{q_1\}$	$\{q_1\}$	$\varnothing$
q_2	$\varnothing$	$\{q_2\}$	$\{q_2\}$	$\{q_0\}$

E ε-AFEDan $\lambda(q_2, \varepsilon) = \{q_0\}$ da eta, ondorioz, 3.4.29 irudiko trantsizio-diagraman ε etiketa duten trantsizio bat dugu.

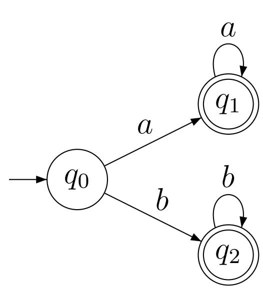
3.4.2.2.2 $\{a\}^+ \cup \{b\}^+$ lengoaiaren itxidura unibertsalari dagokion ε-AFED baten diseinua

Har dezagun $A = \{a, b, c\}$ alfabetoa eta $L = (\{a\}^+ \cup \{b\}^+)^*$ lengoaia.

Har dezagun baita ere $L_1 = \{a\}^+ \cup \{b\}^+$ lengoaia. L_1 lengoaia beste era honetara ere formaliza daiteke:

$$L_1 = \{w | w \in A^* \wedge |w| \geq 1 \wedge (|w| = |w|_a \vee |w| = |w|_b)\}$$

L lengoaia L_1 lengoaiaren itxidura unibertsala da. Beraz, $L = (L_1)^*$.



3.4.30 irudia. $L_1 = \{a\}^+ \cup \{b\}^+$ lengoaiari dagokion E_1 ε-AFEDa.

3.4.30 irudian L_1 lengoaiari dagokion E_1 ε-AFEDaren trantsizio-diagrama dugu. E_1 honako boskote gisa formaliza daiteke:

$$(Q, A, \lambda_1, q_0, Y_1)$$

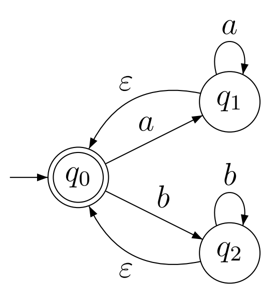
Boskote horretako osagaien definizioa honako hau da:

- $Q = \{q_0, q_1, q_2\}$
- $A = \{a, b, c\}$
- $\lambda_1 : Q \times (A \cup \{\varepsilon\}) \to 2^Q$ honako taula honen bidez definitutako trantsizio-funtzioa da:

λ_1	a	b	c	ε
q_0	$\{q_1\}$	$\{q_2\}$	$\varnothing$	$\varnothing$
q_1	$\{q_1\}$	$\varnothing$	$\varnothing$	$\varnothing$
q_2	$\varnothing$	$\{q_2\}$	$\varnothing$	$\varnothing$

- q_0 hasierako egoera da.
- $Y_1 = \{q_1, q_2\}$.

E_1-en $\lambda_1(q_i, \varepsilon) = \varnothing$ betetzen da $q_i \in Q$ egoera denentzat eta, ondorioz, 3.4.30 irudiko trantsizio-diagraman ez dago ε etiketa duen trantsiziorik.



3.4.31 irudia. $L = (L_1)^* = (\{a\}^+ \cup \{b\}^+)^*$ lengoaiari dagokion E ε-AFEDa.

3.4.31 irudian $L = (L_1)^*$ lengoaiari dagokion E ε-AFEDaren trantsizio-diagrama dugu. E ε-AFEDa (S, A, λ, s_0, Y) erako boskote gisa formaliza dezakegu eta boskote horretako osagaiak honako hauek izango dira:

- $S = Q = \{q_0, q_1, q_2\}$.
- A alfabetoa E_1-eko alfabeto bera izango da.
- s_0 hasierako egoera q_0 da.
- Onarpen egoeren multzoari dagokionez, $Y = \{q_0\}$.
- Bukatzeko, λ trantsizio-funtzioa $S \times (A \cup \{\varepsilon\}) \to 2^S$ motakoa izango da eta honako taula honen bidez defini daiteke:

λ	a	b	c	ε
q_0	$\{q_1\}$	$\{q_2\}$	$\varnothing$	$\varnothing$
q_1	$\{q_1\}$	$\varnothing$	$\varnothing$	$\{q_0\}$
q_2	$\varnothing$	$\{q_2\}$	$\varnothing$	$\{q_0\}$

E ε-AFEDan $\lambda(q_1, \varepsilon) = \{q_0\}$ da eta $\lambda(q_2, \varepsilon) = \{q_0\}$ da eta, ondorioz, 3.4.31 irudiko trantsizio-diagraman ε etiketa duten bi trantsizio ditugu.

3.4.2.3 ε-AFEDen diseinua lengoaien itxidura positiboaren kasurako

L lengoaia batentzat ε-AFED bat baldin badugu, L^+ lengoaiari dagokion ε-AFED bat diseinatzeko nahikoa da onarpen egoera bakoitzetik hasierako egoerara ε trantsizio bat ipintzearekin. Onarpen egoerak abiapuntuko automatan onarpen egoerak direnak izango dira. Kasu berezi gisa, abiapuntuko automatan hasierako egoera onarpen egoera baldin bada, ez da beharrezkoa hasierako egoeratik hasierako egoerara doan ε trantsizio bat ipintzea.

Formalki, demagun L_1 lengoaiaren itxidudura positibo gisa defini daitekeen L lengoaiarentzat ε-AFED bat diseinatu nahi dugula. Beraz, $L = (L_1)^+$. Demagun baita ere, L_1 lengoaiarentzat $E_1 = (Q, A, \lambda_1, q_0, Y_1)$ ε-AFEDa diseinatuta dugula. L lengoaiarentzat $E = (S, A, \lambda, s_0, Y)$ ε-AFED bat honela defini daiteke:

- $S = Q$.
- A alfabetoa E_1-eko alfabeto bera izango da.
- s_0 hasierako egoera q_0 izango da.
- Onarpen egoeren multzoari dagokionez, $Y = Y_1$.
- Bukatzeko, λ trantsizio-funtzioa $S \times (A \cup \{\varepsilon\}) \to 2^S$ motakoa izango da eta honako hau beteko da:
  - $\lambda(q_i, \alpha) = \lambda_1(q_i, \alpha)$ izango da $q_i \in Q$ egoera denentzat eta $\alpha \in A$ sinbolo denentzat.
  - $\lambda(q_i, \varepsilon) = \lambda_1(q_i, \varepsilon)$ izango da $q_i \in \{q_0\} \cup (Q \setminus Y_1)$ egoera denentzat.
  - $\lambda(q_i, \varepsilon) = \lambda_1(q_i, \varepsilon) \cup \{q_0\}$ izango da $q_i \in Y_1 \setminus \{q_0\}$ egoera denentzat.

Orain bi adibide emango dira teknika hori nola aplikatzen den erakusteko.

3.4.2.3.1 Zero izan gabe, a kopuru bikoitia duten hitzen lengoaiarentzat ε-AFED baten diseinua

Har ditzagun $A = \{a, b, c\}$ alfabetoa eta, zero izan gabe, a kopuru bikoitia duten hitzez osatutako L lengoaia. Lengoaia hori honela formaliza daiteke:

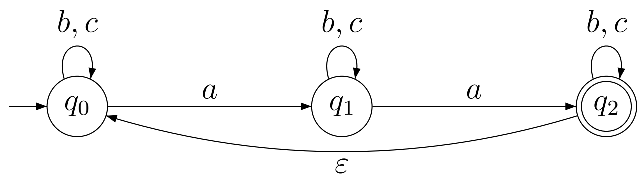
$$L = \{w | w \in A^* \wedge |w|_a \bmod 2 = 0 \wedge |w|_a \neq 0\}$$

Har dezagun baita ere zehazki bi a eta nahi adina b eta c dituzten hitzez osatutako L_1 lengoaia:

$$L_1 = \{w | w \in A^* \wedge |w|_a = 2\}$$

L lengoaia L_1 lengoaiaren itxidura positiboa da. Izan ere, zero izan gabe, a kopuru bikoitia edukitzea bi a behin edo gehiagotan edukitzea da.

3.4.28 irudian, L_1 lengoaiari dagokion E_1 ε-AFEDa dugu. E_1 automata hori $(Q, A, \lambda_1, q_0, Y_1)$ boskote gisa 3.4.2.2.1 atalean definitu da.



3.4.32 irudia. Zeroren desberdina den a kopuru bikoitia duten hitzen lengoaiarentzat E ε-AFEDa.

3.4.32 irudian $L = (L_1)^+$ lengoaiari dagokion E ε-AFEDaren trantsizio-diagrama dugu. E ε-AFEDa (S, A, λ, s_0, Y) erako boskote gisa formaliza dezakegu eta boskote horretako osagaiak honako hauek izango dira:

- $S = Q = \{q_0, q_1, q_2\}$.
- A alfabetoa E_1-eko alfabeto bera izango da.
- s_0 hasierako egoera q_0 da.
- Onarpen egoeren multzoari dagokionez, $Y = Y_1 = \{q_2\}$.
- λ trantsizio-funtzioa $S \times (A \cup \{\varepsilon\}) \rightarrow 2^S$ motakoa izango da eta honako taula honen bidez defini daiteke:

λ	a	b	c	ε
q_0	$\{q_1\}$	$\{q_0\}$	$\{q_0\}$	$\varnothing$
q_1	$\{q_2\}$	$\{q_1\}$	$\{q_1\}$	$\varnothing$
q_2	$\varnothing$	$\{q_2\}$	$\{q_2\}$	$\{q_0\}$

E ε-AFEDan $\lambda(q_2, \varepsilon) = \{q_0\}$ betetzen da eta, ondorioz, 3.4.32 irudian ε etiketa duen trantsizio bat dugu.

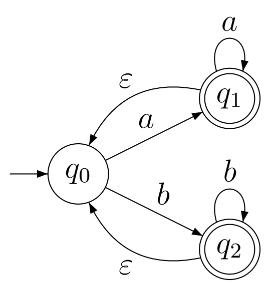
3.4.2.3.2 $\{a\}^+ \cup \{b\}^+$ lengoaiaren itxidura positiboari dagokion ε-AFED baten diseinua

Har ditzagun $A = \{a, b, c\}$ alfabetoa eta $L = (\{a\}^+ \cup \{b\}^+)^+$ lengoaia. Har dezagun baita ere $L_1 = \{a\}^+ \cup \{b\}^+$ lengoaia. L_1 lengoaia beste era honetara ere formaliza daiteke:

$$L_1 = \{w | w \in A^* \wedge |w| \geq 1 \wedge (|w| = |w|_a \vee |w| = |w|_b)\}$$

L lengoaia L_1 lengoaiaren itxidura positiboa da. Beraz, $L = (L_1)^+$.

3.4.30 irudian L_1 lengoaiari dagokion E_1 ε-AFEDa dugu. E_1 $(Q, A, \lambda_1, q_0, Y_1)$ boskote gisa 3.4.2.2.2 atalean formalizatu da.



3.4.33 irudia. $\{a\}^+ \cup \{b\}^+$ lengoaiaren itxidura positiboa den $(\{a\}^+ \cup \{b\}^+)^+$ lengoaiari dagokion E ε-AFEDa.

3.4.33 irudian L lengoaiari dagokion E ε-AFED dugu. E ε-AFEDa (S, A, λ, s_0, Y) erako boskote gisa formaliza dezakegu eta boskote horretako osagaiak honako hauek izango dira:

- $S = Q = \{q_0, q_1, q_2\}$.
- A alfabetoa E_1-eko alfabeto bera izango da.
- s_0 hasierako egoera q_0 da.
- Onarpen egoeren multzoari dagokionez, $Y = Y_1 = \{q_1, q_2\}$.
- λ trantsizio-funtzioa $S \times (A \cup \{\varepsilon\}) \to 2^S$ motakoa izango da eta honako taula honen bidez defini daiteke:

λ	a	b	c	ε
q_0	$\{q_1\}$	$\{q_2\}$	$\varnothing$	$\varnothing$
q_1	$\{q_1\}$	$\varnothing$	$\varnothing$	$\{q_0\}$
q_2	$\varnothing$	$\{q_2\}$	$\varnothing$	$\{q_0\}$

E ε-AFEDan $\lambda(q_1, \varepsilon) = \{q_0\}$ da eta $\lambda(q_2, \varepsilon) = \{q_0\}$ da eta, ondorioz, 3.4.33 irudiko trantsizio-diagraman ε etiketa duten bi trantsizio ditugu.

3.4.2.4 ε-AFEDen diseinua lengoaien berredurarentzat

L lengoaia batentzat ε-AFED bat baldin badugu, $n \geq 1$ izanda, L^n lengoaiari dagokion ε-AFED bat diseinatzeko kontuan hartu beharreko lehenengo gauza $L^n = \underbrace{LL\ldots L}_{n \text{ aldiz}}$ betetzen dela da. Ondorioz, ideia orokorra kateaketarako dugun teknika $n-1$ aldiz aplikatzea da. Hala ere, arazo bat agertzen da: ε-AFED bera n aldiz erabili behar da baina ε-AFED berriak ezin ditzake errepikatutako egoerak izan. Beraz, era batera edo bestera egoeren kopia desberdinak sortu behar dira. Hori egiteko era bat baino gehiago daude. Hemen aukera bat aurkeztuko da.

Formalki, demagun L_1 lengoaiaren berredura gisa defini daitekeen L lengoaiarentzat ε-AFED bat diseinatu nahi dugula. Beraz, $L = (L_1)^n$, $n \geq 1$ izanda. Demagun baita ere, L_1 lengoaiarentzat $E_1 = (Q, A, \lambda_1, q_0, Y_1)$ ε-AFEDa diseinatuta dugula. L lengoaiarentzat $E = (S, A, \lambda, s_0, Y)$ ε-AFED bat honela defini daiteke:

- $S = \{1, 2, \ldots, n\} \times Q$.
- A alfabetoa E_1-eko alfabeto bera izango da.
- s_0 hasierako egoera $(1, q_0)$ izango da.
- Onarpen egoeren multzoari dagokionez, $Y = \{n\} \times Y_1$.
- Bukatzeko, λ trantsizio-funtzioa $S \times (A \cup \{\varepsilon\}) \to 2^S$ motakoa izango da eta honako hau beteko da:
  - $\lambda((j, q_i), \alpha) = \{j\} \times \lambda_1(q_i, \alpha)$ izango da $j \in \{1, 2, \ldots, n\}$ zenbaki denentzat, $q_i \in Q$ egoera denentzat eta $\alpha \in A$ sinbolo denentzat.
  - $\lambda((j, q_i), \varepsilon) = \{j\} \times \lambda_1(q_i, \varepsilon)$ izango da $j \in \{1, 2, \ldots, n\}$ zenbaki denentzat eta $q_i \in Q \setminus Y_1$ egoera denentzat.
  - $\lambda((j, q_i), \varepsilon) = (\{j\} \times \lambda_1(q_i, \varepsilon)) \cup \{(j + 1, q_0)\}$ izango da $j \in \{1, 2, \ldots, n - 1\}$ zenbaki denentzat eta $q_i \in Y_1$ egoera denentzat.
  - $\lambda((n, q_i), \varepsilon) = \{n\} \times \lambda_1(q_i, \varepsilon)$ izango da $q_i \in Y_1$ egoera denentzat.

Orain adibide bat emango da teknika hori nola aplikatzen den erakusteko.

3.4.2.4.1 Sei a dituzten hitzen lengoaiari dagokion ε-AFED baten diseinua

Har ditzagun $A = \{a, b, c\}$ eta sei a dituzten hitzez osatutako L lengoaia.

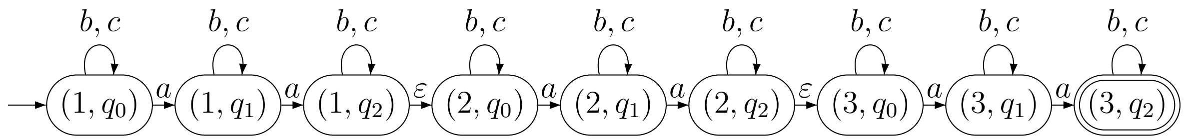
$$L = \{w | w \in A^* \wedge |w|_a = 6\}$$

Har dezagun baita ere zehazki bi a eta nahi adina b eta c dituzten hitzez osatutako L_1 lengoaia:

$$L_1 = \{w | w \in A^* \wedge |w|_a = 2\}$$

L lengoaia $(L_1)^3$ da. Izan ere, sei a edukitzea bi a hiru aldiz edukitzea da. 3.4.28 irudian L_1 lengoaiari dagokion E_1 ε-AFEDa dugu. E_1 $(Q, A, \lambda_1, q_0, Y_1)$ boskote gisa 3.4.2.2.1 atalean definitu da.

3.4.34 irudian $L = (L_1)^3$ lengoaiari dagokion E ε-AFEDa dugu. E ε-AFEDa (S, A, λ, s_0, Y) erako boskote bat gisa formaliza dezakegu eta boskote horretako osagaiak honako hauek izango dira:



3.4.34 irudia. Sei a dituzten hitzen L lengoaiarentzat E ε-AFEDa.

- S egoeren multzoa honako hau da:

$$\begin{aligned} S &= \{1,2,3\} \times Q \\ &= \{(1,q_0),(1,q_1),(1,q_2),(2,q_0),(2,q_1),(2,q_2),(3,q_0),(3,q_1),(3,q_2)\} \end{aligned}$$

  Beraz, Q-ko egoera bakoitzeko hiru kopia edukiko dira. Horretarako, bikoteak sortuko dira eta bikote bakoitzeko lehenengo elementuak kopia zenbakia adieraziko du.

- A alfabetoa E_1-eko alfabeto bera izango da.

- s_0 hasierako egoera $(1, q_0)$ da.

- Onarpen egoeren multzoari dagokionez, $Y = \{3\} \times Y_1 = \{(3, q_2)\}$.

- λ trantsizio-funtzioa $S \times (A \cup \{\varepsilon\}) \to 2^S$ motakoa izango da eta honako taula honen bidez defini daiteke:

λ	a	b	c	ε
$(1,q_0)$	$\{(1,q_1)\}$	$\{(1,q_0)\}$	$\{(1,q_0)\}$	$\varnothing$
$(1,q_1)$	$\{(1,q_2)\}$	$\{(1,q_1)\}$	$\{(1,q_1)\}$	$\varnothing$
$(1,q_2)$	$\varnothing$	$\{(1,q_2)\}$	$\{(1,q_2)\}$	$\{(2,q_0)\}$
$(2,q_0)$	$\{(2,q_1)\}$	$\{(2,q_0)\}$	$\{(2,q_0)\}$	$\varnothing$
$(2,q_1)$	$\{(2,q_2)\}$	$\{(2,q_1)\}$	$\{(2,q_1)\}$	$\varnothing$
$(2,q_2)$	$\varnothing$	$\{(2,q_2)\}$	$\{(2,q_2)\}$	$\{(3,q_0)\}$
$(3,q_0)$	$\{(3,q_1)\}$	$\{(3,q_0)\}$	$\{(3,q_0)\}$	$\varnothing$
$(3,q_1)$	$\{(3,q_2)\}$	$\{(3,q_1)\}$	$\{(3,q_1)\}$	$\varnothing$
$(3,q_2)$	$\varnothing$	$\{(3,q_2)\}$	$\{(3,q_2)\}$	$\varnothing$

3.4.2.5 ε-AFEDaren diseinua lengoaia baten alderantzizkoarentzat

L lengoaia batentzat ε-AFED bat baldin badugu, L^R lengoaiari dagokion ε-AFED bat diseinatzeko honako puntuak kontsideratzea nahikoa da:

- Abiapuntuko ε-AFEDaren egoera denak mantenduko dira.

- Hasierako egoera izango den beste egoera bat ipiniko da. Egoera horretatik ε trantsizio bat ipini beharko da abiapuntuko ε-AFEDan onarpen egoera den egoera bakoitzera.

- ε-AFED berriko onarpen egoera bakarra abiapuntuko ε-AFEDko hasierako egoera izango da.
- Abiapuntuko ε-AFEDko begizta denak mantenduko dira.
- Begiztak ez diren trantsizioen kasuan, aurkako zentzuan ipini behar dira.

Formalki, demagun L_1 lengoaiaren alderantzizkoa den L lengoaiarentzat ε-AFED bat diseinatu nahi dugula. Beraz, $L = (L_1)^R$. Demagun baita ere, L_1 lengoaiarentzat $E_1 = (Q, A, \lambda_1, q_0, Y_1)$ ε-AFEDa diseinatuta dugula. L lengoaiarentzat $E = (S, A, \lambda, s_0, Y)$ ε-AFED bat honela defini daiteke:

- $S = Q \cup \{z_0\}$. Hor, z_0 egoera Q-koa ez den egoera bat izango da.
- A alfabetoa E_1-eko alfabeto bera izango da.
- s_0 hasierako egoera z_0 izango da.
- Onarpen egoeren multzoari dagokionez, $Y = \{q_0\}$.
- Bukatzeko, λ trantsizio-funtzioa $S \times (A \cup \{\varepsilon\}) \to 2^S$ motakoa izango da eta honako hau beteko da:
  - $\lambda(z_0, \alpha) = \varnothing$ izango da $\alpha \in A$ sinbolo denentzat.
  - $\lambda(z_0, \varepsilon) = Y_1$.
  - $\lambda(q_i, \alpha) = \{q_j \mid q_i \in \lambda_1(q_j, \alpha)\}$ izango da $q_i \in Q$ egoera denentzat eta $\alpha \in A$ sinbolo denentzat.
  - $\lambda(q_i, \varepsilon) = \{q_j \mid q_i \in \lambda_1(q_j, \varepsilon)\}$ izango da $q_i \in Q$ egoera denentzat.

Orain adibide bat emango da teknika hori nola aplikatzen den erakusteko.

3.4.2.5.1 a-rekin bukatu edo b-rekin hasten diren hitzen lengoaiari dagokion ε-AFEDa

Har ditzagun $A = \{a, b, c\}$ alfabetoa eta a-rekin bukatu edo b-rekin hasten diren hitzen L lengoaia. L lengoaia hori honela formaliza daiteke:

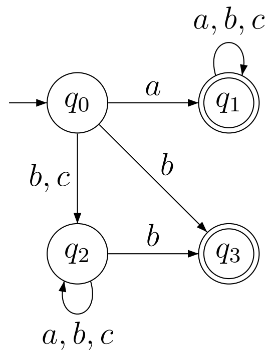
$$L = \{w | w \in A^* \wedge \exists u(u \in A^* \wedge (w = ua \vee w = bu))\}$$

Har dezagun baita ere a-rekin hasi edo b-rekin bukatzen diren hitzez osatutako L_1 lengoaia. L_1 honela formaliza daiteke:

$$L_1 = \{w | w \in A^* \wedge \exists u(u \in A^* \wedge (w = au \vee w = ub))\}$$

Beraz, $L = (L_1)^R$.

3.4.35 irudian L_1 lengoaiari dagokion E_1 ε-AFEDa dugu. E_1 ε-AFEDa boskote gisa honela formaliza daiteke:



3.4.35 irudia. a-rekin hasi edo b-rekin bukatzen diren hitzen L_1 lengoaiari dagokion E_1 ε-AFEDa.

$$(Q, A, \lambda_1, q_0, Y_1)$$

Boskote horretako osagaien definizioa honako hau da:

- $Q = \{q_0, q_1, q_2, q_3\}$
- $A = \{a, b, c\}$
- $\lambda_1 : Q \times (A \cup \{\varepsilon\}) \rightarrow 2^Q$ honako taula honen bidez definitutako trantsizio-funtzioa da:

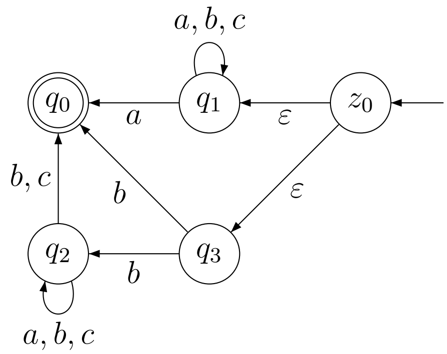
λ_1	a	b	c	ε
q_0	$\{q_1\}$	$\{q_2, q_3\}$	$\{q_2\}$	$\varnothing$
q_1	$\{q_1\}$	$\{q_1\}$	$\{q_1\}$	$\varnothing$
q_2	$\{q_2\}$	$\{q_2, q_3\}$	$\{q_2\}$	$\varnothing$
q_3	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$

- q_0 hasierako egoera da.
- $Y_1 = \{q_1, q_3\}$.

E_1-en $\lambda_1(q_i, \varepsilon) = \varnothing$ betetzen da $q_i \in Q$ egoera denentzat eta, ondorioz, 3.4.35 irudiko trantsizio-diagraman ez dago ε etiketa duen trantsiziorik.

3.4.36 irudian $(L_1)^R$ lengoaiari dagokion ε-AFEDa dugu. Egoera berria z_0 da. E ε-AFEDa (S, A, λ, s_0, Y) erako boskote gisa formaliza dezakegu eta boskote horretako osagaiak honako hauek izango dira:

- $S = Q \cup \{z_0\} = \{q_0, q_1, q_2, q_3, z_0\}$.
- A alfabetoa E_1-eko alfabeto bera izango da.



3.4.36 irudia. a-rekin bukatu edo b-rekin hasten diren hitzen L lengoaiari dagokion E ε-AFEDa.

- s_0 hasierako egoera z_0 da.
- Onarpen egoeren multzoari dagokionez, $Y = \{q_0\}$.
- λ trantsizio-funtzioa $S \times (A \cup \{\varepsilon\}) \to 2^S$ motakoa izango da eta honako taula honen bidez defini daiteke:

λ	a	b	c	ε
q_0	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
q_1	$\{q_0, q_1\}$	$\{q_1\}$	$\{q_1\}$	$\varnothing$
q_2	$\{q_2\}$	$\{q_0, q_2\}$	$\{q_0, q_2\}$	$\varnothing$
q_3	$\varnothing$	$\{q_0, q_2\}$	$\varnothing$	$\varnothing$
z_0	$\varnothing$	$\varnothing$	$\varnothing$	$\{q_1, q_3\}$

E ε-AFEDan $\lambda(z_0, \varepsilon) = \{q_1, q_3\}$ da eta, ondorioz, 3.4.36 irudiko trantsizio-diagraman ε etiketa duten bi trantsizio ditugu.

3.4.2.6 ε-AFEDen diseinua propietateen ukapenentzat: lengoaia osagarria

3.2.2.3 atalean azaldutakoa gogoratuz, ψ propietate baten ukapena bete behar duten hitzez (hau da, $\neg\psi$ bete behar duten hitzez) osatutako L lengoaia baten AFDa eraikitzeko aukera bat honako hau da:

- Hasteko, lengoaia horren osagarriarentzat, hau da, ψ propietatea betetzen duten hitzen $\overline{L}$ lengoaiarentzat D AFD bat eraiki.
- Gero, D AFDan zirkulu bikoitza duten egoerei zirkulu bikoitza kenduz eta zirkulu bikoitzik ez duten egoerei zirkulu bikoitza ipiniz D' AFD berri bat eraiki.

Baina teknika horrek ez du funtzionatzen AFEDentzat (ikusi 3.3.3.3 atala) eta ε-AFEDentzat.

3.4.2.7 ε-AFEDen diseinua propietateen konjuntzioa dugunerako (lengoaien arteko ebakidura)

Propietateen konjuntzioa bete behar duten hitzez osatutako lengoaia bati dagokion AFDa eraikitzeko aurkeztu den teknikak ε-AFEDak diseinatzeko ere balio du baina ε trantsizioen ezaugarri bereziak kontuan hartu beharko dira. Beraz, $L = L_1 \cap L_2$ lengoaia batentzat ε-AFED bat eraiki nahi bada eta dagoeneko L_1 eta L_2 lengoaientzat $E_1 = (Q, A, \lambda_1, q_0, Y_1)$ eta $E_2 = (R, A, \lambda_2, r_0, Y_2)$ ε-AFEDak diseinatuta baldin baditugu, L-rentzat era sistematikoan diseina dezakegu ε-AFED bat:

- Egoeren multzoa E_1-eko eta E_2-ko egoerekin osa daitezkeen bikote denez eratuta egongo da.
- Hasierako egoera E_1-eko eta E_2-ko hasierako egoerek osatzen duten bikotea izango da.
- Onarpen egoerak E_1-eko eta E_2-ko onarpen egoerez eratutako bikoteak izango dira.
- Trantsizioei dagokienez, E ε-AFEDan (q_i, r_j) egoera batetik (q_k, r_ℓ) beste egoera batera A alfabetoko α sinbolo batekin trantsizioa egongo da baldin eta soilik baldin q_i egoeratik q_k egoerara α sinboloarekin trantsizio bat baldin badago E_1-en eta r_j egoeratik r_ℓ egoerara α sinboloarekin trantsizio bat baldin badago E_2-n. ε trantsizioak bereziak dira. E_1-en q_i egoeratik q_k egoerara ε sinboloarekin trantsizio bat baldin badago, orduan, (q_i, r_j) egoeratik (q_k, r_j) egoerara ε trantsizio bat egongo da $r_j \in R$ egoera denentzat. Era berean, E_2-n r_j egoeratik r_ℓ egoerara ε trantsizio bat baldin badago, orduan, (q_i, r_j) egoeratik (q_i, r_ℓ) egoerara ε trantsizio bat egongo da $q_i \in Q$ egoera denentzat. E_1-en q_i egoeratik q_k egoerara ε sinboloarekin trantsizio bat baldin badago eta E_2-n r_j egoeratik r_ℓ egoerara ε trantsizio bat baldin badago, orduan, (q_i, r_j) egoeratik (q_k, r_ℓ) egoerara ε trantsizio bat egongo da.

Formalki, demagun L_1 eta L_2 lengoaien ebakidura gisa defini daitekeen L lengoaiarentzat ε-AFED bat diseinatu nahi dugula. Beraz, $L = L_1 \cap L_2$. Demagun baita ere, L_1 eta L_2 lengoaientzat $E_1 = (Q, A, \lambda_1, q_0, Y_1)$ eta $E_2 = (R, A, \lambda_2, r_0, Y_2)$ ε-AFEDak diseinatuta ditugula. L lengoaiarentzat $E = (S, A, \lambda, s_0, Y)$ ε-AFED bat honela defini daiteke:

- $S = Q \times R$.
- A alfabetoa E_1-eko eta E_2-ko alfabeto bera izango da.
- s_0 hasierako egoera (q_0, r_0) izango da.
- Onarpen egoeren multzoari dagokionez, $Y = Y_1 \times Y_2$.
- Bukatzeko, λ trantsizio-funtzioa $S \times (A \cup \{\varepsilon\}) \to 2^S$ motakoa izango da eta honako hau beteko da:
  - $\lambda((q_i, r_j), \alpha) = \lambda_1(q_i, \alpha) \times \lambda_2(r_j, \alpha)$ izango da $q_i \in Q$ egoera denentzat, $r_j \in R$ egoera denentzat eta $\alpha \in A$ sinbolo denentzat.

– $\lambda((q_i, r_j), \varepsilon) = (\lambda_1(q_i, \varepsilon) \times \{r_j\}) \cup (\{q_i\} \times \lambda_2(r_j, \varepsilon)) \cup (\lambda_1(q_i, \varepsilon) \times \lambda_2(r_j, \varepsilon))$ izango da $q_i \in Q$ egoera denentzat eta $r_j \in R$ egoera denentzat.

Orain bi adibide emango dira teknika hori nola aplikatzen den erakusteko.

3.4.2.7.1 a kopuru bikoitia eta b kopuru bikoitia duten hitzen lengoaiarentzat ε-AFED baten diseinua.

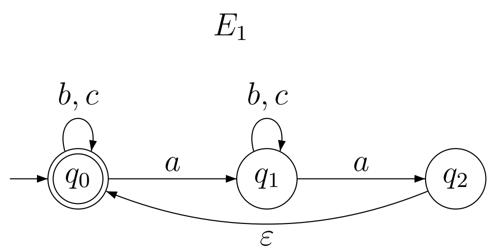
Har dezagun $A = \{a, b, c\}$ alfabetoa. Demagun a kopuru bikoitia eta b kopuru bikoitia duten hitzen lengoaiarentzat ε-AFED bat diseinatu nahi dugula. L lengoaia formalki honela adieraz daiteke:

$$L = \{w \mid w \in A^* \wedge (|w|_a \bmod 2 = 0) \wedge (|w|_b \bmod 2 = 0)\}$$

Hor garbi gelditzen da L lengoaiako hitzek bi propietateren konjuntzioa den baldintza bat bete behar dutela. Har ditzagun L_1 eta L_2 lengoaiak:

$$L_1 = \{w \mid w \in A^* \wedge (|w|_a \bmod 2 = 0)\}$$
$$L_2 = \{w \mid w \in A^* \wedge (|w|_b \bmod 2 = 0)\}$$

L_1 lengoaiako hitzek a kopuru bikoitia dute eta L_2 lengoaiako hitzek b kopuru bikoitia dute. L lengoaia $L_1 \cap L_2$ eran defini daiteke. Hasteko, L_1 lengoaiari dagokion E_1 ε-AFED bat eta L_2 lengoaiari dagokion E_2 ε-AFED bat diseina ditzakegu.



3.4.37 irudia. a kopuru bikoitia duten hitzen lengoaiari dagokion E_1 ε-AFEDa.

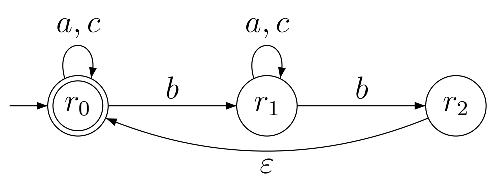
3.4.37 irudian E_1 ε-AFEDaren trantsizio-diagrama erakusten da. E_1 ε-AFEDa $(Q, A, \lambda_1, q_0, Y_1)$ boskote gisa formaliza daiteke. Boskote horretako osagaiak honako era honetara definituta daude:

- $Q = \{q_0, q_1, q_2\}$
- $A = \{a, b, c\}$
- $\lambda_1 : Q \times (A \cup \{\varepsilon\}) \to 2^Q$ honako taula honen bidez definitutako trantsizio-funtzioa da:

λ_1	a	b	c	ε
q_0	$\{q_1\}$	$\{q_0\}$	$\{q_0\}$	$\varnothing$
q_1	$\{q_2\}$	$\{q_1\}$	$\{q_1\}$	$\varnothing$
q_2	$\varnothing$	$\varnothing$	$\varnothing$	$\{q_0\}$

- q_0 hasierako egoera da.
- $Y_1 = \{q_0\}$.

E_2



3.4.38 irudia. b kopuru bikoitia duten hitzen lengoaiari dagokion E_2 ε-AFEDa.

3.4.38 irudian E_2 ε-AFEDaren trantsizio-diagrama dugu. E_2 honako boskote honen bidez formaliza daiteke:

$$(R, A, \lambda_2, r_0, Y_2)$$

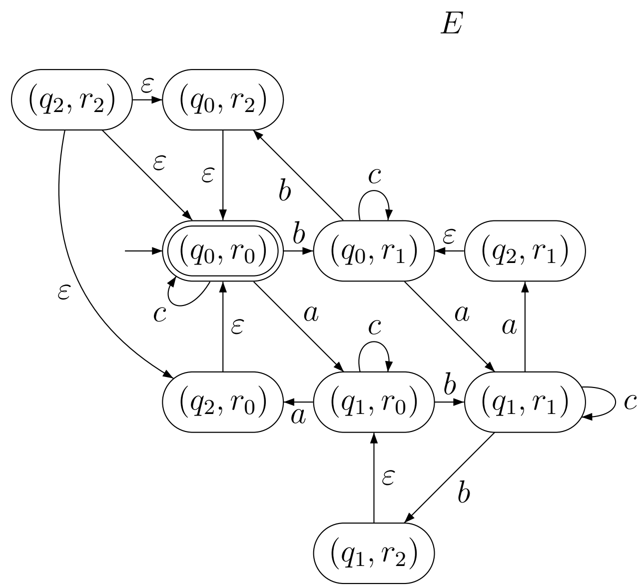
Boskote horretako osagaiak honela defini daitezke:

- $R = \{r_0, r_1, r_2\}$
- $A = \{a, b, c\}$
- $\lambda_2 : R \times (A \cup \{\varepsilon\}) \to 2^R$ honako taula honen bidez definitutako trantsizio-funtzioa da:

λ_2	a	b	c	ε
r_0	$\{r_0\}$	$\{r_1\}$	$\{r_0\}$	$\varnothing$
r_1	$\{r_1\}$	$\{r_2\}$	$\{r_1\}$	$\varnothing$
r_2	$\varnothing$	$\varnothing$	$\varnothing$	$\{q_0\}$

- r_0 hasierako egoera da.
- $Y_2 = \{r_0\}$.

3.4.39 irudian, a kopuru bikoitia eta b kopuru bikoitia duten hitzen lengoaiari dagokion E ε-AFEDaren trantsizio-diagrama dugu. E ε-AFED hori E_1 eta E_2 abiapuntutzat hartuz eta konjuntzioaren teknika aplikatuz lortu da. E ε-AFEDa (S, A, λ, s_0, Y) boskote gisa formaliza daiteke. Boskote horretako osagaiak honela definituta daude:



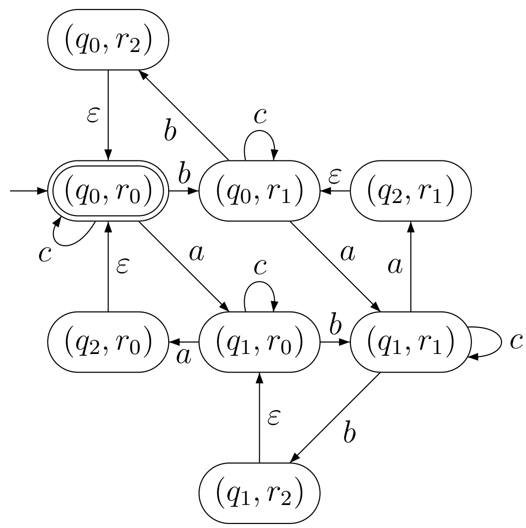
3.4.39 irudia. a kopuru bikoitia eta b kopuru bikoitia duten hitzen lengoaiari dagokion E ε-AFEDa.

- $S = Q \times R = \{(q_0, r_0), (q_0, r_1), (q_0, r_2), (q_1, r_0), (q_1, r_1), (q_1, r_2), (q_2, r_0), (q_2, r_1), (q_2, r_2)\}$.
- A alfabetoa E_1 eta E_2 ε-AFEDetako alfabeto bera da.
- s_0 hasierako egoera (q_0, r_0) da.
- Onarpen egoeren multzoari dagokionez, $Y = Y_1 \times Y_2 = \{(q_0, r_0)\}$.
- λ trantsizio-funtzioa $S \times (A \cup \{\varepsilon\}) \to 2^S$ motakoa izango da eta honako taula honen bidez defini daiteke:

λ	a	b	c	ε
(q_0, r_0)	$\{(q_1, r_0)\}$	$\{(q_0, r_1)\}$	$\{(q_0, r_0)\}$	$\varnothing$
(q_0, r_1)	$\{(q_1, r_1)\}$	$\{(q_0, r_2)\}$	$\{(q_0, r_1)\}$	$\varnothing$
(q_0, r_2)	$\varnothing$	$\varnothing$	$\varnothing$	$\{(q_0, r_0)\}$
(q_1, r_0)	$\{(q_2, r_0)\}$	$\{(q_1, r_1)\}$	$\{(q_1, r_0)\}$	$\varnothing$
(q_1, r_1)	$\{(q_2, r_1)\}$	$\{(q_1, r_2)\}$	$\{(q_1, r_1)\}$	$\varnothing$
(q_1, r_2)	$\varnothing$	$\varnothing$	$\varnothing$	$\{(q_1, r_0)\}$
(q_2, r_0)	$\varnothing$	$\varnothing$	$\varnothing$	$\{(q_0, r_0)\}$
(q_2, r_1)	$\varnothing$	$\varnothing$	$\varnothing$	$\{(q_0, r_1)\}$
(q_2, r_2)	$\varnothing$	$\varnothing$	$\varnothing$	$\{(q_0, r_0), (q_0, r_2), (q_2, r_0)\}$

(q_2, r_2) egoera iristezina da (q_0, r_0) egoeratik eta, ondorioz, ken daiteke. 3.4.40 irudian, egoera hori kenduz gelditzen den ε-AFEDaren trantsizio-diagrama dugu.

E



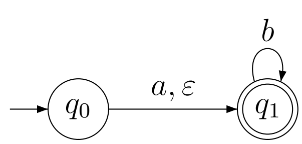
3.4.40 irudia. a kopuru bikoitia eta b kopuru bikoitia duten hitzen lengoaiari dagokion E ε-AFEDa (iristezinak diren egoerarik gabe).

3.4.2.7.2 $(\{a\}\{b\}^*) \cup \{b\}^*$ eta $(\{b\}\{b\}^*) \cup (\{c\}\{c\}^*)$ lengoaien arteko ebakidurarentzat ε-AFED baten diseinua.

Har dezagun $A = \{a, b, c\}$ lengoaia. Har ditzagun baita ere $L_1 = (\{a\}\{b\}^*) \cup \{b\}^*$ eta $L_2 = (\{b\}\{b\}^*) \cup (\{c\}\{c\}^*)$ lengoaiak. $L_1 \cap L_2$ lengoaiarentzat ε-AFED bat diseinatu nahi da.

Hasteko, L_1 lengoaiarentzat E_1 ε-AFED bat eta L_2 lengoaiarentzat E_2 ε-AFED bat diseinatuko dira.

E_1



3.4.41 irudia. $(\{a\}\{b\}^*) \cup \{b\}^*$ lengoaiari dagokion E_1 ε-AFEDa.

3.4.41 irudian L_1 lengoaiari dagokion E_1 ε-AFED baten trantsizio-diagrama dugu. E_1-i dagokion boskotea honela formaliza daiteke:

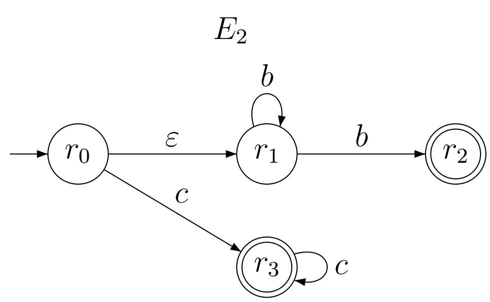
$$(Q, A, \lambda_1, q_0, Y_1)$$

Boskote horretako osagaiak honako era honetara definituko genituzke:

- $Q = \{q_0, q_1\}$
- $A = \{a, b, c\}$
- $\lambda_1 : Q \times (A \cup \{\varepsilon\}) \to 2^Q$ honako taula honen bidez definitutako trantsizio-funtzioa da:

λ_1	a	b	c	ε
q_0	$\{q_1\}$	$\varnothing$	$\varnothing$	$\{q_1\}$
q_1	$\varnothing$	$\{q_1\}$	$\varnothing$	$\varnothing$

- q_0 hasierako egoera da.
- $Y_1 = \{q_1\}$.



3.4.42 irudia. $(\{b\}\{b\}^*) \cup (\{c\}\{c\}^*)$ lengoaiari dagokion E_2 ε-AFEDa.

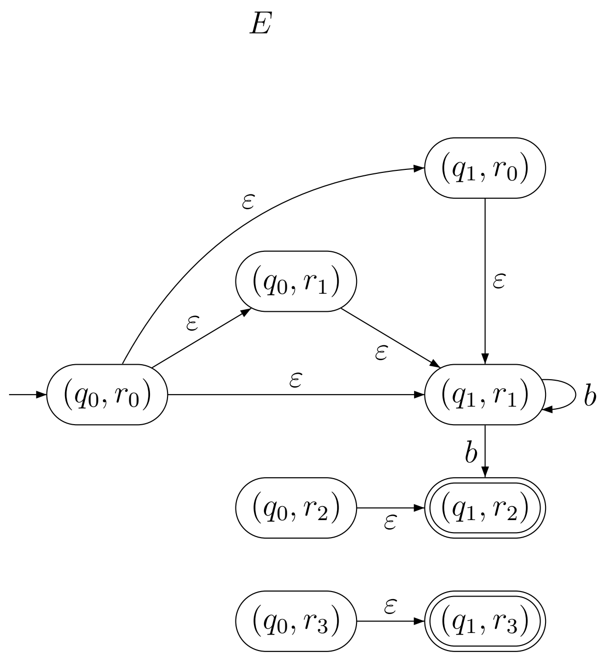
3.4.42 irudian E_2 ε-AFEDari dagokion trantsizio-diagrama erakusten da. E_2 boskote gisa honela formaliza daiteke:

$$(R, A, \lambda_2, r_0, Y_2)$$

Boskote horretako osagaiak honako era honetara definituko genituzke:

- $R = \{r_0, r_1, r_2, r_3\}$
- $A = \{a, b, c\}$
- $\lambda_2 : R \times (A \cup \{\varepsilon\}) \to 2^R$ honako taula honen bidez definitutako trantsizio-funtzioa da:

λ_2	a	b	c	ε
r_0	$\varnothing$	$\varnothing$	$\{r_3\}$	$\{r1\}$
r_1	$\varnothing$	$\{q_1, r_2\}$	$\varnothing$	$\varnothing$
r_2	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
r_3	$\varnothing$	$\varnothing$	$\{r_3\}$	$\varnothing$



3.4.43 irudia. $((\{a\}\{b\}^*) \cup \{b\}^*) \cap ((\{b\}\{b\}^*) \cup (\{c\}\{c\}^*))$ lengoaiari dagokion E ε-AFEDa.

- r_0 hasierako egoera da.
- $Y_2 = \{r_2, r_3\}$.

3.4.43 irudian

$$L_1 \cap L_2 = ((\{a\}\{b\}^*) \cup \{b\}^*) \cap ((\{b\}\{b\}^*) \cup (\{c\}\{c\}^*))$$

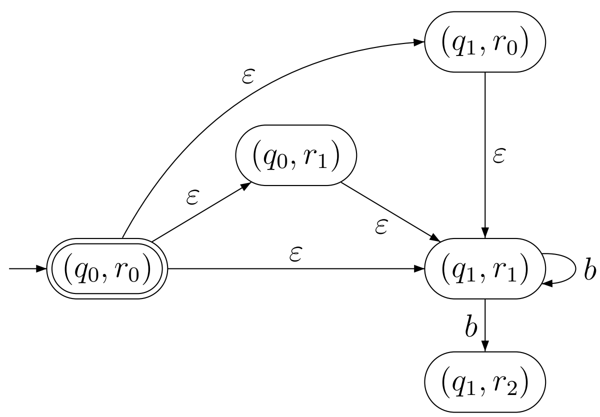
lengoaiari dagokion E ε-AFEDaren trantsizio-diagrama dugu. E ε-AFEDa E_1 eta E_2 ε-AFEDak abiapuntutzat hartuz eta konjuntzioaren teknika aplikatuz eraiki da. L-rentzat eraikitako E ε-AFEDa (S, A, λ, s_0, Y) boskote gisa formaliza daiteke. Boskote horretako osagaien definizioa honako hau da:

- $S = Q \times R = \{(q_0, r_0), (q_0, r_1), (q_0, r_2), (q_0, r_3), (q_1, r_0), (q_1, r_1), (q_1, r_2), (q_1, r_3)\}$.
- A alfabetoa E_1-eko eta E_2-ko alfabeto bera da.
- s_0 hasierako egoera (q_0, r_0) da.
- Onarpen egoeren multzoari dagokionez, $Y = Y_1 \times Y_2 = \{(q_1, r_2), (q_1, r_3)\}$.
- λ trantsizio-funtzioa $S \times (A \cup \{\varepsilon\}) \rightarrow 2^S$ motakoa izango da eta honako taula honen bidez defini daiteke:

λ	a	b	c	ε
(q_0, r_0)	$\varnothing$	$\varnothing$	$\varnothing$	$\{(q_0, r_1), (q_1, r_0), (q_1, r_1)\}$
(q_0, r_1)	$\varnothing$	$\varnothing$	$\varnothing$	$\{(q_1, r_1)\}$
(q_0, r_2)	$\varnothing$	$\varnothing$	$\varnothing$	$\{(q_1, r_2)\}$
(q_0, r_3)	$\varnothing$	$\varnothing$	$\varnothing$	$\{(q_1, r_3)\}$
(q_1, r_0)	$\varnothing$	$\varnothing$	$\varnothing$	$\{(q_1, r_1)\}$
(q_1, r_1)	$\varnothing$	$\{(q_1, r_1), (q_1, r_2)\}$	$\varnothing$	$\varnothing$
(q_1, r_2)	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
(q_1, r_3)	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$

E ε-AFED horretan (q_0, r_2), (q_0, r_3) eta (q_1, r_3) iristezinak dira (q_0, r_0) egoeratik eta, ondorioz, ken daitezke.

E



3.4.44 irudia. $((\{a\}\{b\}^*) \cup \{b\}^*) \cap ((\{b\}\{b\}^*) \cup (\{c\}\{c\}^*))$ lengoaiari dagokion E ε-AFEDa iristezinak diren egoerarik gabe.

3.4.44 irudian

$$L_1 \cap L_2 = ((\{a\}\{b\}^*) \cup \{b\}^*) \cap ((\{b\}\{b\}^*) \cup (\{c\}\{c\}^*))$$

lengoaiari dagokion E ε-AFEDa erakusten da, baina hasierako egoeratik iristezinak diren egoerak kenduta.

$L_1 \cap L_2$ lengoaia b sinboloaren errepikapenez eratutako hitz ez-hutsez osatutako lengoaia da.

3.4.2.8 ε-AFEDen diseinua propietateen disjuntzioa dugunerako (lengoaien bildura)

Aurreko atalean konjuntzioarentzat azaldu den teknikak AFDentzat, AFEDentzat eta ε-AFEDentzat balio du. AFDen kasuan bai ukapenarentzat (ikus 3.2.2.3 atala) eta bai konjuntzioarentzat (ikus 3.2.2.4 atala) teknika sistematikoa dugunez, konjuntzioaren teknika horrek berak disjuntzioarentzat ere balio du (ikus 3.2.2.5 atala) aldaketa txiki batekin. Hori horrela da L_1 eta L_2 lengoaien bildura ukapena eta konjuntzioa erabiliz defini daitekeelako: $L_1 \cup L_2 = \overline{\overline{L_1} \cap \overline{L_2}}$. Baina AFEDen eta ε-AFEDen kasuan ukapenarentzat ez dago teknika sistematikorik eta, ondorioz, AFDen kasurako disjuntzioarentzat aurkeztu den teknikak ez du balio AFEDen eta ε-AFEDen kasuan. Hala ere, disjuntzioa dugunean AFEDak sistematikoki diseinatzeko beste teknika bat aurkeztu da 3.4.2.8 atalean. ε-AFEDen kasuan ere disjuntzioa edo lengoaien bildura dugunean metodo sistematiko bat erabil dezakegu. Nahikoa da hasierako egoera izango den egoera berri bat ipintzea eta egoera horretatik abiapuntuko automatetako (elkartu nahi diren automatetako) hasierako egoeretara ε trantsizioak ipintzea.

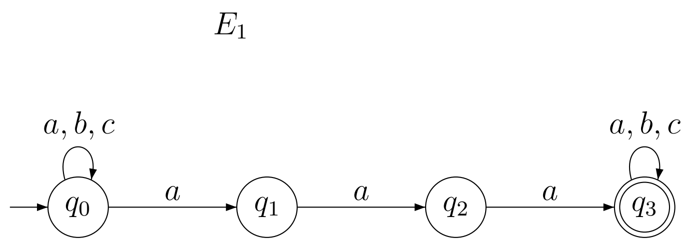
Formalki, demagun L_1 eta L_2 lengoaien bildura gisa defini daitekeen L lengoaiarentzat ε-AFED bat diseinatu nahi dugula. Beraz, $L = L_1 \cup L_2$. Demagun baita ere, L_1 eta L_2 lengoaientzat $E_1 = (Q, A, \lambda_1, q_0, Y_1)$ eta $E_2 = (R, A, \lambda_2, r_0, Y_2)$ ε-AFEDak diseinatuta ditugula. L lengoaiarentzat $E = (S, A, \lambda, s_0, Y)$ ε-AFED bat honela defini daiteke:

- $S = Q \cup R \cup \{z_0\}$. Hor, z_0 egoera Q-koa ez den eta R-koa ez den egoera bat izango da.
- A alfabetoa E_1-eko eta E_2-ko alfabeto bera izango da.
- s_0 hasierako egoera z_0 da.
- Onarpen egoeren multzoari dagokionez, $Y = Y_1 \cup Y_2$.
- Bukatzeko, λ trantsizio-funtzioa $S \times (A \cup \{\varepsilon\}) \to 2^S$ motakoa izango da eta honako hau beteko da:
  - $\lambda(q_i, \alpha) = \lambda_1(q_i, \alpha)$ izango da $q_i \in Q$ egoera denentzat eta $\alpha \in A$ sinbolo denentzat.
  - $\lambda(q_i, \varepsilon) = \lambda_1(q_i, \varepsilon)$ izango da $q_i \in Q$ egoera denentzat.
  - $\lambda(r_j, \alpha) = \lambda_2(r_j, \alpha)$ izango da $r_j \in R$ egoera denentzat eta $\alpha \in A$ sinbolo denentzat.
  - $\lambda(r_j, \varepsilon) = \lambda_2(r_j, \varepsilon)$ izango da $r_j \in R$ egoera denentzat.
  - $\lambda(z_0, \alpha) = \varnothing$ izango da $\alpha \in A$ sinbolo denentzat.
  - $\lambda(z_0, \varepsilon) = \{q_0, r_0\}$ izango da.

Orain adibide bat emango da teknika hori nola aplikatzen den erakusteko.

3.4.2.8.1 aaa duten edo a sinboloarekin hasten diren hitzen lengoaiarentzat ε-AFED baten diseinua

aaa duten edo a sinboloarekin hasten diren hitzen L lengoaiarentzat ε-AFED bat eraikitzeko, aaa duten hitzen lengoaiari dagokion E_1 ε-AFEDa eta a sinboloarekin hasten diren hitzen lengoaiari dagokion E_2 ε-AFEDa har ditzakegu abiapuntutzat. Nahikoa da hasierako egoera izango den egoera berri bat ipintzea eta egoera berri horretatik abiapuntuko E_1 eta E_2 ε-AFEDetako hasierako egoeretara ε trantsizioak ipintzea.



3.4.45 irudia. aaa duten hitzen L_1 lengoaiarentzat E_1 ε-AFED bat.

3.4.45 irudian, aaa duten hitzen L_1 lengoaiarentzat E_1 ε-AFEDa erakusten da. E_1-i dagokion boskotea honela formaliza daiteke:

$$(Q, A, \lambda_1, q_0, Y_1)$$

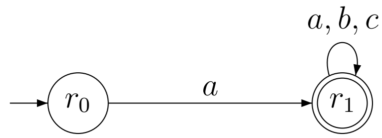
Boskote horretako osagaien definizioa honako hau da:

- $Q = \{q_0, q_1, q_2, q_3\}$
- $A = \{a, b, c\}$
- $\lambda_1 : Q \times (A \cup \{\varepsilon\}) \to 2^Q$ honako taula honen bidez definitutako trantsizio-funtzioa da:

λ_1	a	b	c	ε
q_0	$\{q_0, q_1\}$	$\{q_0\}$	$\{q_0\}$	$\varnothing$
q_1	$\{q_2\}$	$\varnothing$	$\varnothing$	$\varnothing$
q_2	$\{q_3\}$	$\varnothing$	$\varnothing$	$\varnothing$
q_3	$\{q_3\}$	$\{q_3\}$	$\{q_3\}$	$\varnothing$

- q_0 hasierako egoera da.
- $Y_1 = \{q_3\}$.

E_2



3.4.46 irudia. a sinboloarekin hasten diren hitzen L_2 lengoaiari dagokion E_2 ε-AFEDa.

E_1-en $\lambda_1(q_i, \varepsilon) = \varnothing$ betetzen da $q_i \in Q$ egoera denentzat eta, ondorioz, 3.4.45 irudiko trantsizio-diagraman ez dago ε etiketa duen trantsiziorik.

3.4.46 irudian, a sinboloarekin hasten diren hitzen L_2 lengoaiari dagokion E_2 ε-AFEDa erakusten da. E_2 honako boskote honen bidez formaliza daiteke:

$$(R, A, \lambda_2, r_0, Y_2)$$

Boskote horretako osagaien definizioa honako hau da:

- $R = \{r_0, r_1\}$
- $A = \{a, b, c\}$
- $\lambda_2 : R \times (A \cup \{\varepsilon\}) \to 2^R$ honako taula honen bidez definitutako trantsizio-funtzioa da:

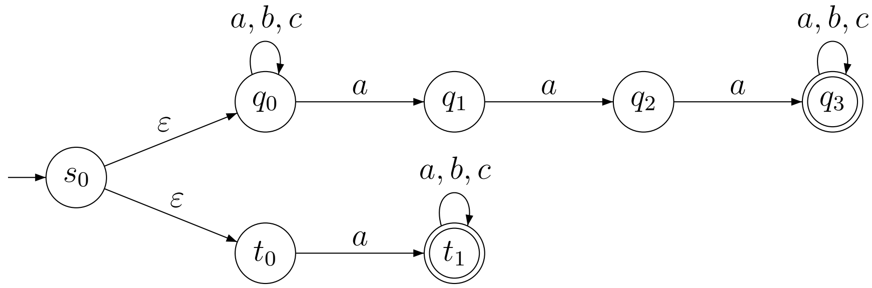
λ_2	a	b	c	ε
r_0	$\{r_1\}$	$\varnothing$	$\varnothing$	$\varnothing$
r_1	$\{r_1\}$	$\{r_1\}$	$\{r_1\}$	$\varnothing$

- r_0 hasierako egoera da.
- $Y_2 = \{r_1\}$.

E_2-n $\lambda_2(r_j, \varepsilon) = \varnothing$ betetzen da $r_j \in R$ egoera denentzat eta, ondorioz, 3.4.46 irudiko trantsizio-diagraman ez dago ε etiketa duen trantsiziorik.

3.4.47 irudian, aaa duten edo a-rekin hasten diren hitzen L lengoaiarentzat E ε-AFEDa erakusten da. E ε-AFED hori E_1 eta E_2 ε-AFEDak abiapuntutzat hartuz eta disjuntzioaren kasurako dugun teknika sistematikoa aplikatuz lortu da. 3.3.3.5.1 atalean eraiki den AFEDa-rekin konparatzea komeni da. L-rentzat eraiki den E ε-AFEDa (S, A, λ, s_0, Y) boskote gisa formaliza daiteke. Boskote horretako osagaien definizioa honako hau da:

- $S = Q \cup R \cup \{z_0\} = \{q_0, q_1, q_2, q_3, r_0, r_1, z_0\}$.



3.4.47 irudia. aaa duten edo a-rekin hasten diren hitzen L lengoaiarentzat E ε-AFEDa.

- A alfabetoa E_1-eko eta E_2-ko alfabeto bera izango da.
- s_0 hasierako egoera z_0 da.
- Onarpen egoeren multzoari dagokionez, $Y = Y_1 \cup Y_2 = \{q_3, r_1\}$.
- λ trantsizio-funtzioa $S \times (A \cup \{\varepsilon\}) \to 2^S$ motakoa izango da eta honako taula honen bidez defini daiteke:

λ	a	b	c	ε
q_0	$\{q_0, q_1\}$	$\{q_0\}$	$\{q_0\}$	$\varnothing$
q_1	$\{q_2\}$	$\varnothing$	$\varnothing$	$\varnothing$
q_2	$\{q_3\}$	$\varnothing$	$\varnothing$	$\varnothing$
q_3	$\{q_3\}$	$\{q_3\}$	$\{q_3\}$	$\varnothing$
r_0	$\{r_1\}$	$\varnothing$	$\varnothing$	$\varnothing$
r_1	$\{r_1\}$	$\{r_1\}$	$\{r_1\}$	$\varnothing$
z_0	$\varnothing$	$\varnothing$	$\varnothing$	$\{q_0, r_0\}$

E ε-AFEDan $\lambda(z_0, \varepsilon) = \{q_0, r_0\}$ da eta, ondorioz, 3.4.47 irudiko trantsizio-diagraman ε etiketa duten bi trantsizio ditugu.

3.4.2.9 ε-AFEDen diseinua $L_1 \setminus L_2$ eran definitutako lengoaientzat

L_1 eta L_2 bi lengoairen kendura gisa defini daitekeen L lengoaia bat baldin badugu, hau da, $L = L_1 \setminus L_2$ baldin bada, $L_1 \setminus L_2 = L_1 \cap \overline{L_2}$ betetzen dela kontuan hartu beharko da. Beraz, L_1 lengoaiari dagokion ε-AFED eta $\overline{L_2}$ lengoaiari dagokion ε-AFED bat har ditzakegu abiapuntutzat eta lengoaien ebakidura dugunerako teknika, hau da, konjuntzioaren teknika aplika dezakegu.

3.5.

AFD, AFED eta ε-AFEDen arteko baliokidetasuna

Gai honetan hiru motatako automata finituak bereizi ditugu: AFDak, AFEDak eta ε-AFEDak.

Automata finitu bat diseinatzerakoan, oro har, aukera erosoena ε-AFED bat eraikitzea izan ohi da. ε-AFED bidez hitzek bete beharreko propietateak eta hitzen egiturak era errazagoan adierazi ahal izango dira. Izan ere, ε-AFEDek modularitate maila handia ahalbidetzen dute eta propietate disjuntiboak bakoitza bere aldetik aztertzeko aukera ematen dute (esate baterako, konfigurazio deterministez eratutako zuhaitzetan propietate bakoitzeko adar desberdin bat eraikiz). ε-AFEDen ondoren AFEDak dira erosoenak lengoaiak adierazteko erraztasuna kontuan hartzen badugu. AFEDek ere modulartitate maila nahiko handia ahalbidetzen dute eta propietate disjuntiboak bakoitza bere aldetik aztertzeko aukera ematen dute (hala ere, modularitatea eta propietate disjuntiboen bereizketa ez da hain garbia ε-AFEDekin konparatuz). Azkenik, AFDetan propietate desberdinak adar desberdinetan aztertzeko aukera ematen duten zuhaitzak garatzerik ez dagoenez eta propietate denak bide bakar edo konfigurazio deterministez eratutako adar edo sekuentzia bakar baten bidez kontrolatu behar direnez, lengoaia batzuei dagokien AFDa diseinatzea nahiko zaila gerta daiteke.

Hiru automata finitu mota horietan propietateak adierazteko erraztasuna desberdina denez, lengoaiak definitzeko ahalmena ere desberdina izan daitekeela pentsa daiteke. Beste era batera esanda, gerta al daiteke AFD edo AFED baten bidez definitu ezin den lengoaia bat ε-AFED baten bidez definitu ahal izatea? Era berean, gerta al daiteke AFD baten bidez definitu ezin den lengoaia bat AFED baten bidez definitu ahal izatea?

Kasu bietan erantzuna ezezkoa da. Hiru automata finitu mota horiek baliokideak dira lengoaiak definitzeko duten ahalmenari dagokionez. Beraz, ε-AFED baten bidez defini daitekeen edozein lengoaia AFED baten bidez eta AFD baten bidez ere definitu ahal izango da. Era berean, AFED baten bidez defini daitekeen edozein lengoaia AFD baten bidez ere definitu ahal izango da.

Hiru automata finitu mota horien arteko baliokidetasuna era formalean froga daiteke. Alde batetik, AFED bat emanda, lengoaia bera definitzen edo adierazten duen AFDa itzultzen duen algoritmoa daukagu. Bestetik, ε-AFED bat hartu eta lengoaia bera definitzen duen AFED bat definitzeko balio duen prozedura bat ere badugu. Azkenik, AFDtik AFEDrako eraldaketa eta AFEDtik ε-AFEDrako eraldaketa oso errazak dira.

3.5.1 AFED eta ε-AFEDen arteko baliokidetasuna

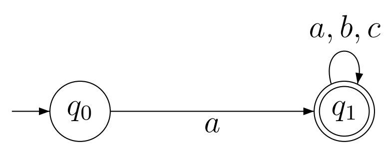
AFEDak eta ε-AFEDak baliokideak direla frogatzeko, bi gauza egin behar dira. Alde batetik, AFED bat emanda, AFED horrek definitzen duen lengoaia bera definitzen duen ε-AFED bat diseina daitekeela frogatu behar da. Bestetik, ε-AFED bat emanda, lengoaia bera definitzen duen AFED bat defini daitekeela frogatu behar da.

3.5.1.1 Edozein AFED hartuz, lengoaia bera definitzen duen ε-AFED bat eraiki daiteke

Automata finituak definitzeko erabiltzen diren grafikoak edo trantsizio-diagramak aztertzen baditugu, ε-AFEDetan ε trantsizioak baimenduta daudela ikus dezakegu, baina ez da derrigorrezkoa ε sinboloa duten trantsizioak egotea. Beraz, grafikoa aztertuz, edozein AFED hartuz, AFED hori ε-AFED bat ere bada. Adibidez, har dezagun $A = \{a, b, c\}$ alfabetoaren gainean definituta dagoen eta a sinboloaz hasten diren hitzez osatuta dagoen lengoaiari dagokion AFEDa:

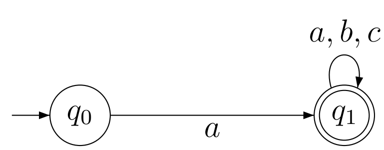
$$L_3 = \{w \mid w \in A^* \land \exists u(u \in A^* \land w = au)\}$$

N



Oraintxe esan den bezala, automata finitu hori ε-AFED bat ere badela esan dezakegu, nahiz eta ez eduki ε sinboloa duen trantsiziorik.

E



Era formalagoan, N AFED bat (Q, A, ν, q_0, Y) erako boskote bat dela kontuan hartuz eta hor $\nu : Q \times A \to 2^Q$ erakoa izango dela kontuan hartuz, automata finitu horri dagokion E ε-AFEDa (Q, A, λ, q_0, Y) erako boskote bat izango da eta $\lambda : Q \times (A \cup \{\varepsilon\}) \to 2^Q$ erakoa izango da eta bere definizioa honako hau izango da: $q_j \in Q$ egoera bakoitzeko eta A alfabetoko α sinbolo bakoitzeko, $\lambda(q_j, \alpha) = \nu(q_j, \alpha)$ eta, gainera, $q_j \in Q$ egoera bakoitzeko, $\lambda(q_j, \varepsilon) = \varnothing$.

a sinboloaz hasten diren hitzen adibidera itzuliz, ν-ren definizioa honako hau izango litzateke:

ν	a	b	c
q_0	$\{q_1\}$	$\varnothing$	$\varnothing$
q_1	$\{q_1\}$	$\{q_1\}$	$\{q_1\}$

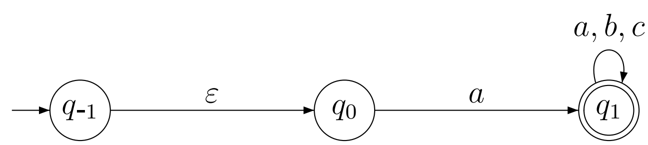
λ, aldiz, honela definituta egongo litzateke:

λ	a	b	c	ε
q_0	$\{q_1\}$	$\varnothing$	$\varnothing$	$\varnothing$
q_1	$\{q_1\}$	$\{q_1\}$	$\{q_1\}$	$\varnothing$

Adibide honetako N AFEDa eta E ε-AFEDa grafikoki berdinak izan arren, ν eta λ definitzen dituzten taulak desberdinak dira eta hor garbi ikus daiteke ν funtzioa AFED bati dagokion trantsizio-funtzioa dela eta λ funtzioa ε-AFED bati dagokion trantsizio-funtzioa dela.

Dena den, AFED bat hartu eta ε-AFED bat eraikitzean grafikoki ere gutxienez ε sinboloa duen trantsizio bat agertzea nahi baldin badugu, hori era sistemakioan egiteko balio duen bide bat badugu. Horretarako, hasierako egoera izango den $q_{\text{-}1}$ egoera berri bat ipintzea eta egoera berri horretatik lehen hasierako egoera zen egoerara (beraz, $q_{\text{-}1}$ egoeratik q_0 egoerara) ε trantsizio bat ipintzea nahikoa izango da. a sinboloaz hasten diren hitzei dagokien AFEDa berriro hartuz, metodo hau jarraituz lortuko genukeen E' ε-AFEDa honako hau izango litzateke:

E'



E'-ri dagokion λ funtzioaren taula honako hau izango litzateke:

λ	a	b	c	ε
$q_{\text{-}1}$	$\varnothing$	$\varnothing$	$\varnothing$	$\{q_0\}$
q_0	$\{q_1\}$	$\varnothing$	$\varnothing$	$\varnothing$
q_1	$\{q_1\}$	$\{q_1\}$	$\{q_1\}$	$\varnothing$

3.5.1.2 Edozein ε-AFED hartuz, lengoaia bera definitzen duen AFED bat eraiki daiteke

(Q, A, λ, q_0, Y) egitura duen ε-AFED bat emanda, ε-AFED horri dagokion (Q', A', ν, q_0', Y') AFEDak Q izango du egoera multzo gisa, A izango du alfabeto gisa eta q_0 hasierako egoeratzat. Beraz, $Q' = Q$, $A' = A$ eta $q_0' = q_0$ izango dira. Gainera, Y multzoko egoera denak Y' multzoan ere egongo dira eta zalantza bakarra q_0'-rekin izango dugu. q_0 egoera Y multzokoa baldin bada, orduan q_0' ere Y' multzokoa izango da. Baina q_0 ez bada Y multzokoa, orduan honako irizpide hau hartu beharko da kontuan: $\mathsf{Itxidura}\text{-}\varepsilon^*(\{q_0\}, \lambda) \cap Y \neq \varnothing$ betetzen bada, orduan q_0' egoerak onarpen egoeren multzokoa izan beharko du AFED horretan, hau da, q_0' egoerak Y' multzokoa izan beharko du; bestalde, $\mathsf{Itxidura}\text{-}\varepsilon^*(\{q_0\}, \lambda) \cap Y = \varnothing$ betetzen bada, orduan q_0'

egoerak ez du egon behar onarpen egoeren multzoan AFED horretan, hau da, q_0' egoerak Y' multzokoa ez du izan behar. Beste era batera esanda, ε-AFEDan q_0 egoeratik abiatuz eta ε trantsizioak bakarrik jarraituz Y multzoko egoera batera iristerik baldin badago, orduan q_0' egoerak Y' multzokoa izan beharko du AFEDan. Hasierako egoera den q_0'-ri lotutako berezitasun hau AFEDak ε hitzarentzat erantzun zuzena eman ahal izateko da. Azkenik, Q' multzoko q_j egoera bakoitzeko eta A alfabetoko α sinbolo bakoitzeko, $\nu(q_j, \alpha) = \lambda^*(\{q_j\}, \alpha)$ beteko da. Hor, ν funtzioa $Q \times A \to 2^Q$ erakoa izango da eta λ^* funtzioa $2^Q \times A^* \to 2^Q$ motakoa izango da. Beraz, $\lambda^*(\{q_j\}, \alpha)$ idazten dugunean berez $\lambda^*(\{q_j\}, \alpha\varepsilon)$ daukagu eta, ondorioz, hasteko $\{q_j\}$ multzotik Itxidura-ε^* funtzioaren bidez zein egoeretara iritsiko garen kalkulatuko da, gero Itxidura-$\varepsilon^*(\{q_j\})$ multzoko egoera bakoitzetik α sinboloa irakurriz zein egoeretara iritsiko garen kalkulatuko da eta, bukatzeko, α irakurri ondoren lortu den egoera-multzoari Itxidura-ε^* aplikatuko zaio. Laburtuz, era horretara lortutako AFEDak (Q, A, ν, q_0, Y') egitura izango du. Hor, ν funtzioa oraintxe adierazi den erara definitutako trantsizio-funtzioa izango da eta Y'-ri dagokionez, gerta daiteke Y-ren berdina izatea edo $Y \cup \{q_0\}$ multzoaren berdina izatea, q_0-ren kasurako zehaztu den irizpidea kontuan hartuz. Edozein kasutan, $Y \subseteq Y'$ beteko da.

Formalki, E ε-AFED bati dagokion N AFEDaren trantsizioak kalkulatzeko, ν kalkulatu behar da $q_i \in Q$ egoera bakoitzeko eta $\alpha \in A$ sinbolo bakoitzeko. $\nu(q_i, \alpha)$ kalkulatzeko $\lambda^*(\{q_i\}, \alpha)$ kalkulatu behar da eta $\lambda^*(\{q_i\}, \alpha)$ kalkulatzeko hiru era daude: definizioari jarraituz; konfigurazio ez-deterministez osatutako sekuentzia kalkulatuz; edo konfigurazio deterministez osatutako zuhaitza kalkulatuz. Ipini beharreko trantsizioak zein diren erabakitzeko, honako hau hartu beharko da kontuan:

- $\lambda^*(\{q_i\}, \alpha)$ kalkulatzeko, definizioari jarraitzen badiogu, multzo bat lortuko da eta q_i-tik multzo horretako egoera bakoitzera trantsizio bat ipini beharko da α sinboloarekin. Multzo hori hutsa baldin bada, ez da trantsiziorik ipini behar q_i-tik α sinboloarekin.

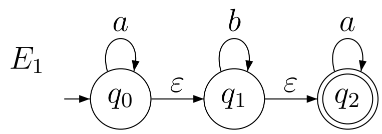
- $\lambda^*(\{q_i\}, \alpha)$ kalkulatzeko, konfigurazio ez-deterministez osatutako sekuentzia kalkulatzen badugu, multzo batez eta ε hitzaz osatutako konfigurazio ez-determinista bat lortuko da eta q_i-tik multzo horretako egoera bakoitzera trantsizio bat ipini beharko da α sinboloarekin. Multzo hori hutsa baldin bada, ez da trantsiziorik ipini behar q_i-tik α sinboloarekin.

- Azkenik, $\lambda^*(\{q_i\}, \alpha)$ kalkulatzeko, konfigurazio deterministez osatutako zuhaitza kalkulatzen badugu, egoera batez eta ε hitzaz osatutako konfigurazio deterministak hartu beharko dira kontuan zuhaitz horretan eta q_i-tik egoera horietako bakoitzera trantsizio bat ipini beharko da α sinboloarekin. Gerta daiteke zuhaitzean egoera batez eta ε hitzaz osatutako konfigurazio deterministarik ez agertzea eta, kasu horretan, ez da trantsiziorik ipini behar q_i-tik α sinboloarekin.

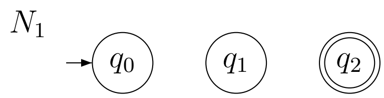
Jarraian, 3.4.1 atalaren hasieran aurkeztu diren E_1 eta E_2 ε-AFEDak hartuko ditugu berriro eta bakoitzari dagokion AFEDa kalkulatuko dugu.

3.5.1.2.1 Adibidea: ε-AFED bati dagokion AFEDaren kalkulua

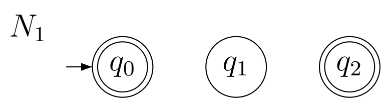
Atal honetan, E_1 ε-AFEDari dagokion N_1 AFEDa kalkulatuko dugu:



Badakigu N_1 AFED berriko egoerak E_1 ε-AFEDan ditugun egoerak izango direla:



Itxidura-$\varepsilon^*(\{q_0\}, \lambda) = \{q_0, q_1, q_2\}$ denez eta, ondorioz, Itxidura-$\varepsilon^*(\{q_0\}, \lambda) \cap Y \neq \varnothing$ betetzen denez, q_0 egoera Y' multzokoa izango da, hau da, zirkulu bikoitza izango du:



Orain N_1 AFEDaren trantsizioak kalkulatu behar dira, hau da, ν kalkulatu behar da $q_i \in Q$ egoera bakoitzeko eta $\alpha \in A$ sinbolo bakoitzeko. Kasu bakoitzean hiru eratara egingo dugu: definizioari jarraituz; konfigurazio ez-deterministez osatutako sekuentzia kalkulatuz; edo konfigurazio deterministez osatutako zuhaitza kalkulatuz.

- $\nu(q_0, a)$:

  - Definizioari jarraituz: $\nu(q_0, a) = \lambda^*(\{q_0\}, a) = \lambda^*(\lambda(q_0, a) \cup \lambda(q_1, a) \cup \lambda(q_2, a), \varepsilon) = \lambda^*(\{q_0\} \cup \{q_1\} \cup \{q_2\}, \varepsilon) = \lambda^*(\{q_0, q_1, q_2\}, \varepsilon) =$ Itxidura-$\varepsilon^*(\{q_0, q_1, q_2\}, \lambda) = \{q_0, q_1, q_2\}$.
  - Konfigurazio ez-deterministez osatutako sekuentzia gisa

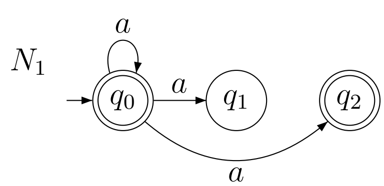
    $$(\{q_0\}, a) \equiv_{\text{Itxidura-}\varepsilon^*} (\{q_0, q_1, q_2\}, a)$$
    $$\Big|$$
    $$(\{q_0, q_2\}, \varepsilon) \equiv_{\text{Itxidura-}\varepsilon^*} (\{q_0, q_1, q_2\}, \varepsilon)$$

    Sekuentzia horretako osagai bakoitzean ε trantsizioak kontuan hartuz, hau da, Itxidura-ε^* kontuan hartuz lortzen den konfigurazioa zein den zehazten da.
  - Konfigurazio deterministez osatutako zuhaitz gisa:

$$\begin{array}{cc} \multicolumn{2}{c}{(q_0, a)} \\ (q_0, \varepsilon) & (q_1, a) \\ | & | \\ (q_1, \varepsilon) & (q_2, a) \\ | & | \\ (q_2, \varepsilon) & (q_2, \varepsilon) \end{array}$$

Zuhaitza garatzea aukeratzen bada, hitz bezala ε duten konfigurazioak interesatuko zaizkigu bukaeran. q_0, q_1 eta q_2 egoerak dira ε duten konfigurazioetan agertzen diren egoerak.

Kalkulua era batera edo bestera egin, ondorioa bat bera da: q_0-tik a-rekin q_0, q_1 eta q_2 egoeretara trantsizioa ipini behar da. Beraz, honako hiru trantsizio hauek gehitu behar zaizkio eraikitzen ari garen AFEDari:



- $\nu(q_0, b)$:

  - Definizioari jarraituz: $\nu(q_0, b) = \lambda^*(\{q_0\}, b) = \lambda^*(\lambda(q_0, b) \cup \lambda(q_1, b) \cup \lambda(q_2, b), \varepsilon) = \lambda^*(\varnothing \cup \{q_1\} \cup \varnothing, \varepsilon) = \lambda^*(\{q_1\}, \varepsilon) = \textsf{Itxidura-}\varepsilon^*(\{q_1\}, \lambda) = \{q_1, q_2\}$.
  - Konfigurazio ez-deterministez osatutako sekuentzia gisa:

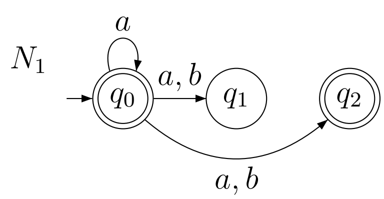
$$\begin{array}{c} (\{q_0\}, b) \equiv_{\textsf{Itxidura-}\varepsilon^*} (\{q_0, q_1, q_2\}, b) \\ | \\ (\{q_1\}, \varepsilon) \equiv_{\textsf{Itxidura-}\varepsilon^*} (\{q_1, q_2\}, \varepsilon) \end{array}$$

    Sekuentzia horretako osagai bakoitzean ε trantsizioak kontuan hartuz, hau da, $\textsf{Itxidura-}\varepsilon^*$ kontuan hartuz lortzen den konfigurazioa zein den zehazten da.
  - Konfigurazio deterministez osatutako zuhaitz gisa:

$$\begin{array}{cc} \multicolumn{2}{c}{(q_0, b)} \\ \multicolumn{2}{c}{|} \\ \multicolumn{2}{c}{(q_1, b)} \\ (q_1, \varepsilon) & (q_2, b) \\ | & \\ (q_2, \varepsilon) & \end{array}$$

Zuhaitz horretan hitz bezala ε duten konfigurazioak interesatzen zaizkigu. q_1 eta q_2 egoerak dira ε duten konfigurazioetan agertzen diren egoerak.

Ondorioa honako hau da: q_0-tik b-rekin q_1 eta q_2 egoeretara trantsizioa ipini behar da. Beraz, honako bi trantsizio hauek gehitu behar zaizkio eraikitzen ari garen AFEDari:



- $\nu(q_0, c)$:

  - Definizioari jarraituz:
  - $\nu(q_0, c) = \lambda^*(\{q_0\}, c) = \lambda^*(\lambda(q_0, c) \cup \lambda(q_1, c) \cup \lambda(q_2, c), \varepsilon) = \lambda^*(\varnothing \cup \varnothing \cup \varnothing, \varepsilon) = \lambda^*(\varnothing, \varepsilon) = \text{Itxidura-}\varepsilon^*(\varnothing, \lambda) = \varnothing$.
  - Konfigurazio ez-deterministez osatutako sekuentzia gisa:

$$\begin{array}{c} (\{q_0\}, c) \equiv_{\text{Itxidura-}\varepsilon^*} (\{q_0, q_1, q_2\}, c) \\ | \\ (\varnothing, \varepsilon) \equiv_{\text{Itxidura-}\varepsilon^*} (\varnothing, \varepsilon) \end{array}$$

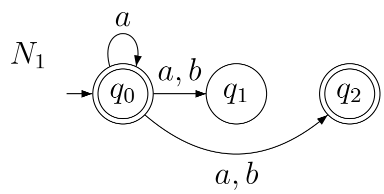
    Sekuentzia horretako osagai bakoitzean ε trantsizioak kontuan hartuz, hau da, Itxidura-ε^* kontuan hartuz lortzen den konfigurazioa zein den zehazten da.

  - Konfigurazio deterministez osatutako zuhaitz gisa:

$$\begin{array}{c} (q_0, c) \\ | \\ (q_1, c) \\ | \\ (q_2, c) \end{array}$$

    Zuhaitz horretan, ε duten konfigurazioak hartu beharko dira kontuan. Kasu honetan ez dago ε hitza duen konfiguraziorik.

Definizioari jarraituz eta konfigurazio ez-deterministen bidez multzo hutsa lortu da eta konfigurazio deterministen bidez zuhaitza garatuz ε hitza duen konfiguraziorik ez da lortu. Horrek guztiak trantsiziorik ez dela gehitu behar esan nahi du.



- $\nu(q_1, a)$:

  - Definizioari jarraituz: $\nu(q_1, a) = \lambda^*(\{q_1\}, a) = \lambda^*(\lambda(q_1, a) \cup \lambda(q_2, a), \varepsilon) = \lambda^*(\varnothing \cup \{q_2\}, \varepsilon) = \lambda^*(\{q_2\}, \varepsilon) = \text{Itxidura-}\varepsilon^*(\{q_2\}, \lambda) = \{q_2\}$.
  - Konfigurazio ez-deterministez osatutako sekuentzia gisa:

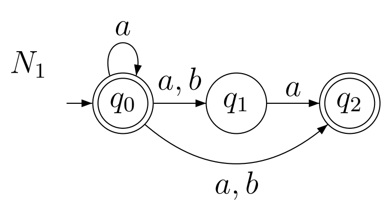
$$\begin{array}{c} (\{q_1\}, a) \equiv_{\text{Itxidura-}\varepsilon^*} (\{q_1, q_2\}, a) \\ | \\ (\{q_2\}, \varepsilon) \equiv_{\text{Itxidura-}\varepsilon^*} (\{q_2\}, \varepsilon) \end{array}$$

    Sekuentzia horretako osagai bakoitzean ε trantsizioak kontuan hartuz, hau da, Itxidura-ε^* kontuan hartuz lortzen den konfigurazioa zein den zehazten da.
  - Konfigurazio deterministez osatutako zuhaitz gisa:

$$\begin{array}{c} (q_1, a) \\ | \\ (q_2, a) \\ | \\ (q_2, \varepsilon) \end{array}$$

    Kasu honetan ε hitza duen konfigurazio batean agertzen den egoera bakarra q_2 da.

Ondorioa: trantsizio bat gehitu behar zaio eraikitzen ari garen AFEDari. Hain zuzen ere a sinboloarekin q_1-etik q_2-ra doan trantsizioa:



- $\nu(q_1, b)$:

  - Definizioari jarraituz: $\nu(q_1, b) = \lambda^*(\{q_1\}, b) = \lambda^*(\lambda(q_1, b) \cup \lambda(q_2, b), \varepsilon) = \lambda^*(\{q_1\} \cup \varnothing, \varepsilon) = \lambda^*(\{q_1\}, \varepsilon) = \text{Itxidura-}\varepsilon^*(\{q_1\}, \lambda) = \{q_1, q_2\}$.

– Konfigurazio ez-deterministez osatutako sekuentzia gisa:

$$\begin{array}{c} (\{q_1\}, b) \equiv_{\textsf{Itxidura-}\varepsilon^*} (\{q_1, q_2\}, b) \\ | \\ (\{q_1\}, \varepsilon) \equiv_{\textsf{Itxidura-}\varepsilon^*} (\{q_1, q_2\}, \varepsilon) \end{array}$$

Sekuentzia horretako osagai bakoitzean ε trantsizioak kontuan hartuz, hau da, Itxidura-ε^* kontuan hartuz lortzen den konfigurazioa zein den zehazten da.

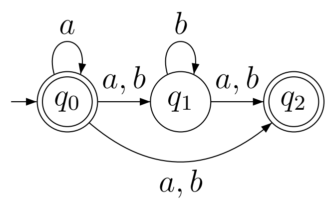
– Konfigurazio deterministez osatutako zuhaitz gisa:

$$\begin{array}{ccc} & (q_1, b) & \\ \swarrow & & \searrow \\ (q_1, \varepsilon) & & (q_2, b) \\ | & & \\ (q_2, \varepsilon) & & \end{array}$$

Zuhaitz horretan ε hitza duten bi konfigurazio daude. Konfigurazio horietako egoerak q_1 eta q_2 dira.

Kalkulua egiteko hiru eratatik edozein aukeratuta ere, ondorioa bat bera da: q_1-etik q_1-era eta q_2-ra trantsizio bana ipini beharko da b sinboloarekin.

N_1



- $\nu(q_1, c)$:

  – Definizioari jarraituz: $\nu(q_1, c) = \lambda^*(\{q_1\}, c) = \lambda^*(\lambda(q_1, c) \cup \lambda(q_2, c), \varepsilon) = \lambda^*(\varnothing \cup \varnothing, \varepsilon) = \lambda^*(\varnothing, \varepsilon) = \textsf{Itxidura-}\varepsilon^*(\varnothing, \lambda) = \varnothing$.

  – Konfigurazio ez-deterministez osatutako sekuentzia gisa:

$$\begin{array}{c} (\{q_1\}, c) \equiv_{\textsf{Itxidura-}\varepsilon^*} (\{q_1, q_2\}, c) \\ | \\ (\varnothing, \varepsilon) \equiv_{\textsf{Itxidura-}\varepsilon^*} (\varnothing, \varepsilon) \end{array}$$

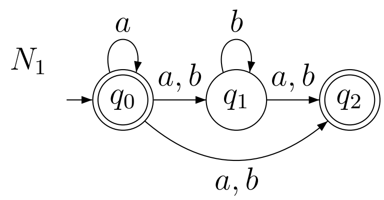
  Sekuentzia horretako osagai bakoitzean ε trantsizioak kontuan hartuz, hau da, Itxidura-ε^* kontuan hartuz lortzen den konfigurazioa zein den zehazten da.

  – Konfigurazio deterministez osatutako zuhaitz gisa:

$$\begin{array}{c}(q_1, c)\\ | \\ (q_2, c)\end{array}$$

Zuhaitz horretan, ε hitza duen konfigurazioak ez dago.

Definizioari jarraituz eta konfigurazio ez-deterministen bidez multzo hutsa lortu da eta konfigurazio deterministen bidez zuhaitza garatuz ε hitza duen konfiguraziorik ez da lortu. Hori dela-eta, trantsiziorik ez da gehitu behar.



- $\nu(q_2, a)$:

  - Definizioari jarraituz: $\nu(q_2, a) = \lambda^*(\{q_2\}, a) = \lambda^*(\lambda(q_2, a), \varepsilon) = \lambda^*(\{q_2\}, \varepsilon) = \textsf{Itxidura-}\varepsilon^*(\{q_2\}, \lambda) = \{q_2\}$.
  - Konfigurazio ez-deterministez osatutako sekuentzia gisa:

$$\begin{array}{c}(\{q_2\}, a) \equiv_{\textsf{Itxidura-}\varepsilon^*} (\{q_2\}, a)\\ | \\ (\{q_2\}, \varepsilon) \equiv_{\textsf{Itxidura-}\varepsilon^*} (\{q_2\}, \varepsilon)\end{array}$$

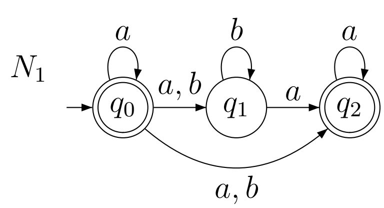
    Sekuentzia horretako osagai bakoitzean ε trantsizioak kontuan hartuz, hau da, $\textsf{Itxidura-}\varepsilon^*$ kontuan hartuz lortzen den konfigurazioa zein den zehazten da.

  - Konfigurazio deterministez osatutako zuhaitz gisa:

$$\begin{array}{c}(q_2, a)\\ | \\ (q_2, \varepsilon)\end{array}$$

    Zuhaitz horretan ε hitza duen konfigurazio bakarra daukagu. Konfigurazio horretako egoera q_2 da.

Kalkulua egiteko hiru eretan ondorio bera dugu: a sinboloarekin q_2-tik q_2-ra doan trantsizio bat gehitu beharko da:



- $\nu(q_2, b)$:

  - Definizioari jarraituz: $\nu(q_2, b) = \lambda^*(\{q_2\}, b) = \lambda^*(\lambda(q_2, b), \varepsilon) = \lambda^*(\varnothing, \varepsilon) = \mathsf{Itxidura}\text{-}\varepsilon^*(\varnothing, \lambda) = \varnothing$.
  - Konfigurazio ez-deterministez osatutako sekuentzia gisa:

$$\begin{array}{c} (\{q_2\}, b) \equiv_{\mathsf{Itxidura}\text{-}\varepsilon^*} (\{q_2\}, b) \\ | \\ (\varnothing, \varepsilon) \equiv_{\mathsf{Itxidura}\text{-}\varepsilon^*} (\varnothing, \varepsilon) \end{array}$$

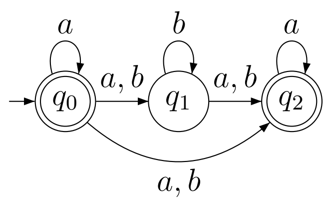
    Sekuentzia horretako osagai bakoitzean ε trantsizioak kontuan hartuz, hau da, Itxidura-ε^* kontuan hartuz lortzen den konfigurazioa zein den zehazten da.
  - Konfigurazio deterministez osatutako zuhaitz gisa:

$$(q_2, b)$$

    Zuhaitz horretan ez dago ε hitza duen konfiguraziorik.

  Ez da trantsiziorik gehitu behar:

N_1



- $\nu(q_2, c)$:

  - Definizioari jarraituz: $\nu(q_2, c) = \lambda^*(\{q_2\}, c) = \lambda^*(\lambda(q_2, c), \varepsilon) = \lambda^*(\varnothing, \varepsilon) = \mathsf{Itxidura}\text{-}\varepsilon^*(\varnothing, \lambda) = \varnothing$.
  - Konfigurazio ez-deterministez osatutako sekuentzia gisa:

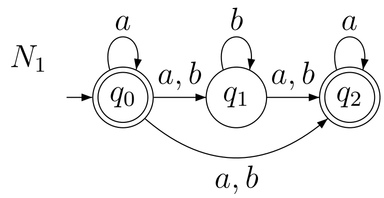
$$\begin{array}{c} (\{q_2\}, c) \equiv_{\mathsf{Itxidura}\text{-}\varepsilon^*} (\{q_2\}, c) \\ | \\ (\varnothing, \varepsilon) \equiv_{\mathsf{Itxidura}\text{-}\varepsilon^*} (\varnothing, \varepsilon) \end{array}$$

    Sekuentzia horretako osagai bakoitzean ε trantsizioak kontuan hartuz, hau da, Itxidura-ε^* kontuan hartuz lortzen den konfigurazioa zein den zehazten da.
  - Konfigurazio deterministez osatutako zuhaitz gisa:

$$(q_2, c)$$

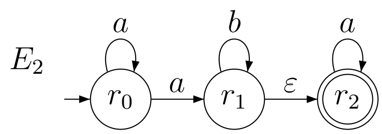
    Zuhaitz horretan ere ez dugu lortu ε hitza duen konfiguraziorik.

Ez da gezi berririk gehitu behar.

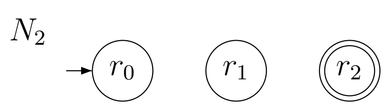


3.5.1.2.2 Adibidea: beste ε-AFED bati dagokion AFEDaren kalkulua

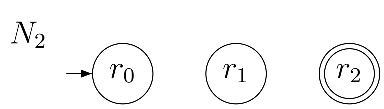
Jarraian E_2 ε-AFEDari dagokion N_2 AFEDa lortuko dugu:



Hasteko, badakigu N_2 AFEDaren egoerak E_2 ε-AFEDak dituen egoerak izango direla:



Itxidura-$\varepsilon^*(\{r_0\}, \lambda) = \{r_0\}$ denez, Itxidura-$\varepsilon^*(\{r_0\}, \lambda) \cap Y = \varnothing$ betetzen da eta, ondorioz, r_0 ez da Y' multzokoa izango, hau da, r_0 egoerak ez ditu bi zirkulu izango:



Orain N_2 AFEDari dagozkion trantsizioak kalkulatu behar dira, hau da, ν kalkulatu behar da. Horretarako r_j egoera bakoitzeko eta alfabetoko α sinbolo bakoitzeko $\nu(r_0 j, \alpha)$ kalkulatu beharko da. Kalkulu hori egiteko hiru erak emango dira kasu bakoitzean:

- $\nu(r_0, a)$:

- Definizioari jarraituz:
- $\nu(r_0, a) = \lambda^*(\{r_0\}, a) = \lambda^*(\lambda(q_0, a), \varepsilon) = \lambda^*(\{r_0, r_1\}, \varepsilon) = \lambda^*(\{r_0, r_1\}, \varepsilon) = \text{Itxidura-}\varepsilon^*(\{r_0, r_1\}, \lambda) = \{r_0, r_1\}$.
- Konfigurazio ez-deterministez osatutako sekuentzia gisa:

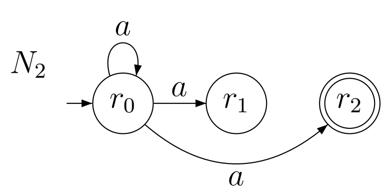
$$\begin{array}{c} (\{r_0\}, a) \equiv_{\text{Itxidura-}\varepsilon^*} (\{r_0\}, a) \\ | \\ (\{r_0, r_1\}, \varepsilon) \equiv_{\text{Itxidura-}\varepsilon^*} (\{r_0, r_1, r_2\}, \varepsilon) \end{array}$$

  Sekuentzia horretako osagai bakoitzean ε trantsizioak kontuan hartuz, hau da, Itxidura-ε^* kontuan hartuz lortzen den konfigurazioa zein den zehazten da.
- Konfigurazio deterministez osatutako zuhaitz gisa:

$$\begin{array}{ccc} & (r_0, a) & \\ (r_0, \varepsilon) & & (r_1, \varepsilon) \\ & & | \\ & & (r_2, \varepsilon) \end{array}$$

  Zuhaitza garatzea aukeratzen bada, ε hitza duten konfigurazioak hartu beharko dira kontuan. Kasu honetan, ε hitza duten konfigurazioetan r_0, r_1 eta r_2 egoerak ditugu.

Kalkulua era batera edo bestera egin, ondorioa berera iritsi gara: r_0-tik a-rekin r_0, r_1 eta r_2 egoeretara trantsizioa ipini behar da. Beraz, honako hiru trantsizio hauek gehitu behar zaizkio eraikitzen ari garen AFEDari:



- $\nu(r_0, b)$:

  - Definizioari jarraituz: $\nu(r_0, b) = \lambda^*(\{r_0\}, b) = \lambda^*(\lambda(r_0, b), \varepsilon) = \lambda^*(\varnothing, \varepsilon) = \text{Itxidura-}\varepsilon^*(\varnothing, \lambda) = \varnothing$.
  - Konfigurazio ez-deterministez osatutako sekuentzia gisa:

$$\begin{array}{c} (\{r_0\}, b) \equiv_{\text{Itxidura-}\varepsilon^*} (\{r_0\}, b) \\ | \\ (\varnothing, \varepsilon) \equiv_{\text{Itxidura-}\varepsilon^*} (\varnothing, \varepsilon) \end{array}$$

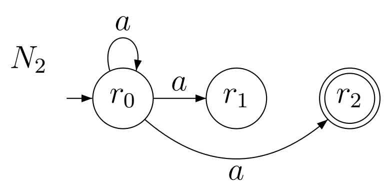
    Sekuentzia horretako osagai bakoitzean ε trantsizioak kontuan hartuz, hau da, Itxidura-ε^* kontuan hartuz lortzen den konfigurazioa zein den zehazten da.

– Konfigurazio deterministez osatutako zuhaitz gisa:

$$(r_0, b)$$

Zuhaitz horretan ez dago ε hitza duen konfiguraziorik.

Definizioari jarraituz eta konfigurazio ez-deministen bidez multzo hutsa lortu da eta konfigurazio deterministen bidez zuhaitza garatuz ε hitza duen konfiguraziorik ez da lortu. Ondorioz, ez da trantsiziorik gehitu behar.



- $\nu(r_0, c)$:

  – Definizioari jarraituz: $\nu(r_0, c) = \lambda^*(\{r_0\}, c) = \lambda^*(\lambda(r_0, c), \varepsilon) = \lambda^*(\varnothing\varepsilon) = \textsf{Itxidura-}\varepsilon^*(\varnothing, \lambda) = \varnothing$.

  – Konfigurazio ez-deterministez osatutako sekuentzia gisa:

$$\begin{array}{c} (\{r_0\}, c) \equiv_{\textsf{Itxidura-}\varepsilon^*} (\{r_0\}, c) \\ | \\ (\varnothing, \varepsilon) \equiv_{\textsf{Itxidura-}\varepsilon^*} (\varnothing, \varepsilon) \end{array}$$

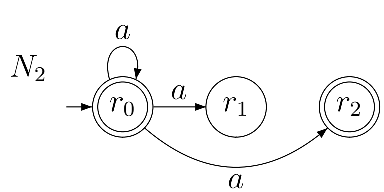
  Sekuentzia horretako osagai bakoitzean ε trantsizioak kontuan hartuz, hau da, Itxidura-ε^* kontuan hartuz lortzen den konfigurazioa zein den zehazten da.

  – Konfigurazio deterministez osatutako zuhaitz gisa:

$$(r_0, c)$$

  Zuhaitz horretan ε duen konfiguraziorik ez dago.

Definizioari jarraituz eta konfigurazio ez-deministen bidez multzo hutsa lortu da eta konfigurazio deterministen bidez zuhaitza garatuz ε hitza duen konfiguraziorik ez da lortu. Horrek guztiak trantsiziorik ez dela gehitu behar esan nahi du.



- $\nu(r_1, a)$:

  - Definizioari jarraituz: $\nu(r_1, a) = \lambda^*(\{r_1\}, a) = \lambda^*(\lambda(r_1, a) \cup \lambda(r_2, a), \varepsilon) = \lambda^*(\varnothing \cup \{r_2\}, \varepsilon) = \lambda^*(\{r_2\}, \varepsilon) = \textsf{Itxidura-}\varepsilon^*(\{r_2\}, \lambda) = \{r_2\}$.
  - Konfigurazio ez-deterministez osatutako sekuentzia gisa:

$$\begin{array}{c} (\{r_1\}, a) \equiv_{\textsf{Itxidura-}\varepsilon^*} (\{r_1, r_2\}, a) \\ | \\ (\{r_2\}, \varepsilon) \equiv_{\textsf{Itxidura-}\varepsilon^*} (\{r_2\}, \varepsilon) \end{array}$$

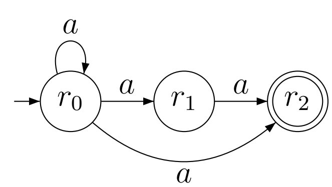
    Sekuentzia horretako osagai bakoitzean ε trantsizioak kontuan hartuz, hau da, $\textsf{Itxidura-}\varepsilon^*$ kontuan hartuz lortzen den konfigurazioa zein den zehazten da.
  - Konfigurazio deterministez osatutako zuhaitz gisa:

$$\begin{array}{c} (r_1, a) \\ | \\ (r_2, a) \\ | \\ (r_2, \varepsilon) \end{array}$$

    Zuhaitz horretan ε hitza duen konfigurazio bakarra dugu eta konfigurazio horretako egoera r_2 da.

  Kalkulua egiteko hiru eretan ondorio berera iritsi gara: r_1-etik a-rekin r_2 egoerara trantsizioa ipini behar da.

N_2



- $\nu(r_1, b)$:

  - Definizioari jarraituz: $\nu(r_1, b) = \lambda^*(\{r_1\}, b) = \lambda^*(\lambda(r_1, b) \cup \lambda(r_2, b), \varepsilon) = \lambda^*(\{r_1\} \cup \varnothing, \varepsilon) = \lambda^*(\{r_1\}, \varepsilon) = \textsf{Itxidura-}\varepsilon^*(\{r_1\}, \lambda) = \{r_1, r_2\}$.
  - Konfigurazio ez-deterministez osatutako sekuentzia gisa:

$$\begin{array}{c} (\{r_1\}, b) \equiv_{\textsf{Itxidura-}\varepsilon^*} (\{r_1, r_2\}, b) \\ | \\ (\{r_1\}, \varepsilon) \equiv_{\textsf{Itxidura-}\varepsilon^*} (\{r_1, r_2\}, \varepsilon) \end{array}$$

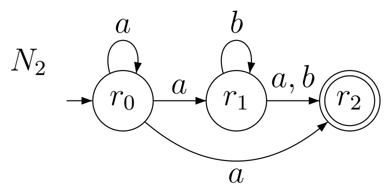
    Sekuentzia horretako osagai bakoitzean ε trantsizioak kontuan hartuz, hau da, $\textsf{Itxidura-}\varepsilon^*$ kontuan hartuz lortzen den konfigurazioa zein den zehazten da.

– Konfigurazio deterministez osatutako zuhaitz gisa:

$$\begin{array}{ccc} & (r_1, b) & \\ (r_1, \varepsilon) & & (r_2, b) \\ | & & \\ (r_2, \varepsilon) & & \end{array}$$

ε hitza duten bi konfigurazio ditugu zuhaitz horretan eta konfigurazio horietako egoerak r_1 eta r_2 dira.

Definizioari jarraituz eta konfigurazio ez-deterministen bidez $\{r_1, r_2\}$ multzoa lortu da eta konfigurazio deterministen bidez zuhaitza garatuz ε hitza duten konfigurazioetan r_1 eta r_2 egoerak ditugu. Ondorioz, bi trantsizio gehitu behar dira r_1-etik b sinboloarekin: bata r_1-era eta bestea r_2-ra.



- $\nu(r_1, c)$:

  – Definizioari jarraituz: $\nu(r_1, c) = \lambda^*(\{r_1\}, c) = \lambda^*(\lambda(r_1, c) \cup \lambda(r_2, c), \varepsilon) = \lambda^*(\varnothing \cup \varnothing, \varepsilon) = \lambda^*(\varnothing, \varepsilon) = \text{Itxidura-}\varepsilon^*(\varnothing, \lambda) = \varnothing$.

  – Konfigurazio ez-deterministez osatutako sekuentzia gisa:

$$\begin{array}{ccc} (\{r_1\}, c) & \equiv_{\text{Itxidura-}\varepsilon^*} & (\{r_1, r_2\}, c) \\ & & | \\ (\varnothing, \varepsilon) & \equiv_{\text{Itxidura-}\varepsilon^*} & (\varnothing, \varepsilon) \end{array}$$

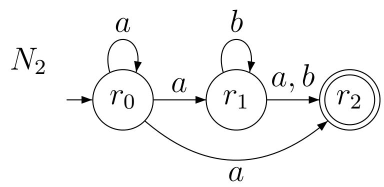
  Sekuentzia horretako osagai bakoitzean ε trantsizioak kontuan hartuz, hau da, Itxidura-ε^* kontuan hartuz lortzen den konfigurazioa zein den zehazten da.

  – Konfigurazio deterministez osatutako zuhaitz gisa:

$$\begin{array}{c} (r_1, c) \\ | \\ (r_2, c) \end{array}$$

  Zuhaitz horretan ez da lortu ε hitza duen konfiguraziorik.

Definizioari jarraituz eta konfigurazio ez-deterministen bidez multzo hutsa lortu da eta konfigurazio deterministen bidez zuhaitza garatuz ε hitza duen konfiguraziorik ez da lortu. Beraz, ez da trantsiziorik gehitu behar r_1-etik c sinboloarekin.



- $\nu(r_2, a)$:

  - Definizioari jarraituz: $\nu(r_2, a) = \lambda^*(\{r_2\}, a) = \lambda^*(\lambda(r_2, a), \varepsilon) = \lambda^*(\{r_2\}, \varepsilon) =$ Itxidura-$\varepsilon^*(\{r_2\}, \lambda) = \{r_2\}$.
  - Konfigurazio ez-deterministez osatutako sekuentzia gisa:

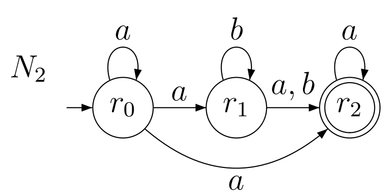
$$\begin{array}{ccc} (\{r_2\}, a) \equiv_{\text{Itxidura-}\varepsilon^*} & (\{r_2\}, a) \\ & | \\ (\{r_2\}, \varepsilon) \equiv_{\text{Itxidura-}\varepsilon^*} & (\{r_2\}, \varepsilon) \end{array}$$

    Sekuentzia horretako osagai bakoitzean ε trantsizioak kontuan hartuz, hau da, Itxidura-ε^* kontuan hartuz lortzen den konfigurazioa zein den zehazten da.
  - Konfigurazio deterministez osatutako zuhaitz gisa:

$$\begin{array}{c} (r_2, a) \\ | \\ (r_2, \varepsilon) \end{array}$$

    Zuhaitz horretan, ε hitza duen konfigurazio bat daukagu eta konfigurazio horri dagokion egoera hartu beharko da kontuan.

Definizioari jarraituz eta konfigurazio ez-deterministen bidez $\{r_2\}$ multzoa lortu da eta konfigurazio deterministen bidez zuhaitza garatuz ε hitza duen konfigurazio bakarrean r_2 egoera dugu. Ondorioz, a sinboloarekin r_2-tik r_2-ra doan trantsizioa gehitu beharko da.



- $\nu(r_2, b)$:

  - Definizioari jarraituz: $\nu(r_2, b) = \lambda^*(\{r_2\}, b) = \lambda^*(\lambda(r_2, b), \varepsilon) = \lambda^*(\varnothing, \varepsilon) =$ Itxidura-$\varepsilon^*(\varnothing, \lambda) = \varnothing$.

– Konfigurazio ez-deterministez osatutako sekuentzia gisa:

$$\begin{array}{c}(\{r_2\}, b) \equiv_{\mathsf{Itxidura}\text{-}\varepsilon^*} (\{r_2\}, b)\\ | \\ (\varnothing, \varepsilon) \equiv_{\mathsf{Itxidura}\text{-}\varepsilon^*} (\varnothing, \varepsilon)\end{array}$$

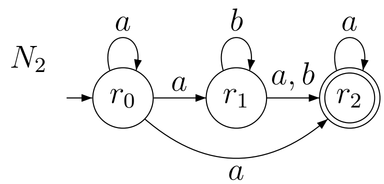
Sekuentzia horretako osagai bakoitzean ε trantsizioak kontuan hartuz, hau da, Itxidura-ε^* kontuan hartuz lortzen den konfigurazioa zein den zehazten da.

– Konfigurazio deterministez osatutako zuhaitz gisa:

$$(r_2, b)$$

Hitz bezala ε duen konfiguraziorik ez da lortu zuhaitz horretan.

Trantsiziorik ez da gehitu behar r_2-tik b-rekin.



- $\nu(r_2, c)$:

  – Definizioari jarraituz: $\nu(r_2, c) = \lambda^*(\{r_2\}, c) = \lambda^*(\lambda(r_2, c), \varepsilon) = \lambda^*(\varnothing, \varepsilon) = \mathsf{Itxidura}\text{-}\varepsilon^*(\varnothing, \lambda) = \varnothing$.

  – Konfigurazio ez-deterministez osatutako sekuentzia gisa:

$$\begin{array}{c}(\{r_2\}, c) \equiv_{\mathsf{Itxidura}\text{-}\varepsilon^*} (\{r_2\}, c)\\ | \\ (\varnothing, \varepsilon) \equiv_{\mathsf{Itxidura}\text{-}\varepsilon^*} (\varnothing, \varepsilon)\end{array}$$

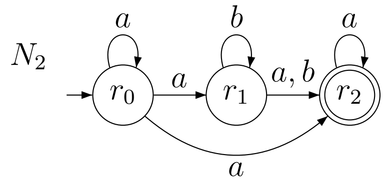
  Sekuentzia horretako osagai bakoitzean ε trantsizioak kontuan hartuz, hau da, Itxidura-ε^* kontuan hartuz lortzen den konfigurazioa zein den zehazten da.

  – Konfigurazio deterministez osatutako zuhaitz gisa:

$$(r_2, c)$$

  Zuhaitz honetan ere ez da lortu ε hitza duen konfiguraziorik.

Definizioari jarraituz eta konfigurazio ez-deministen bidez multzo hutsa lortu denez eta konfigurazio deterministen bidez zuhaitza garatuz ε hitza duen konfiguraziorik ez denez lortu, ez da gezirik gehitu behar r_2-tik c-rekin.



3.5.2 AFDen eta AFEDen arteko baliokidetasuna

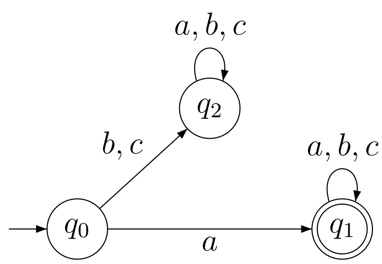
AFDak eta AFEDak baliokideak direla frogatzeko, bi gauza egin behar dira. Alde batetik, AFD bat emanda, AFD horrek definitzen duen lengoaia bera definitzen duen AFED bat diseina daitekeela frogatu behar da. Bestetik, AFED bat emanda, lengoaia bera definitzen duen AFD bat defini daitekeela frogatu behar da.

3.5.2.1 Edozein AFD emanda, lengoaia bera definitzen duen AFED bat eraiki daiteke

AFEDen grafikoak aztertzen baditugu, oro har, egoera batetik eta sinbolo berarentzat gezi bat baino gehiago egon daitezkeela ikus dezakegu. Baita ere gerta daiteke egoera batetik sinboloren batentzat gezirik ez ateratzea. Baina ezaugarri horiek ez dira derrigorrezkoak AFEDetan eta gerta daiteke egoera bakoitzeko eta sinbolo bakoitzeko gezi bakarra edukitzea. Hori horrela izanda, edozein AFD, grafikoki, AFED bat ere bada. Esate baterako, har dezagun $A = \{a, b, c\}$ alfabetoaren gainean definituta dauden eta a sinboloaz hasten diren hitzez eratutako lengoaiari dagokion honako D AFD hau:

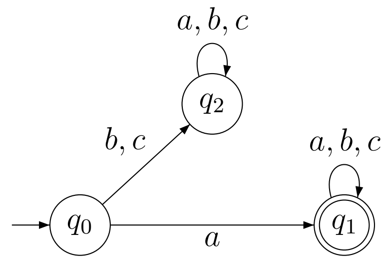
$$L_3 = \{w \mid w \in A^* \wedge \exists u(u \in A^* \wedge w = au)\}$$

D



Oraintxe esan den moduan, automata finitu hori AFED bat dela ere esan dezakegu, nahiz eta ez gertatu sinbolo batentzat egoera beretik gezi bat baino gehiago ateratzea edo sinboloren batentzat egoeraren batetik gezirik ez ateratzea.

N



D AFDa (Q, A, δ, q_0, Y) erako boskote bezala hartzen badugu, $\delta : Q \times A \to Q$ izanda, AFD horri dagokion N AFEDa (Q, A, ν, q_0, Y) erako boskote bat izango da, $\nu : Q \times A \to 2^Q$ izanda eta $\nu(q_j, \alpha) = \{\delta(q_j, \alpha)\}$ betez $q_j \in Q$ egoera denentzat eta A alfabetoko α sinbolo denentzat. Beraz, kasu berezi honetan, egoera bakoitzeko eta sinbolo bakoitzeko, ν funtzioak egoera bakar batez eratutako multzoa itzuliko du.

a sinboloaz hasten diren hitzez osatutako lengoaiari dagokion adibidera itzuliz, δ funtzioaren definizioa honako hau izango litzateke:

δ	a	b	c
q_0	q_1	q_2	q_2
q_1	q_1	q_1	q_1
q_2	q_2	q_2	q_2

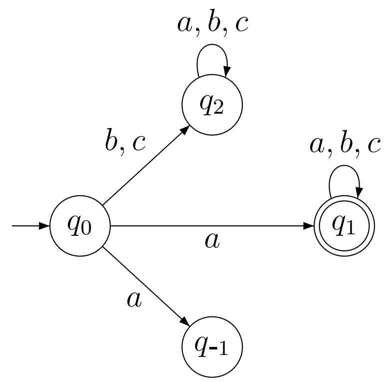
Aldiz, ν funtzioaren definizioa honako hau izango litzateke:

ν	a	b	c
q_0	$\{q_1\}$	$\{q_2\}$	$\{q_2\}$
q_1	$\{q_1\}$	$\{q_1\}$	$\{q_1\}$
q_2	$\{q_2\}$	$\{q_2\}$	$\{q_2\}$

Adibide honetako D AFDa eta N AFEDa grafikoki berdinak izan arren, δ eta ν definitzen dituzten taulak desberdinak dira eta definizio horietan garbi gelditzen da δ funtzioa AFD bati dagokion funtzioa dela eta ν funtzioa AFED bati dagokion funtzioa dela.

Dena den, AFD bat grafikoki emanda, dagokion AFEDa kalkulatzean gutxienez egoera batean eta sinbolo batentzat gezi bat baino gehiago edukitzea nahi badugu eta gutxienez egoera batean sinbolo batentzat gezirik ez egotea nahi badugu, hori era sistematikoan lortzeko balio duen metodo bat badugu. Horretarako nahikoa da q_{-1} egoera berri bat gehitzearekin eta q_0-tik q_{-1} egoerara gezi bat ipintzearekin edozein sinbolo erabiliz. Bestalde, q_{-1} egoeratik ez da gezirik ipini behar inora. Metodo hau jarraituz, a sinboloaz hasten diren hitzez osatutako lengoaiari dagokion D AFDari dagokion N' AFEDa honako hau izango litzateke:

N'



N' AFED-ari dagokion ν trantsizio-funtzioaren taula honako hau izango litzateke:

ν	a	b	c
q_0	$\{q_1, q_{-1}\}$	$\{q_2\}$	$\{q_2\}$
q_1	$\{q_1\}$	$\{q_1\}$	$\{q_1\}$
q_2	$\{q_2\}$	$\{q_2\}$	$\{q_2\}$
q_{-1}	$\varnothing$	$\varnothing$	$\varnothing$

3.5.2.2 Edozein AFED emanda, lengoaia bera definitzen duen AFD bat eraiki daiteke

(Q, A, ν, q_0, Y) egitura duen AFED bat emanda, dagokion AFDak $(Q', A', \delta, r_0, Y')$ egitura izango du eta egoera-multzoak kontsideratuz kalkulatuko da:

- A' alfabetoa A izango da.
- r_0 hasierako egoera $\{q_0\}$ etiketa izango duen Q' multzoko egoera izango da. Q' multzoko egoerak 2^Q multzoko elementuak izango dira. Beraz, Q' multzoko egoera bakoitza Q-ren azpimultzo bat izango da. $Q' \subseteq 2^Q$ beteko da.
- A alfabetoko α sinbolo bakoitzeko, $\nu(q_0, \alpha)$ kalkulatu beharko da. Hori $\nu^*(\{q_0\}, \alpha)$ kalkulatzearen baliokidea da.
- $\nu(q_0, \alpha)$ erako multzo bakoitza kalkulatzen ari garen AFDko egoera berri bat izango da eta $\{q_0\}$ (edo r_0) egoeratik gezi bat ipini beharko da $\nu(q_0, \alpha)$ egoerara α sinboloarekin. Beraz, $\delta(r_0, \alpha) = \nu(q_0, \alpha)$.
- A alfabetoko α sinbolo bakoitzeko, $\nu(q_0, \alpha)$ kalkulatuz lortu diren multzoak kontuan hartuz, egoera berriak sortzeko prozesua errepikatu beharko da egoera berriak sortzea posible den bitartean. $\{q_{j_1}, q_{j_2}, \ldots, q_{j_n}\}$ multzoa eraikitzen ari garen AFDko egoera bat baldin bada, orduan $\alpha \in A$ sinbolo bakoitzeko $\nu^*(\{q_{j_1}, q_{j_2}, \ldots, q_{j_n}\}, \alpha) = \nu(q_{j_1}, \alpha) \cup \nu(q_{j_2}, \alpha) \cup \cdots \cup \nu(q_{j_n}, \alpha)$ kalkulatu beharko da. Hortik egoera-multzo bat lortuko da eta egoera-multzo hori eraikitzen ari garen AFDko beste egoera bat izango da eta egoera

horretara α-rekin trantsizio bat ipini beharko da $\{q_{j_1}, q_{j_2}, \ldots, q_{j_n}\}$ egoeratik. Formalki, $\delta(\{q_{j_1}, q_{j_2}, \ldots, q_{j_n}\}, \alpha) = \nu^*(\{q_{j_1}, q_{j_2}, \ldots, q_{j_n}\}, \alpha) = \nu(q_{j_1}, \alpha) \cup \nu(q_{j_2}, \alpha) \cup \cdots \cup \nu(q_{j_n}, \alpha)$.

- Y'-ko egoerak, hau da, onarpen egoerak Y-ko egoeraren bat dutenak izango dira (nahikoa da Y-ko egoera bat edukitzea). Beraz, $\{q_{j_1}, q_{j_2}, \ldots, q_{j_n}\}$ multzoa Q'-ko egoera bat baldin bada eta $\{q_{j_1}, q_{j_2}, \ldots, q_{j_n}\} \cap Y \neq \varnothing$ betetzen bada, orduan $\{q_{j_1}, q_{j_2}, \ldots, q_{j_n}\}$ egoera Y'-koa izango da.

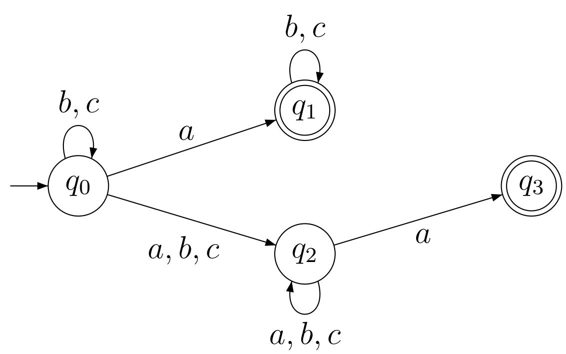
3.5.2.2.1 Lehenengo adibidea: AFED bati dagokion AFDaren kalkulua

3.5.1 irudian a bakarra duten hitzez (nahi adina b eta c edukiz) edo a sinboloaz bukatzen diren hitzez osatutako lengoaiari dagokion AFED bat erakusten da. Lengoaia horren definizio formala honako hau da:

$$\{w \mid w \in A^* \wedge (|w|_a = 1 \vee \exists u(u \in A^* \wedge w = ua))\}$$

$A = \{a, b, c\}$ izanda. Lengoaia hori definitzeko beste aukera bat honako hau izango litzateke:

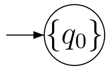
$$((\{b, c\}^*)\{a\}(\{b, c\}^*)) \cup (A^*\{a\})$$



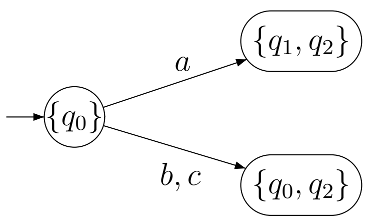
3.5.1 irudia. a bakarra duten edo a sinboloaz bukatzen diren hitzez osatutako lengoaiari dagokion AFEDa.

Jarraian, 3.5.1 irudiko AFEDari dagokion AFDa kalkulatuko da. AFD hori kalkulatzeko prozesuan egoera berriak sortuz joango gara eta egoera berriak sortzeko prozesu hori urratsez urrats erakutsiko da. AFDko egoera berri bakoitza AFEDko egoera-multzo batez etiketatuta egongo da eta AFDa sortzeko prozesua bukatu ondoren egoerak berrizendatu egingo dira. Urratsak honako hauek dira:

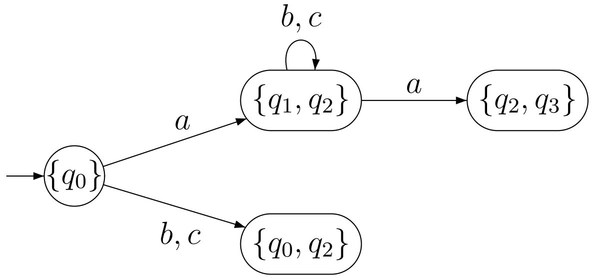
- Hasierako egoera beti abiapuntuko AFEDan hasierako egoera den egoeraz osatutako multzoa izango da. Kasu honetan $\{q_0\}$ izango da. Beraz, hasteko, $\{q_0\}$ etiketa duen egoera bat sortuko da, hasierako egoerek izan ohi duten gezi berezia ere ipiniz.



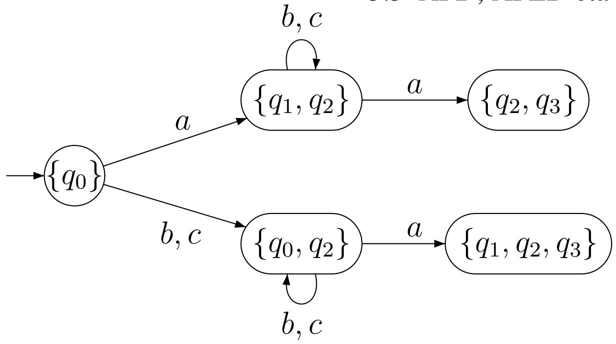
- $\nu^*(\{q_0\}, a) = \nu(q_0, a) = \{q_1, q_2\}$, $\nu^*(\{q_0\}, b) = \nu(q_0, b) = \{q_0, q_2\}$ eta $\nu^*(\{q_0\}, c) = \nu(q_0, c) = \{q_0, q_2\}$ betetzen denez, bi egoera berri gehitu behar dira, $\{q_1, q_2\}$ eta $\{q_0, q_2\}$, eta $\{q_0\}$ egoeratik dagozkien geziak ipini behar dira:



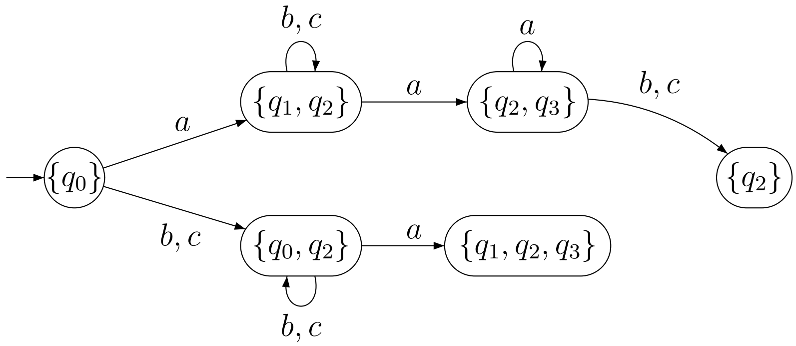
- Orain $\{q_1, q_2\}$ egoera hartuko dugu eta, alde batetik, $\nu^*(\{q_1, q_2\}, a) = \nu(q_1, a) \cup \nu(q_2, a) = \varnothing \cup \{q_2, q_3\} = \{q_2, q_3\}$ betetzen da. Beste aldetik, $\nu^*(\{q_1, q_2\}, b) = \nu(q_1, b) \cup \nu(q_2, b) = \{q_1\} \cup \{q_2\} = \{q_1, q_2\}$ betetzen da. Bukatzeko, $\nu^*(\{q_1, q_2\}, c) = \nu(q_1, c) \cup \nu(q_2, c) = \{q_1\} \cup \{q_2\} = \{q_1, q_2\}$ betetzen da. Beraz, egoera berri bat, $\{q_2, q_3\}$, eta hiru trantsizio berri gehitu beharko dira:



- Orain $\{q_0, q_2\}$ egoera hartuko dugu. Alde batetik, $\nu^*(\{q_0, q_2\}, a) = \nu(q_0, a) \cup \nu(q_2, a) = \{q_1, q_2\} \cup \{q_2, q_3\} = \{q_1, q_2, q_3\}$ betetzen da. Bestetik, $\nu^*(\{q_0, q_2\}, b) = \nu(q_0, b) \cup \nu(q_2, b) = \{q_0, q_2\} \cup \{q_2\} = \{q_0, q_2\}$ betetzen da. Eta hirugarren lekuan, $\nu^*(\{q_0, q_2\}, c) = \nu(q_0, c) \cup \nu(q_2, c) = \{q_0, q_2\} \cup \{q_2\} = \{q_0, q_2\}$ betetzen da. Ondorioz, $\{q_1, q_2, q_3\}$ egoera berria eta beste hiru trantsizio gehitu beharko dira:



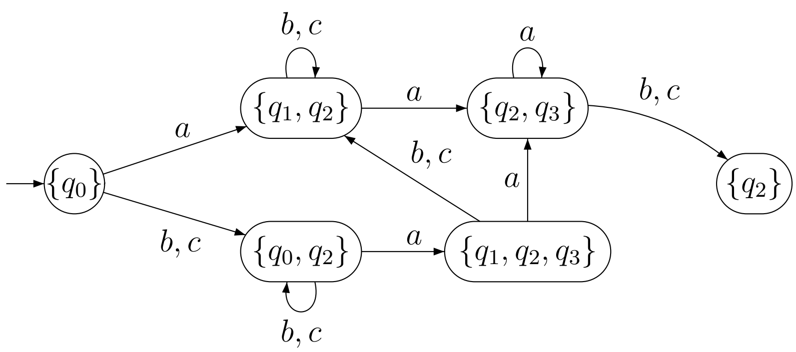
- Jarraian, $\{q_2, q_3\}$ egoera kontsideratuko dugu. Hasteko, $\nu^*(\{q_2, q_3\}, a) = \nu(q_2, a) \cup \nu(q_3, a) = \{q_2, q_3\} \cup \varnothing = \{q_2, q_3\}$ betetzen dela ikus dezakegu. Gero, $\nu^*(\{q_2, q_3\}, b) = \nu(q_2, b) \cup \nu(q_3, b) = \{q_2\} \cup \varnothing = \{q_2\}$ betetzen dela daukagu. Azkenik, $\nu^*(\{q_2, q_3\}, c) = \nu(q_2, c) \cup \nu(q_3, c) = \{q_2\} \cup \varnothing = \{q_2\}$ betezen da. Guztira, $\{q_2\}$ egoera berria eta hiru trantsizio gehitu beharko dira:



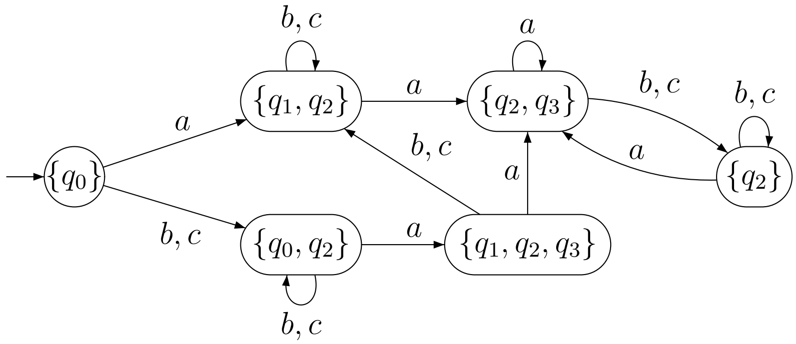
- Orain $\{q_1, q_2, q_3\}$ egoerari dagozkion trantsizioak kalkulatuko ditugu. Alde batetik,

$$\nu^*(\{q_1, q_2, q_3\}, a) = \nu(q_1, a) \cup \nu(q_2, a) \cup \nu(q_3, a) = \varnothing \cup \{q_2, q_3\} \cup \varnothing = \{q_2, q_3\}$$

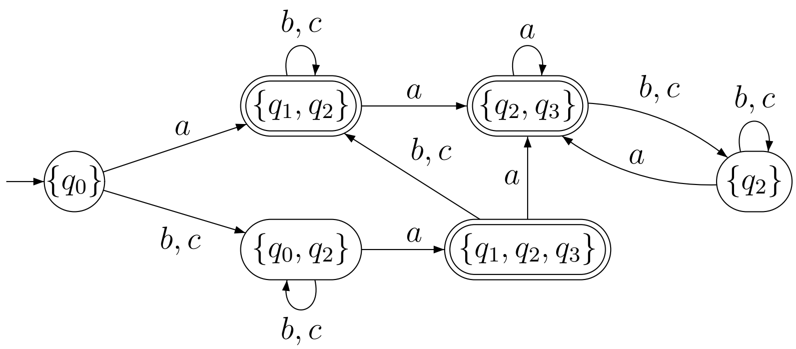
  betetzen da. Bestetik, $\nu^*(\{q_1, q_2, q_3\}, b) = \nu(q_1, b) \cup \nu(q_2, b) \cup \nu(q_3, b) = \{q_1\} \cup \{q_2\} \cup \varnothing = \{q_1, q_2\}$ betetzen da. Azkenik, $\nu^*(\{q_1, q_2, q_3\}, c) = \nu(q_1, c) \cup \nu(q_2, c) \cup \nu(q_3, c) = \{q_1\} \cup \{q_2\} \cup \varnothing = \{q_1, q_2\}$ betetzen da. Beraz, ez dago egoera berririk baina hiru trantsizio berri gehitu behar dira:



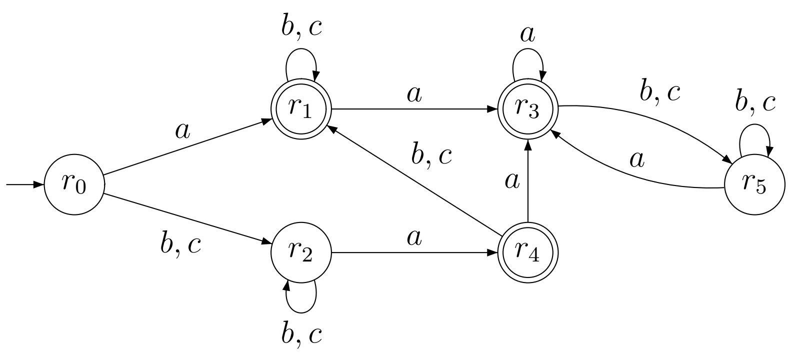
- $\{q_2\}$ egoeraren trantsizioak kalkulatu behar dira orain. Alde batetik, $\nu^*(\{q_2\}, a) = \nu(q_2, a) = \{q_2, q_3\}$ betetzen da. Bestetik, $\nu^*(\{q_2\}, b) = \nu(q_2, b) = \{q_2\}$ eta $\nu^*(\{q_2\}, c) = \nu(q_2, c) = \{q_2\}$.



- Egoera eta trantsizio denak kalkulatu direnez, zirkulu bikoitzeko egoerak zein izango diren erabakitzea gelditzen da. Hasierako AFEDan zirkulu bikoitza duen egoeraren bat duen AFDko egoera bakoitzari zirkulu bikoitza ipini behar zaio. Beraz, $\{q_1, q_2\}$ egoera q_1 duelako, $\{q_2, q_3\}$ egoera q_3 duelako eta $\{q_1, q_2, q_3\}$ egoera q_1 eta q_3 dituelako.

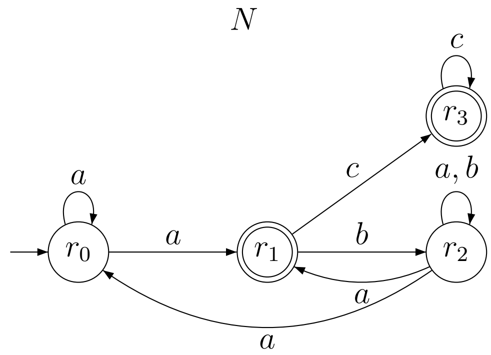


- Bukatzeko, egoera bakoitza honako era honetara berrizendatuko dugu: $r_0 = \{q_0\}$, $r_1 = \{q_1, q_2\}$, $r_2 = \{q_0, q_2\}$, $r_3 = \{q_2, q_3\}$, $r_4 = \{q_1, q_2, q_3\}$ eta $r_5 = \{q_2\}$. AFDa honela geldituko zaigu:



3.5.2.2.2 Bigarren adibidea: AFED bati dagokion AFD baten kalkulua

3.5.2 irudian $A = \{a, b, c\}$ alfabetoaren gainean definitutako lengoaia bati dagokion AFEDaren trantsizio-diagrama dugu. Lengoaia hori formalki honela defini dezakegu:

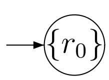


3.5.2 irudia. $(\{a\}(\{b\}(\{a\} \cup \{b\})^*\{a\})^*)^+\{c\}^*$ lengoaiari dagokion N AFEDa.

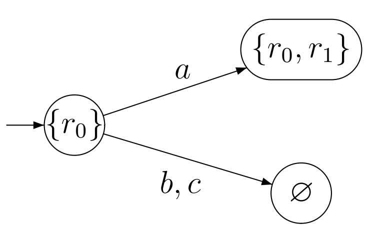
$$(\{a\}(\{b\}(\{a\} \cup \{b\})^*\{a\})^*)^+\{c\}^*$$

Orain 3.5.2 irudiko N AFEDari dagokion AFDa kalkulatuko dugu. Urratsez urrats egingo da eta urrats bakoitzean sortzen diren egoerak eta trantsizioak zein diren erakutsiko da. Eraikiko den AFDko egoera bakoitzaren etiketa AFEDko egoerez osatutako multzo bat izango denez, bukeran AFDko egoerak berrizendatu egingo ditugu. Hona hemen urratsak:

- AFDko hasierako egoeraren etiketa AFEDko hasierako egoeraz osatutako multzoa izango da. Kasu honetan $\{r_0\}$. Beraz, $\{r_0\}$ etiketa duen egoera bat sortuko dugu eta hasierako egoerek izan ohi duten gezi berezia ipiniko diogu.

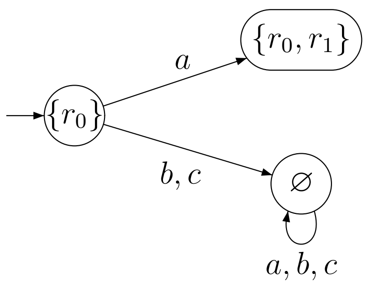


- Orain $\{r_0\}$ egoeratik ateratzen diren trantsizioak kalkulatuko ditugu. Alde batetik, $\nu^*(\{r_0\}, a) = \nu(r_0, a) = \{r_0, r_1\}$ daukagu. Bestetik, $\nu^*(\{r_0\}, b) = \nu(r_0, b) = \varnothing$. Azkenik, $\nu^*(\{r_0\}, c) = \nu(r_0, c) = \varnothing$.

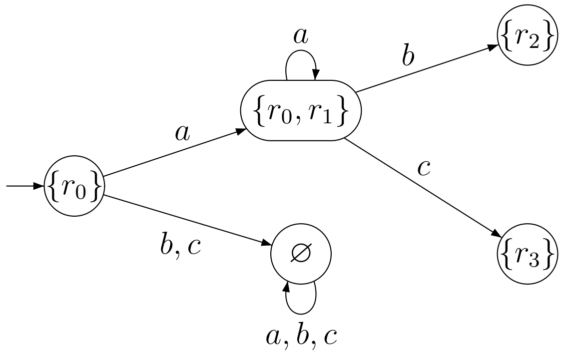


Multzo hutsa egoera baten etiketa izan daitekeela ikus dezakegu hor.

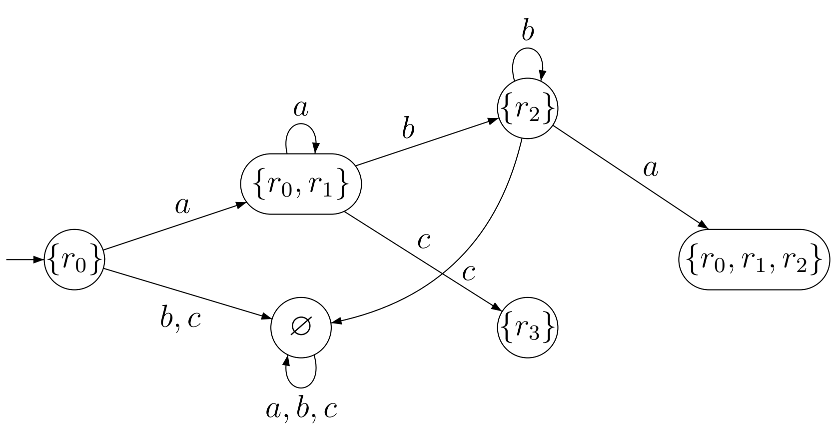
- Orain $\varnothing$ egoeratik ateratzen diren trantsizioak kalkulatuko ditugu. Multzo hutsean N-ko egoerarik ez dagoenez, multzo hutseko egoeretatik ezingo dugu beste egoera batzuetara joan. Ondorioz, bai a-rekin, bai b-rekin eta bai c-rekin $\varnothing$ egoeratik $\varnothing$ egoerara joango gara. Formalki, $\nu^*(\varnothing, a) = \varnothing$, $\nu^*(\varnothing, b) = \varnothing$ eta $\nu^*(\varnothing, c) = \varnothing$.



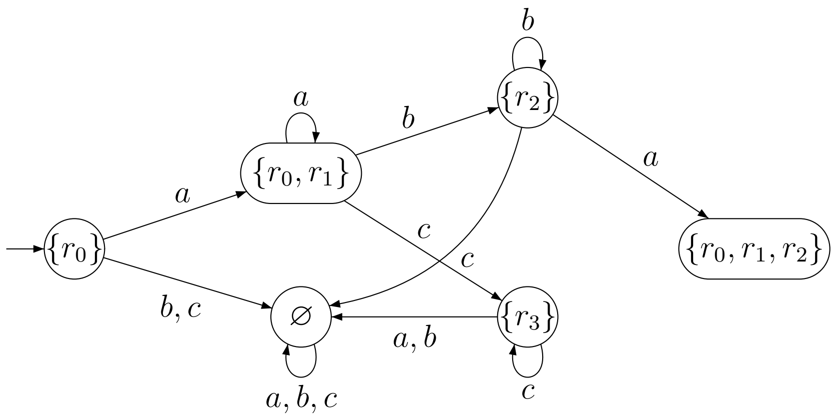
- Orain $\{r_0, r_1\}$ egoeratik aterako diren trantsizioen kalkulua dator. Alde batetik, $\nu^*(\{r_0, r_1\}, a) = \nu(r_0, a) \cup \nu(r_1, a) = \{r_0, r_1\} \cup \varnothing$, hau da, $\{r_0, r_1\}$. Bestetik, $\nu^*(\{r_0, r_1\}, b) = \nu(r_0, b) \cup \nu(r_1, b) = \varnothing \cup \{r_2\}$, hau da, $\{r_2\}$. Eta c sinboloari dagokionez, $\nu^*(\{r_0, r_1\}, c) = \nu(r_0, c) \cup \nu(r_1, c) = \varnothing \cup \{r_3\}$, hau da, $\{r_3\}$



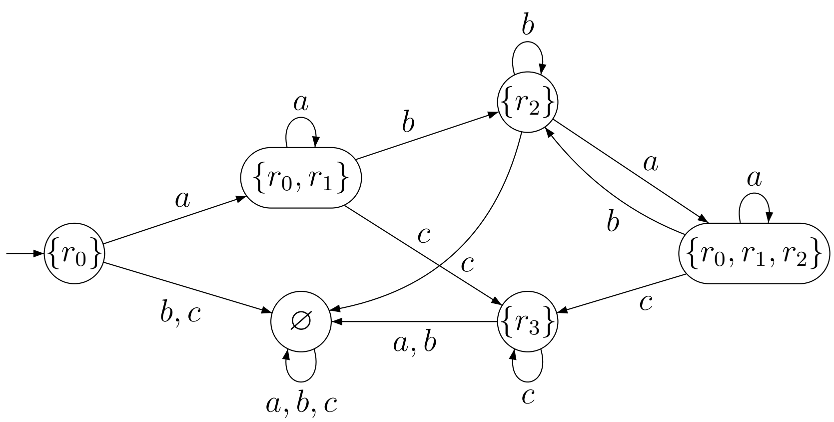
- $\{r_2\}$ egoeratik aterako diren trantsizioen kalkulua dator orain. Hasteko, a sinboloari dagokionez, $\nu^*(\{r_2\}, a) = \nu(r_2, a) = \{r_0, r_1, r_2\}$. Bestetik, b-ri dagokionez, $\nu^*(\{r_2\}, b) = \nu(r_2, b) = \{r_2\}$. Hirugarren lekuan, c-ri dagokionez, $\nu^*(\{r_2\}, c) = \nu(r_2, c) = \varnothing$.



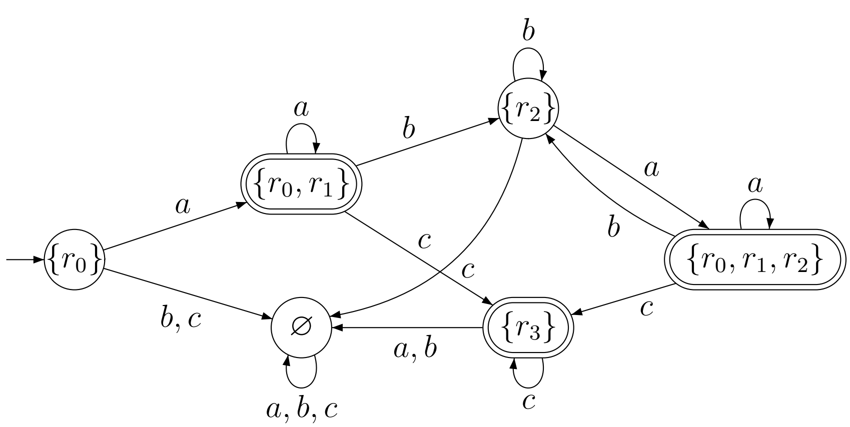
- Orain $\{r_3\}$ egoeratik aterako diren trantsizioak kalkulatuko dira. Alde batetik, $\nu^*(\{r_3\}, a) = \nu(r_3, a) = \varnothing$. Bestetik, $\nu^*(\{r_3\}, b) = \nu(r_3, b) = \varnothing$. Bukatzeko, $\nu^*(\{r_3\}, c) = \nu(r_3, c) = \{r_3\}$.



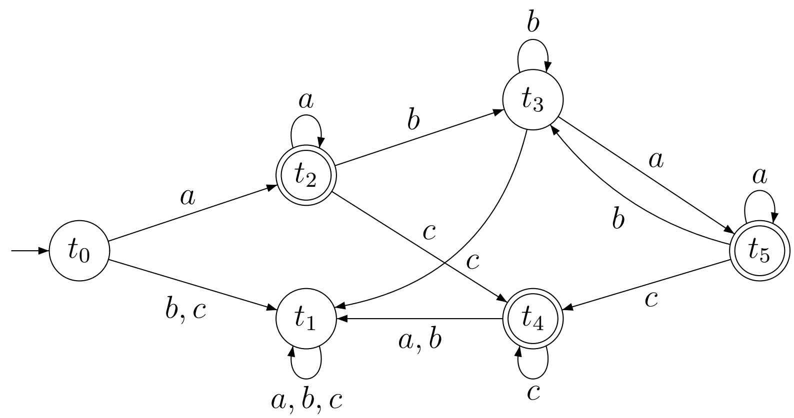
- $\{r_0, r_1, r_2\}$ etiketa duen egoeratik aterako diren trantsizioak datoz orain. Hasteko, a-ri dagokionez, $\nu^*(\{r_0, r_1, r_2\}, a) = \nu(r_0, a) \cup \nu(r_1, a) \cup \nu(r_2, a) = \{r_0, r_1\} \cup \varnothing \cup \{r_0, r_1, r_2\}$, hau da, $\{r_0, r_1, r_2\}$. Bestetik, b-ri dagokionez, $\nu^*(\{r_0, r_1, r_2\}, b) = \nu(r_0, b) \cup \nu(r_1, b) \cup \nu(r_2, b) = \varnothing \cup \{r_2\} \cup \{r_2\}$, hau da, $\{r_2\}$. Eta c-ri dagokionez, $\nu^*(\{r_0, r_1, r_2\}, c) = \nu(r_0, c) \cup \nu(r_1, c) \cup \nu(r_2, c) = \varnothing \cup \{r_3\} \cup \varnothing$, hau da, $\{r_3\}$.



- Trantsizio denak kalkulatuta daudenez, zirkulu bikoitza izango duten egoerak zein diren zehaztea falta da. Abiapuntuko AFEDko onarpen egoeraren bat duten AFDko egoerak izango dira zirkulu bikoitza izango dutenak. Beraz, $\{r_3\}$ egoera r_3 duelako eta $\{r_0, r_1\}$ eta $\{r_0, r_1, r_2\}$ egoerak r_1 dutelako.



- Bukatzeko AFDko egoerak berrizendatu egingo ditugu: $t_0 = \{r_0\}$, $t_1 = \varnothing$, $t_2 = \{r_0, r_1\}$, $t_3 = \{r_2\}$, $t_4 = \{r_3\}$ eta $t_5 = \{r_0, r_1, r_2\}$. Honako AFD hau gelditzen zaigu beraz:



3.6.

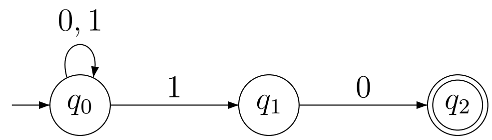
Automata finitu bati (AFD, AFED edo ε-AFED bati) dagokion lengoaiaren kalkulua

Automata finitu bat emanda (AFDa, AFEDa edo ε-AFEDa izan daitekeena), automata finitu horrek definitutako lengoaia era sistematikoan kalkulatzeko balio duen metodoa aurkeztuko da jarraian. Automata finituak AF bezala laburtuko ditugu. Automata finituen barnean AFDak, AFEDak eta ε-AFEDak sartzen direnez, metodoak hiru automa mota horientzat balio du.

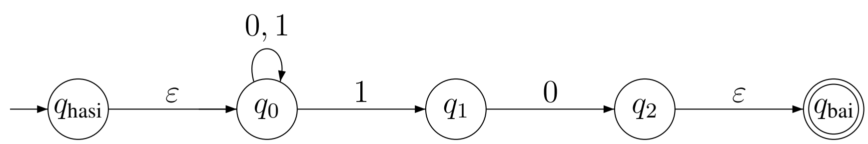
Metodoa aplikatzean ikusiko den moduan, ideia oinarrizko lengoaiak erabiltzea da. Alfabetoa $A = \{a, b, c\}$ baldin bada, oinarrizko lengoaiak $\varnothing$ lengoaia hutsa, $\{\varepsilon\}$ hitz hutsa bakarrik duen lengoaia eta $\{a\}$, $\{b\}$ eta $\{c\}$ dira. Giltzak erabiltzeak irakurgarritasuna zailtzen duenez, ez ditugu erabiliko. Adibidez, $\{a\}$ idatzi beharrean bakarrik a idatziko dugu eta $\{\varepsilon\}$ idatzi beharrean ε idatziko dugu. Irakurgarritasuna oraindik gehiago hobetzeko, lengoaien arteko bilketa adierazteko $+$ sinboloa erabiliko dugu $\cup$ idatzi beharrean. Esate baterako, $\{a\} \cup \{b\}$ idatzi beharrean $a + b$ idatziko dugu, $\{a\}\{b\}$ idatzi beharrean a^* idatziko dugu eta $(\{a\}\{a\}(\{b\}^*)) \cup ((\{b\}\{b\})^*)$ idatzi beharrean $(aa(b^*)) + ((bb)^*)$ idatziko dugu. Gainera, kasu askotan parentesien erabilera ere saihestu daiteke lehentasun handieneko eragilea $*$ dela, bigarren lehentasun handiena duena hitzen arteko kateaketa dela eta hirugarren mailan $+$ dagoela, hau da, bilketa dagoela kontuan hartuz. Irizpide horri jarraituz, $(aa(b^*)) + ((bb)^*)$ idatzi beharrean $aab^* + (bb)^*$ idatz dezakegu.

3.6.1 AF bateko egoerak ezabatuz AFari dagokion lengoaia kalkulatzen duen metodoa

Metodoa aplikatzeko jarraitu behar diren urratsak azaltzeko, adibide bat erabiliko dugu. Demagun honako AF hau daukagula:

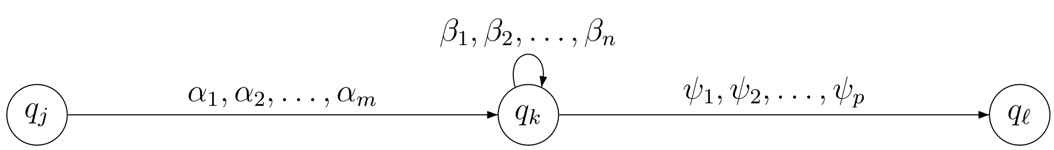


Lehenengo urratsa q_{hasi} eta q_{bai} izeneko bi egoera berri gehitzea da. Hasierako egoera berria q_{hasi} izango da eta zirkulu bikoitza izango duen egoera bakarra q_{bai} izango da. Beraz, q_{hasi} egoeratik q_0 egoerara ε sinboloa duen trantsizio bat ipini beharko da eta hasierako AFan zirkulu bikoitza duen egoera bakoitzetik ere ε sinboloa duen trantsizio bat ipini beharko da q_{bai} egoerara.

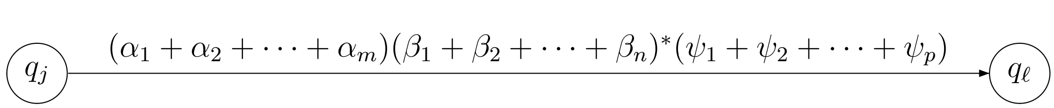


Jarraian, q_0, q_1 eta q_2 egoerak ezabatuz joan behar dugu banan-banan. Berez, berdin da zein ordenatan ezabatzen diren egoerak. Aipatu beharrekoa da aukeratutako ordenaren arabera emaitza gisa espresio desberdina lor daitekeela baina, edozein kasutan ere emaitza zuzena lortuko da. Egoera batzuk ezabatzea beste batzuk ezabatzea baino errazagoa izan daiteke eta, oro har, errazenak ezabatuz hastea da onena. Metodoaren oinarrizko ardatza honako hau da: q_k egoera bat ezabatzean q_k egoera horretatik igarotzen diren bide denak mantendu behar dira. Metodo horretan lau kasu bereiz ditzakegu. Lehenengoa kasu orokorra da eta beste hirurak kasu partikularrak dira.

1. Kasu orokorra honako hau da:

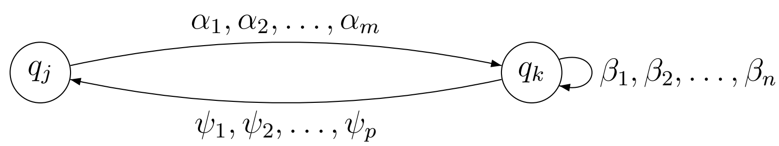


q_k egoera ezabatzean q_k egoera horretatik igarotzen diren bide denak mantendu behar dira. Honako hau geldituko litzaiguke:

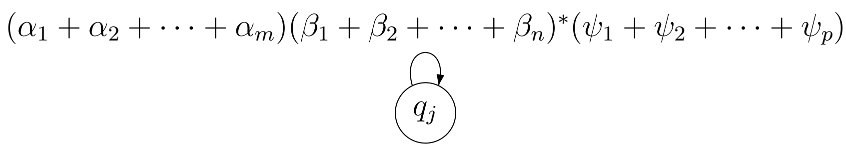


q_k egoeratik igaro eta q_j eta q_ℓ beste bi egoera lotzen dituen bide bakoitzeko eskema hori aplikatu beharko da.

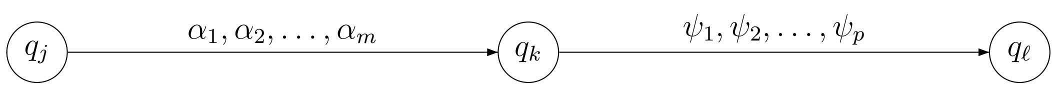
2. q_j eta q_ℓ egoera bera direnean, hau da, honako egitura hau daukagunean:



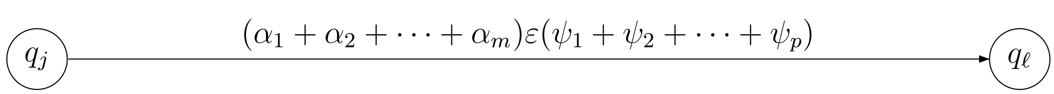
q_k egoera ezabatutakoan honako hau lortuko litzateke:



3. β-rik ez badago, hau da, $n = 0$ baldin bada,

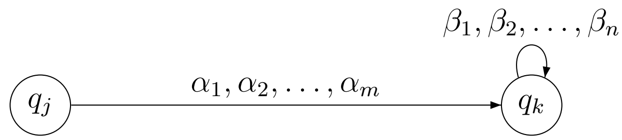


$\beta_1 + \beta_2 + \cdots + \beta_n$ batura edo bildura hutsa izango da eta hori $\varnothing$ da. Baina $\varnothing^*$ espresioa $\{\varepsilon\}$ denez, honako hau geldituko da:

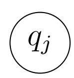


$(\alpha_1 + \alpha_2 + \cdots + \alpha_m)\varepsilon(\psi_1 + \psi_2 + \cdots + \psi_p)$ eta $(\alpha_1 + \alpha_2 + \cdots + \alpha_m)(\psi_1 + \psi_2 + \cdots + \psi_p)$ gauza bera direnez, ε ken daiteke.

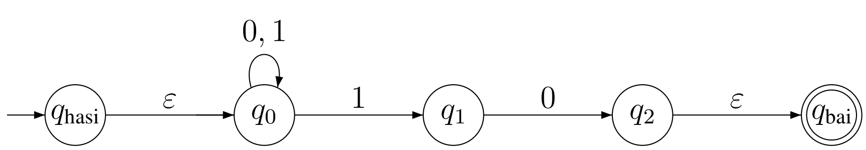
4. Azkenik, ψ-rik ez badago, hau da, $p = 0$ baldin bada, q_k-tik ezingo da beste egoera batera joan:



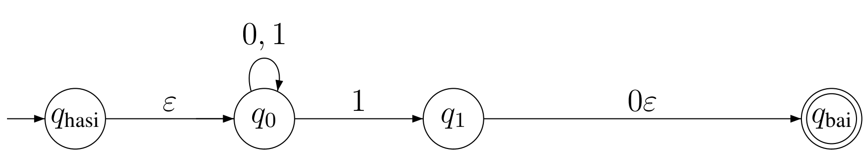
$\psi_1 + \psi_2 + \cdots + \psi_p$ batura edo bildura hutsa denez, $\varnothing$ da eta guztira $(\alpha_1 + \alpha_2 + \cdots + \alpha_m)(\beta_1 + \beta_2 + \cdots + \beta_p)^*\varnothing$ geldit确uko zaigu, baina hori dena $\varnothing$ da. Beraz, kasu honetan q_k ezabatu egin behar da besterik gabe. Beste era batera esanda, q_k-tik ez denez beste egoera batera doan biderik igarotzen, q_k ezabatzean ez da bertatik igarotzen den biderik mantendu behar, ez dagoelako horrelako biderik:



Eskema hauek azaldu aurretik hasi dugun adibidera itzuliko gara orain:

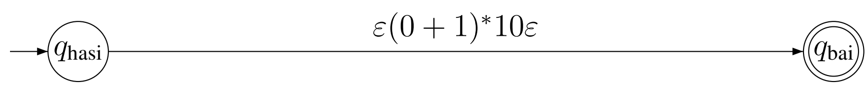


Hasteko q_2 ezabatuko dugu. Kasu honetan α bakarra eta ψ bakarra ditugu ($\alpha_1 = 0$, $\psi_1 = \varepsilon$) eta ez dago β-rik. Honako hau gelditasuko zaigu:



Orain q_1 ezabatuko dugu. Kasu honetan ere α bakarra eta ψ bakarra ditugu ($\alpha_1 = 1$, $\psi_1 = 0\varepsilon$) eta ez dago β-rik. Honako hau gelditasuko zaigu:

Azkenik, q_0 ezabatuko dugu. Orain ere α bakarra eta ψ bakarra ditugu ($\alpha_1 = \varepsilon$, $\psi_1 = 10\varepsilon$) baina bi β daude ($\beta_1 = 0$, $\beta_2 = 1$). Honako hau lortuko dugu:

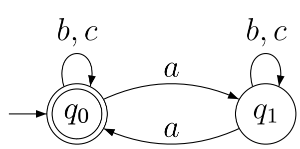


Beraz, abiapuntutzat hartu dugun automataren bidez definitutako lengoaia $\varepsilon(0+1)^*10\varepsilon$ da. Kateatuta dauden ε sinboloak ezabatuz [1] espresioa sinplifikatzen badugu, $(0+1)^*10$ geldituko zaigu. Gogora dezagun espresio horrek $(\{0\} \cup \ \{1\})^*\{1\}\{0\}$ lengoaia adierazten duela. Lengoaia hori 10 katearekin bukatzen diren hitzez eratutako lengoaia da.

3.6.2 AF bati dagokion lengoaiaren kalkuluaren adibideak

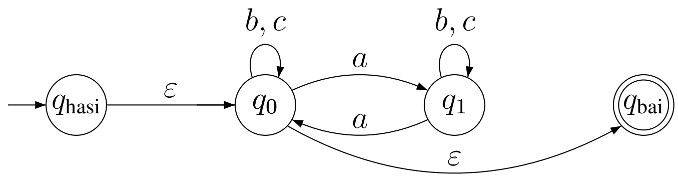
3.6.2.1 Adibidea: a kopuru bikoitia duten hitzez osatutako lengoaiari dagokion AFa

Honako AF honen bidez definitutako lengoaiari dagokion espresioa lortu nahi da:



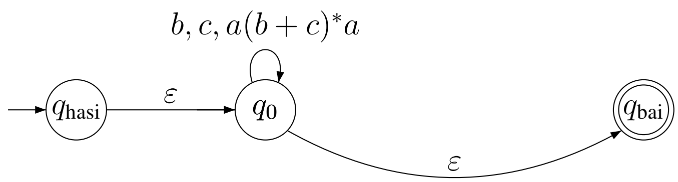
Lehenengo urratsa q_{hasi} eta q_{bai} izeneko bi egoera berri gehitzea da. Hasierako egoera berria q_{hasi} izango da eta zirkulu bikoitza izango duen egoera bakarra q_{bai} izango da. Beraz, q_{hasi} egoeratik q_0 egoerara ε sinboloa duen trantsizio bat ipini beharko da eta hasierako AFan zirkulu bikoitza duen egoera bakoitzetik ere ε sinboloa duen trantsizio bat ipini beharko da q_{bai} egoerara.

[1] $\varepsilon\delta \equiv \delta$ baina $\varepsilon + \delta \not\equiv \delta$, beraz, $\varepsilon\delta$ sinplifika daiteke baina $\varepsilon + \delta$ ezin da sinplifikatu.

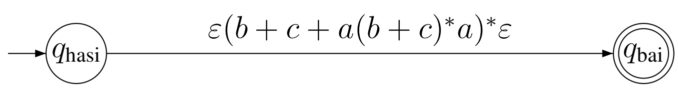


Jarraian, q_0 eta q_1 egoerak ezabatuz joan behar dugu banan-banan. Berez, berdin da zein ordenatan ezabatzen diren egoerak. Aipatu beharrekoa da aukeratutako ordenaren arabera emaitza gisa espresio desberdina lor daitekeela baina, edozein kasutan ere emaitza zuzena lortuko da. Egoera batzuk ezabatzea beste batzuk ezabatzea baino errazagoa izan daiteke eta, oro har, errazenak ezabatuz hastea da onena.

Lehenengo q_1 egoera ezabatuko dugu. Kasu honetan α bakarra eta ψ bakarra ditugu ($\alpha_1 = a$, $\psi_1 = a$) eta, gainera, bi β daude ($\beta_1 = b$, $\beta_2 = c$). Honako hau geldituko zaigu:



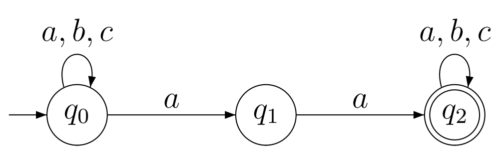
Orain q_0 ezabatuko da. Kasu honetan alde batetik α bakarra eta ψ bakarra ditugu ($\alpha_1 = \varepsilon$, $\psi_1 = \varepsilon$) eta beste aldetik hiru β ditugu ($\beta_1 = b$, $\beta_2 = c$, $\beta_3 = a(b+c)^*a$). Honako hau geldituko zaigu q_0 ezabatutakoan:



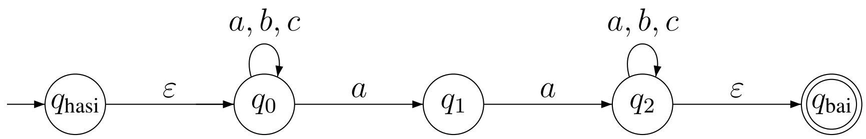
$\varepsilon(b+c+a(b+c)^*a)^*\varepsilon$ espresioa kateatuta dauden ε sinboloaren agerpenak kenduz sinplifika daiteke. Beraz, $(b + c + a(b + c)^*a)^*$ da adibide honetako AFak definitzen duen lengoaiari dagokion espresioa.

3.6.2.2 Adibidea: aa katea dutzen hitzez osatutako lengoaiari dagokion AFa

Demagun honako AF hau dugula:

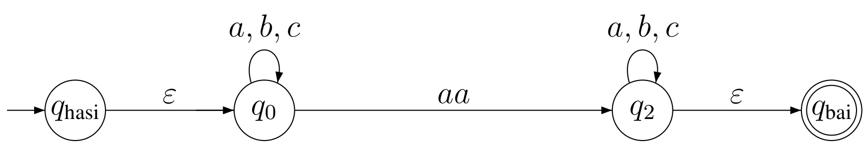


Lehenengo urratsa q_{hasi} eta q_{bai} izeneko bi egoera berri gehitzea da. Hasierako egoera berria q_{hasi} izango da eta zirkulu bikoitza izango duen egoera bakarra q_{bai} izango da. Beraz, q_{hasi} egoeratik q_0 egoerara ε sinboloa duen trantsizio bat ipini beharko da eta hasierako AFan zirkulu bikoitza duen egoera bakoitzetik ere ε sinboloa duen trantsizio bat ipini beharko da q_{bai} egoerara.

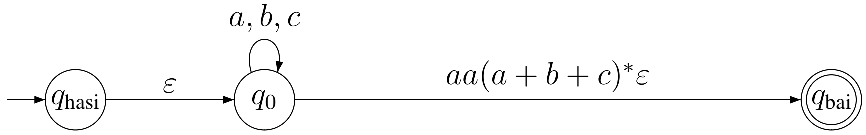


Jarraian, q_0, q_1 eta q_2 egoerak ezabatuz joan behar dugu banan-banan. Berez, berdin da zein ordenatan ezabatzen diren egoerak. Aipatu beharrekoa da aukeratutako ordenaren arabera emaitza gisa espresio desberdina lor daitekeela baina, edozein kasutan ere emaitza zuzena lortuko da. Egoera batzuk ezabatzea beste batzuk ezabatzea baino errazagoa izan daiteke eta, oro har, errazenak ezabatuz hastea da onena. Metodoaren oinarrizko ardatza honako hau da: q_k egoera bat ezabatzean q_k egoera horretatik igarotzen diren bide denak mantendu behar dira.

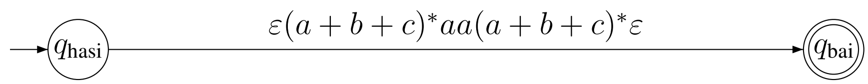
q_1 ezabatuz hasiko gara. Kasu honetan α bakarra eta ψ bakarra ditugu ($\alpha_1 = a$, $\psi_1 = a$) eta ez dago β-rik. Honako hau geldituko zaigu:



Jarraian, q_2 ezabatuko dugu. Kasu honetan ere α bakarra eta ψ bakarra ditugu ($\alpha_1 = aa$, $\psi_1 = \varepsilon$) baina hiru β daude ($\beta_1 = a$, $\beta_2 = b$, $\beta_3 = c$). honako hau geldituko zaigu q_2 ezabatu ondoren:



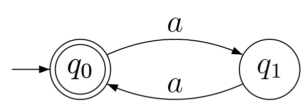
Bukatzeko, q_0 ezabatuko dugu. Oraingoan ere alde batetik α bat eta ψ bat ditugu ($\alpha_1 = \varepsilon$, $\psi_1 = aa(a+b+c)^*\varepsilon$) eta bestetik hiru β ditugu ($\beta_1 = a$, $\beta_2 = b$, $\beta_3 = c$). Honako hau lortuko da q_0 ezabatuz:



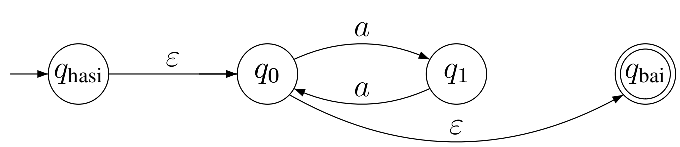
Beraz, adibide honetako automata finituak definitzen duen lengoaiari dagokion espresioa $\varepsilon(a+b+c)^*aa(a+b+c)^*\varepsilon$ da. Espresio hori sinplifika daiteke kateatuta agertzen diren ε sinboloak kenduz. Sinplifikatutako espresioa $(a+b+c)^*aa(a+b+c)^*$ izango litzateke. Gogora dezagun espresio horrek $(\{a\} \cup \{b\} \cup \{c\})^*\{a\}\{a\}(\{a\} \cup \{b\} \cup \{c\})^*$ lengoaia adierazten duela. Lengoaia hori aa katea duten hitzez eratutako lengoaia da.

3.6.2.3 Adibidea: b-rik eta c-rik ez eta a kopuru bikoitia duten hitzez osatutako lengoaiari dagokion AFa

Honako AF honen bidez definitutako lengoaiari dagokion espresioa lortu nahi da:

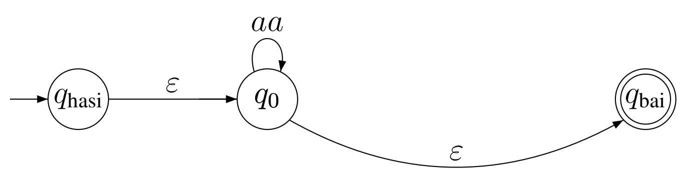


Lehenengo urratsa q_{hasi} eta q_{bai} izeneko bi egoera berri gehitzea da. Hasierako egoera berria q_{hasi} izango da eta zirkulu bikoitza izango duen egoera bakarra q_{bai} izango da. Beraz, q_{hasi} egoeratik q_0 egoerara ε sinboloa duen trantsizio bat ipini beharko da eta hasierako AFan zirkulu bikoitza duen egoera bakoitzetik ere ε sinboloa duen trantsizio bat ipini beharko da q_{bai} egoerara.

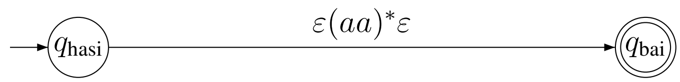


Jarraian, q_0 eta q_1 egoerak ezabatuz joan behar dugu banan-banan. Berez, berdin da zein ordenatan ezabatzen diren egoerak. Hala ere, aukeratutako ordenaren arabera emaitza bezala espresio desberdina lor daiteke baina, edozein kasutan ere emaitza zuzena lortuko da. Egoera batzuk ezabatzea beste batzuk ezabatzea baino errazagoa izan daiteke eta, oro har, errazenak ezabatuz hastea da onena. Metodoaren oinarrizko ardatza honako hau da: q_k egoera bat ezabatzean q_k egoera horretatik igarotzen diren bide denak mantendu behar dira.

Hasteko q_1 egoera ezabatuko da. Kasu honetan α bat eta ψ bat ditugu ($\alpha_1 = a$, $\psi_1 = a$) eta ez dago β-rik. Honako hau gelditzuko zaigu:



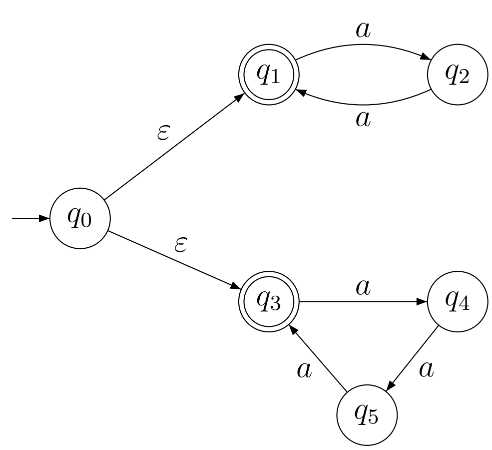
Metodoarekin jarraituz, orain q_0 ezabatuko dugu. Kasu honetan α bat eta ψ bat ditugu ($\alpha_1 = \varepsilon$, $\psi_1 = \varepsilon$) eta β ere bakarra da ($\beta_1 = aa$). Honako hau lortuko dugu q_0 ezabatu ondoren:



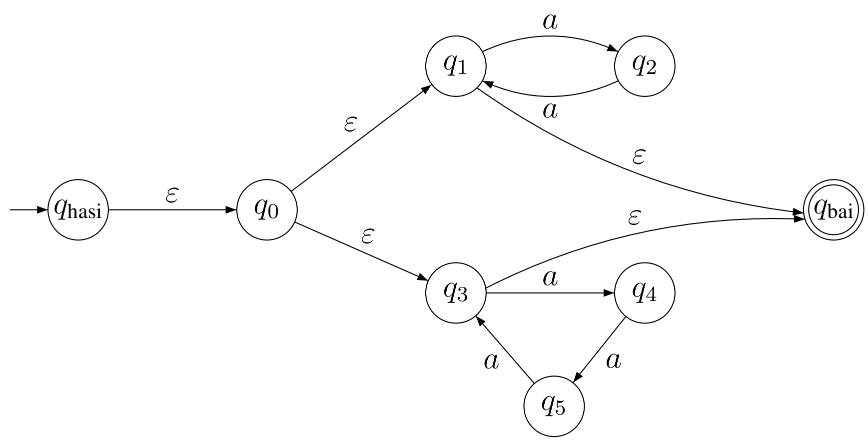
$\varepsilon(aa)^*\varepsilon$ espresioa kateatuta dauden ε sinboloaren agerpenak kenduz sinplifika daiteke. Ondorioz, $(aa)^*$ da abiapuntutzat hartu dugun AFaren bidez definitutako lengoaiari dagokion espresioa. Gogora dezagun $(aa)^*$ espresioak $(\{a\}\{a\})^*$ lengoaia adierazten duela.

3.6.2.4 Adibidea: aa katearen errepikapenez edo aaa katearen errepikapenez eratuta dauden hitzez osatutako lengoaiari dagokion AFa

Demagun honako AF hau dugula:

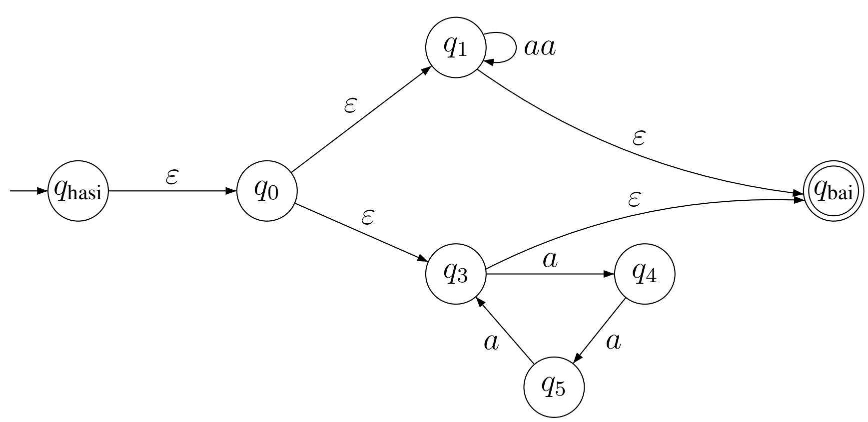


Lehenengo urratsa q_{hasi} eta q_{bai} izeneko bi egoera berri gehitzea da. Hasierako egoera berria q_{hasi} izango da eta zirkulu bikoitza izango duen egoera bakarra q_{bai} izango da. Beraz, q_{hasi} egoeratik q_0 egoerara ε sinboloa duen trantsizio bat ipini beharko da eta hasierako AFan zirkulu bikoitza duen egoera bakoitzetik ere ε sinboloa duen trantsizio bat ipini beharko da q_{bai} egoerara.

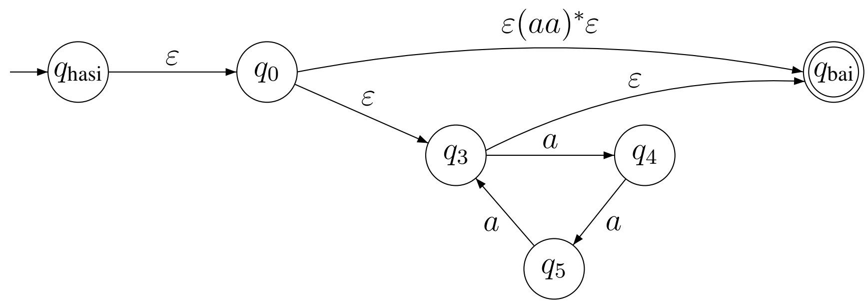


Jarraian, q_0, q_1, q_2, q_3, q_4 eta q_5 egoerak ezabatuz joan behar dugu banan-banan. Berez, berdin da zein ordenatan ezabatzen diren egoerak. Aipatu beharrekoa da aukeratutako ordenaren arabera emaitza gisa espresio desberdina lor daitekeela baina, edozein kasutan ere emaitza zuzena lortuko da. Egoera batzuk ezabatzea beste batzuk ezabatzea baino errazagoa izan daiteke eta, oro har, errazenak ezabatuz hastea da onena. Metodoaren oinarrizko ardatza honako hau da: q_k egoera bat ezabatzean q_k egoera horretatik igarotzen diren bide denak mantendu behar dira.

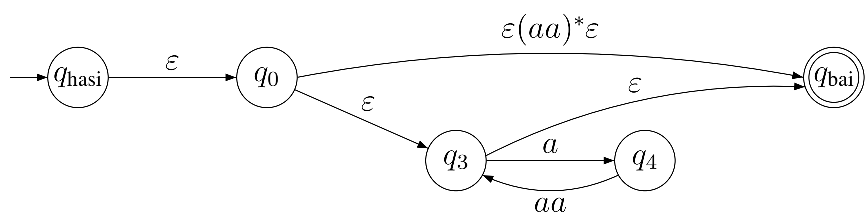
Hasteko, q_2 ezabatuko dugu. Kasu honetan α bakarra eta ψ bakarra ditugu ($\alpha_1 = a$, $\psi_1 = a$) eta ez dago β-rik. Honako hau geldituko zaigu:



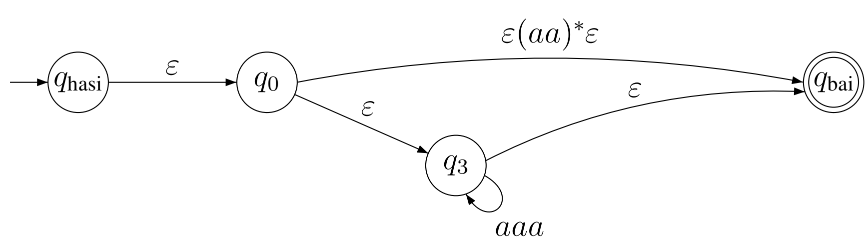
Orain q_1 ezabatuko dugu. Kasu honetan ere α bakarra eta ψ bakarra ditugu ($\alpha_1 = \varepsilon$, $\psi_1 = \varepsilon$) eta gainera β bat dugu ($\beta_1 = aa$). Honako hau geldituko zaigu q_1 ezabatu ondoren:



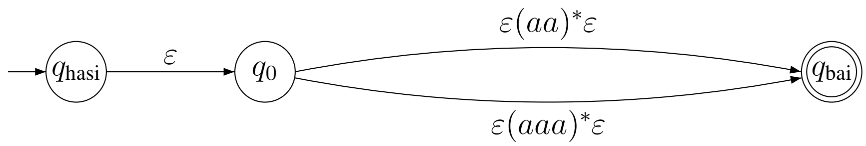
Jarraian q_5 ezabatuko dugu. Kasu honetan ere α bakarra eta ψ bakarra ditugu ($\alpha_1 = a$, $\psi_1 = a$) eta ez dago β-rik. Honako hau lortuko dugu q_5 ezabatu ondoren:



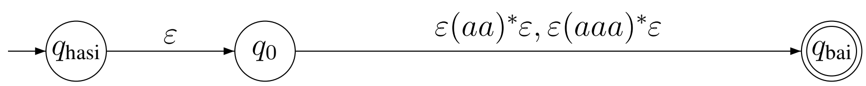
Orain q_4 ezabatuko dugu. Kasu honetan ere α bat eta ψ bat ditugu ($\alpha_1 = a$, $\psi_1 = aa$) eta β-rik ez dago. Honako hau gelditukoko zaigu:



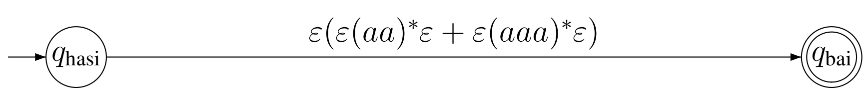
q_3 kenduko dugu orain. Kasu honetan α bat eta ψ bat ditugu ($\alpha_1 = \varepsilon$, $\psi_1 = \varepsilon$) eta baita β bat ere ($\beta_1 = aaa$). Egoera hau kendutakoan honako hau gelditu ko zaigu:



q_0 egoeratik q_{bai} egoerara doazen bi geziak gezi bakar batez ordezkatuko ditugu eta gezi berri horretan lehen genituen bi gezietako etiketak komaz bereizita ipiniko ditugu:



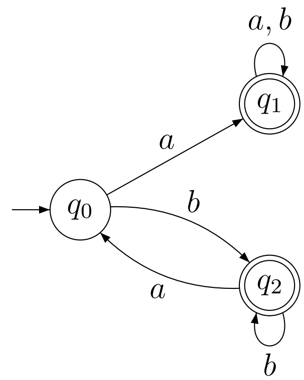
Azkenik, q_0 egoera ezabatuko dugu. Kasu honetan α bat eta bi ψ ditugu ($\alpha_1 = \varepsilon$, $\psi_1 = \varepsilon(aa)^*\varepsilon$, $\psi_2 = \varepsilon(aaa)^*\varepsilon$) eta β-rik ez dago. Honako hau geldituko zaigu q_0 ezabatu ondoren:



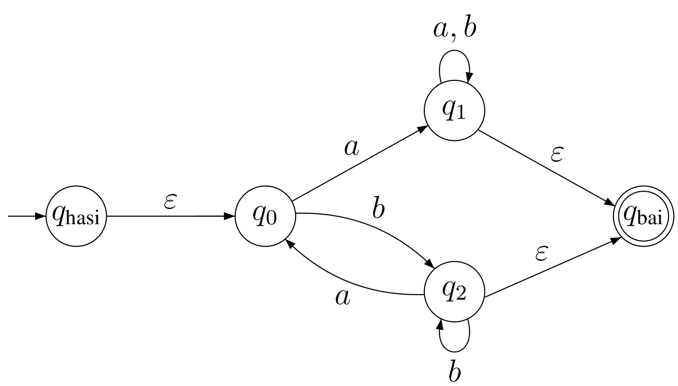
Beraz, adibide honetan abiapuntutzat hartu dugun automatari dagokion lengoaia $\varepsilon(\varepsilon(aa)^*\varepsilon + \varepsilon(aaa)^*\varepsilon)$ espresioaren bidez adieraz daiteke. Espresio hori kateatuta agertzen diren ε sinboloak kenduz sinplifikatzen badugu, $(aa)^* + (aaa)^*$ espresioa geldituko zaigu. Gogora dezagun espresio horrek $(\{a\}\{a\})^* \cup (\{a\}\{a\}\{a\})^*$ adierazten duela. Lengoaia hori, bakarrik aa katearen errepikapenez edo bakarrik aaa katearen errepikapenez eratutako hitzez osatutako lengoaia da.

3.6.2.5 Adibidea: AF bati dagokion lengoaia kalkulatzeko metodoa erakusten duen beste adibide bat

Demagun honako AF hau daukagula:

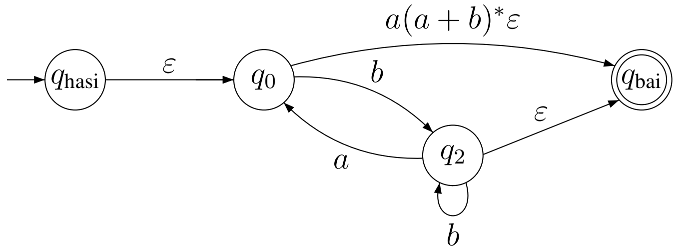


Lehenengo urratsa q_{hasi} eta q_{bai} izeneko bi egoera berri gehitzea da. Hasierako egoera berria q_{hasi} izango da eta zirkulu bikoitza izango duen egoera bakarra q_{bai} izango da. Beraz, q_{hasi} egoeratik q_0 egoerara ε sinboloa duen trantsizio bat ipini beharko da eta hasierako AFan zirkulu bikoitza duen egoera bakoitzetik ere ε sinboloa duen trantsizio bat ipini beharko da q_{bai} egoerara.

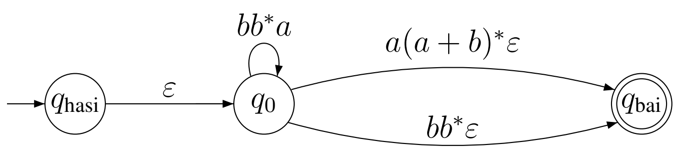


Jarraian, q_0, q_1 eta q_2 egoerak ezabatuz joan behar dugu banan-banan. Berez, berdin da zein ordenatan ezabatzen diren egoerak. Aipatu beharrekoa da aukeratutako ordenaren arabera emaitza gisa espresio desberdina lor daitekeela baina, edozein kasutan ere emaitza zuzena lortuko da. Egoera batzuk ezabatzea beste batzuk ezabatzea baino errazagoa izan daiteke eta, oro har, errazenak ezabatuz hastea da onena. Metodoaren oinarrizko ardatza honako hau da: q_k egoera bat ezabatzean q_k egoera horretatik igarotzen diren bide denak mantendu behar dira.

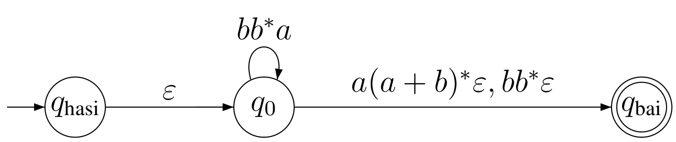
Hasteko, q_1 ezabatuko dugu. Alde batetik, α bat eta ψ bat ditugu ($\alpha_1 = a$, $\psi_1 = \varepsilon$) eta, bestetik, bi β ditugu ($\beta_1 = a$, $\beta_2 = b$). Honako hau geldituko zaigu q_1 ezabatu eta gero:



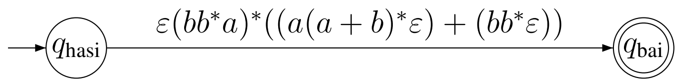
Jarraian, q_2 egoera ezabatuko dugu. Bi bide igarotzen dira q_2 egoeratik: q_0-tik q_0-ra doana, q_2-tik igaroz, eta q_0-tik q_{bai} egoerara doana, q_2-tik igaroz. Bide horietako bakoitza bere aldetik aztertu behar da. Lehenengo bidean α bat, β bat eta ψ bat ditugu ($\alpha_1 = b$, $\beta_1 = b$ eta $\psi_1 = a$). Bigarren bidean ere α bat, β bat eta ψ bat ditugu ($\alpha_1 = b$, $\beta_1 = b$ eta $\psi_1 = \varepsilon$). Honako hau geldituko zaigu q_2 egoera kendu ondoren:



q_0-tik q_{bai} egoerara doazen bi arku ditugunez, bakoitza bere etiketarekin, arku bakarra ipiniko dugu eta arku horrek lehen zeuden bi arkuen etiketak komaz bereizita izango ditu:



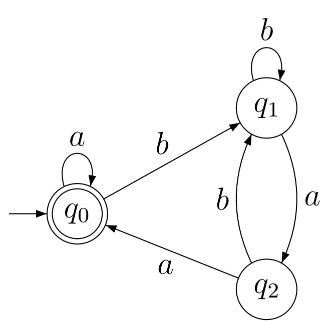
Bukatzeko, q_0 ezabatuko dugu. Kasu honetan, q_0-tik igarotzen den bide bakarra dugu. Bide horretan α bat, β bat eta bi ψ ditugu ($\alpha_1 = \varepsilon$, $\beta_1 = bb^*a$, $\psi_1 = a(a+b)^*\varepsilon$ eta $\psi_2 = bb^*\varepsilon$). Honako hau gelditiko zaigu q_0 ezabatu ondoren:



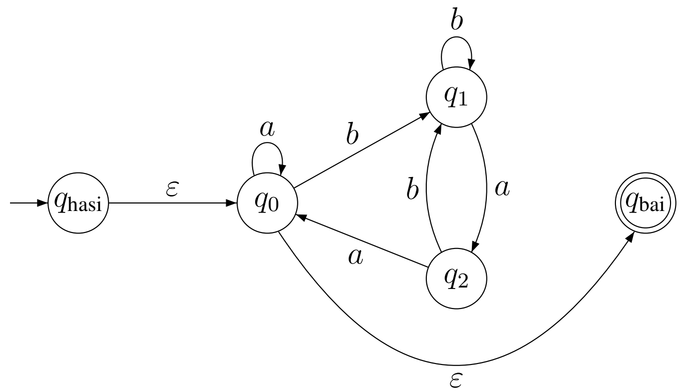
Beraz, adibide honetan abiapuntutzat hartu dugun automatari dagokion lengoaia $\varepsilon(bb^*a)^*((a(a+b)^*\varepsilon)+(bb^*\varepsilon))$ espresioaren bidez adieraz daiteke. Espresio hori kateatuta agertzen diren ε sinboloak kenduz eta beharrezkoak ez diren parentesiak kenduz sinplifika daiteke, eta honako hau lortuko dugu: $(bb^*a)^*(a(a+b)^* + bb^*)$. Espresio horrek $(\{b\}\{b\}^*\{a\})^*(\{a\}(\{a\} \cup \{b\})^* \cup \{b\}\{b\}^*)$ lengoaia adierazten du.

3.6.2.6 Adibidea: AF bati dagokion lengoaia kalkulatzeko metodoa erakusten duen beste adibide bat

Demagun honako AF hau dugula:

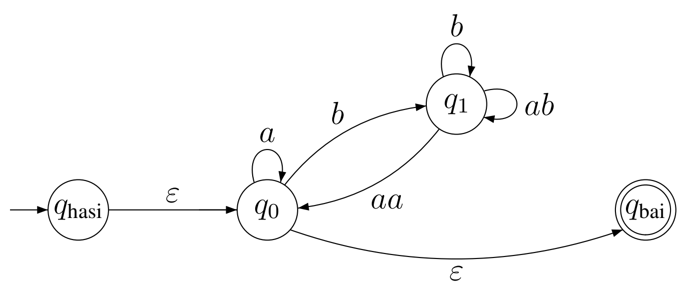


Lehenengo urratsa q_{hasi} eta q_{bai} izeneko bi egoera berri gehitzea da. Hasierako egoera berria q_{hasi} izango da eta zirkulu bikoitza izango duen egoera bakarra q_{bai} izango da. Beraz, q_{hasi} egoeratik q_0 egoerara ε sinboloa duen trantsizio bat ipini beharko da eta hasierako AFan zirkulu bikoitza duen egoera bakoitzetik ere ε sinboloa duen trantsizio bat ipini beharko da q_{bai} egoerara.

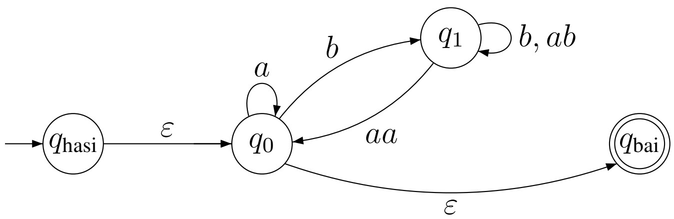


Jarraian, q_0, q_1 eta q_2 egoerak ezabatuz joan behar dugu banan-banan. Berez, berdin da zein ordenatan ezabatzen diren egoerak. Aipatu beharrekoa da aukeratutako ordenaren arabera emaitza gisa espresio desberdina lor daitekeela baina, edozein kasutan ere emaitza zuzena lortuko da. Egoera batzuk ezabatzea beste batzuk ezabatzea baino errazagoa izan daiteke eta, oro har, errazenak ezabatuz hastea da onena. Metodoaren oinarrizko ardatza honako hau da: q_k egoera bat ezabatzean q_k egoera horretatik igarotzen diren bide denak mantendu behar dira.

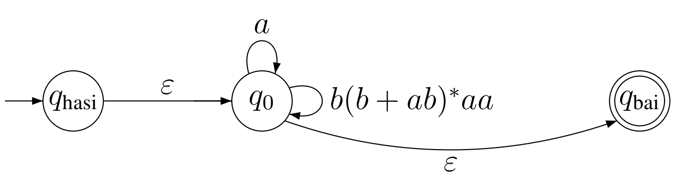
Hasteko, q_2 ezabatuko dugu. Bi bide igarotzen dira q_2 egoeratik: q_1-etik q_0-ra doana eta q_1-etik q_1-era doana. Bide horietako bakoitza bere aldetik aztertu behar da. Lehenengo bidean α bat eta ψ bat ditugu, ez dugu β-rik ($\alpha_1 = a$ eta $\psi_1 = a$). Bigarren bidean ere α bat eta ψ bat ditugu eta ez dugu β-rik ($\alpha_1 = a$ eta $\psi_1 = b$). Honako hau geratuko zaigu q_2 egoera ezabatu ondoren:



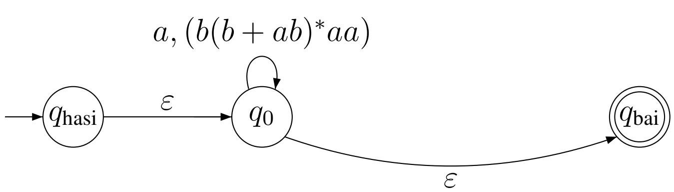
Hor, q_1 egoerak bi begizta ditu, bakoitza bere espresioarekin, baina guk egin ohi dugunari jarraituz etiketa bezala bi espresio horiek komaz bereizita dituen begizta bakarra ipiniko dugu:



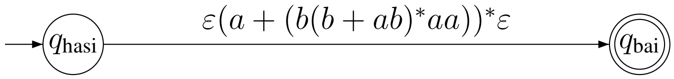
Orain q_1 egoera ezabatuko dugu. Kasu honetan q_1-etik igarotzen den bide bakarra dugu: q_0-tik q_0-ra doana. Bide horretan α bat, bi β eta ψ bat ditugu ($\alpha_1 = b$, $\beta_1 = b$, $\beta_2 = ab$ eta $\psi_1 = aa$). Honako hau geldituko zaigu q_1 egoera kendu ondoren:



Hor, q_0-tik q_0-ra doazen bi arku ditugunez, egoerak ezabatzen jarraitu aurretik arku horietan agertzen diren espresioak komaz bereizita dituen arku bakarra ipiniko dugu:



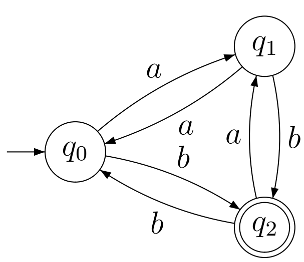
Azkenik, q_0 ezabatuko dugu. Kasu honetan q_0-tik igarotzen den bide bakarra dugu: q_{hasi} egoeratik q_{bai} egoerara doana. Bide horretan α bat, bi β eta ψ bat ditugu ($\alpha_1 = \varepsilon$, $\beta_1 = a$, $\beta_2 = b(b + ab)^*aa$ eta $\psi_1 = \varepsilon$). Honako hau geldituko zaigu q_0 egoera ezabatu ondoren:



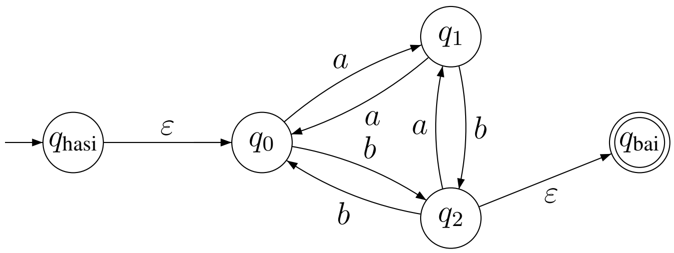
Beraz, adibide honetan abiapuntutzat hartu dugun automatari dagokion lengoaia $\varepsilon(a+(b(b+ab)^*aa))^*\varepsilon$ espresioaren bidez adieraz daiteke. Espresio hori kateatuta agertzen diren ε sinboloak kenduz eta beharrezkoak ez diren parentesiak kenduz sinplifika daiteke eta hori egiten badugu, honako hau geldituko zaigu: $(a + b(b + ab)^*aa)^*$. espresio horrek $(\{a\} \cup \{b\}(\{b\} \cup \{a\}\{b\})^*\{a\}\{a\})^*$ lengoaia adierazten du.

3.6.2.7 Adibidea: AF bati dagokion lengoaia kalkulatzeko metodoa erakusten duen azkeneko adibidea

Demagun honako AF hau dugula:

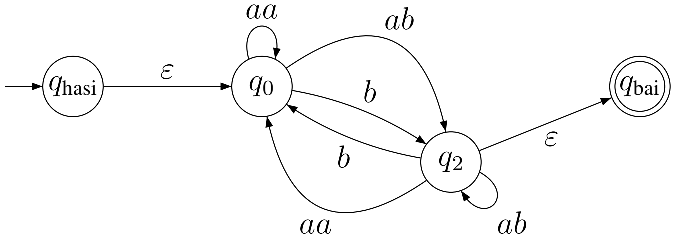


Lehenengo urratsa q_{hasi} eta q_{bai} izeneko bi egoera berri gehitzea da. Hasierako egoera berria q_{hasi} izango da eta zirkulu bikoitza izango duen egoera bakarra q_{bai} izango da. Beraz, q_{hasi} egoeratik q_0 egoerara ε sinboloa duen trantsizio bat ipini beharko da eta hasierako AFan zirkulu bikoitza duen egoera bakoitzetik ere ε sinboloa duen trantsizio bat ipini beharko da q_{bai} egoerara.

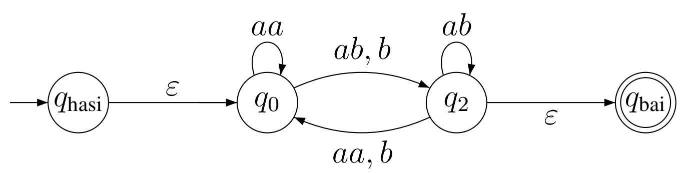


Jarraian, q_0, q_1 eta q_2 egoerak ezabatuz joan behar dugu banan-banan. Berez, berdin da zein ordenatan ezabatzen diren egoerak. Aipatu beharrekoa da aukeratutako ordenaren arabera emaitza bezala espresio desberdina lor daitekeela baina, edozein kasutan ere emaitza zuzena lortuko da. Egoera batzuk ezabatzea beste batzuk ezabatzea baino errazagoa izan daiteke eta, oro har, errazenak ezabatuz hastea da onena. Metodoaren oinarrizko ardatza honako hau da: q_k egoera bat ezabatzean q_k egoera horretatik igarotzen diren bide denak mantendu behar dira.

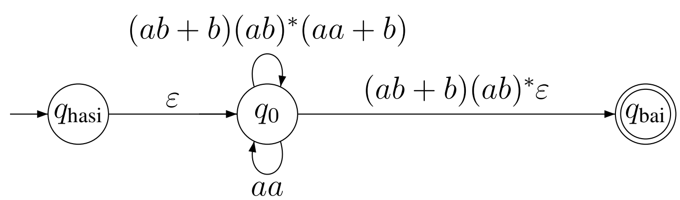
Hasteko, q_1 egoera ezabatuko dugu. Automata horretan, q_1 egoeratik igarotzen diren lau bide daude: q_0-tik q_2-ra doana, q_2-tik q_0-ra doana, q_0-tik q_0-ra doana eta q_2-tik q_2-ra doana. Lau bide horietako bakoitza bere aldetik aztertu behar da. Lehenengo bidean α bat eta ψ bat ditugu eta β-rik ez dugu ($\alpha_1 = a$ eta $\psi_1 = b$). Bigarren bidean ere α bat eta ψ bat ditugu eta ez dugu β-rik ($\alpha_1 = a$ eta $\psi_1 = a$). Hirugarrengo bidean, α bat eta ψ bat ditugu eta ez dugu β-rik ($\alpha_1 = a$ eta $\psi_1 = a$). Azkenik, laugarren bidean ere α bat eta ψ bat ditugu eta ez dugu β-rik ($\alpha_1 = a$ eta $\psi_1 = b$). Honako hau gelditothers zaigu q_1 egoera ezabatu ondoren:



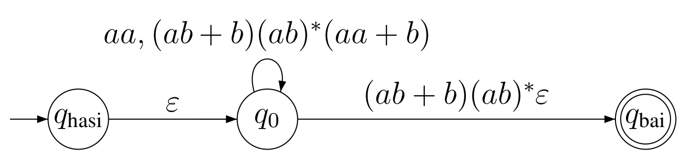
Hor q_0 eta q_2-ren artean norabide bietan bina arku daudenez, arku horiek ohiko formatuan ipiniko ditugu. Beraz, norabide bakoitzean arku bakarra ipiniko dugu eta arku horietako bakoitzak lehengo bi arkuen espresioak izango ditu komaz bereizita:



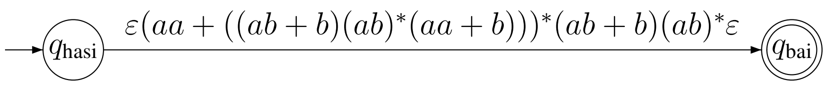
Orain q_2 ezabatuko dugu. Hor, q_2-tik igarotzen diren bi bide daude: q_0-tik q_0-ra doana eta q_0-tik q_{bai} egoerara doana. Bide horietako bakoitza bere aldetik aztertu behar da. Lehenengo bidean bi α, β bat eta bi ψ ditugu ($\alpha_1 = ab$, $\alpha_2 = b$, $\beta_1 = ab$, $\psi_1 = aa$ eta $\psi_2 = b$). Bigarrengo bidean bi α, β bat eta ψ bat ditugu ($\alpha_1 = ab$, $\alpha_2 = b$, $\beta_1 = ab$ eta $\psi_1 = \varepsilon$). Honako hau geldituko zaigu q_2 egoera ezabatu ondoren:



q_0 egoerak bi begizta dituenez, bakoitza bere espresioarekin, begizta bakarra ipiniko dugu bi espresioak koma batez bereiziz:



Bukatzeko q_0 ezabatuko dugu. Hor, q_0-tik igarotzen den bide bakarra dugu. Bide horretan, α bat, bi β eta ψ bat ditugu ($\alpha_1 = \varepsilon$, $\beta_1 = (ab+b)(ab)^*(aa+b)$, $\beta_2 = aa$ eta $\psi_1 = (ab+b)(ab)^*\varepsilon$). Honako hau izango dugu q_0 egoera ezabatu ondoren:



Beraz, adibide honetan abiapuntutzat hartu dugun AFari dagokion lengoaia $\varepsilon(aa + ((ab + b)(ab)^*(aa + b)))^*(ab + b)(ab)^*\varepsilon$ espresioaren bidez adieraz daiteke. Espresio hori kateatuta dauden ε sinboloaren agerpenak kenduz eta beharrezkoak ez diren parentesiak kenduz sinplifika daiteke. Sinplifikazio horren emaitza $(aa+(ab+b)(ab)^*(aa+b))^*(ab+b)(ab)^*$ espresioa izango da. Espresio horrek honako lengoaia hau adierazten du:

$$((\{a\}\{a\}) \cup (\{a\}\{b\} \cup \{b\})(\{a\}\{b\})^*(\{a\}\{a\} \cup \{b\}))^*(\{a\}\{b\} \cup \{b\})(\{a\}\{b\})^*$$

3.7.

Lengoaia erregularrak

Aurreko atalean ikusi den bezala, automata finituen bidez definitutako lengoaiei dagozkien espresioak kalkulatzean, bakarrik lengoaien gaineko honako hiru eragiketa hauek erabiltzen dira: bilketa ($\cup$ edo $+$ eragileen bidez adierazita), itxidura ($*$ eragilearen bidez adierazita) eta kateaketa (ez du eragile esplizitorik). Arrazoi horrengatik, bakarrik hiru eragiketa horiek erabiliz definigarriak diren lengoaiak kontsideratuko ditugu orain. Lengoaia horiei *lengoaia erregularrak* deitzen zaie.

3.7.1 Lengoaia erregularren definizioa

Har dezagun A alfabeto bat. A alfabetoaren gaineko lengoaia erregularren definizio errekurtsiboa honako hau da:

- $\varnothing$ eta $\{\varepsilon\}$ lengoaiak A alfabetoaren gaineko lengoaia erregularrak dira.
- A alfabetoko α sinbolo bakoitzeko, $\{\alpha\}$ lengoaia A alfabetoaren gaineko lengoaia erregularra da.
- L lengoaia A alfabetoaren gaineko lengoaia erregularra baldin bada, orduan L^* lengoaia (L-ren itxidura) A alfabetoaren gaineko lengoaia erregularra da.
- L_1 eta L_2 lengoaiak A alfabetoaren gaineko lengoaia erregularrak baldin badira, orduan L_1L_2 lengoaia (L_1 eta L_2 lengoaien kateadura) eta $L_1 \cup L_2$ lengoaia (L_1 eta L_2 lengoaien bildura) A alfabetoaren gaineko lengoaia erregularrak dira.

Beraz, $A = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$ alfabetoa hartzen badugu ($n \geq 1$ izanda), alde batetik $\varnothing$, $\{\varepsilon\}$, $\{\alpha_1\}$, $\{\alpha_2\}$, $\dots$ eta $\{\alpha_n\}$ lengoaia erregularrak dira. Bestetik, lengoaia erregularren itxidura bezala lortzen diren lengoaiak ere erregularrak dira. Azkenik, lengoaia erregularrei kateaketa eta bilketa eragiketak aplikatuz lortzen diren lengoaiak ere erregularrak dira.

Gogoratu L lengoaiaren itxidura, hau da, L^* lengoaia honela definitzen dela:

$$L^* = L^0 \cup L^1 \cup L^2 \cup L^3 \dots$$

Beraz, L^* lengoaia L lengoaiako hitzak ahal den aldi guztietan kateatuz lortzen da. Adibidez, $L = \{ab, cc\}$ baldin bada, L^* lengoaia honela osatuta egongo litzateke:

- Zero kateaketen bidez lortutako hitzak (L^0): ε
- Kateaketa bakar baten bidez lortutako hitzak (L^1): ab, cc
- Bi kateaketen bidez lortutako hitzak (L^2): $abab$, $abcc$, $ccab$, $cccc$
- Hiru kateaketen bidez lortutako hitzak (L^3): $ababab$, $ababcc$, $abccab$, $abcccc$, $ccabab$, $ccabcc$, $ccccab$, $cccccc$
- Eta abar.

Adibidetzat emandako hitzetan agerian gelditzen da L lengoaiako hitzetan sinboloen ordena ezin dela aldatu eta, ondorioz, esate baterako ba, $baba$ eta $bacc$ hitzak ez direla L^* lengoaiakoak. Izan ere, ba ez da L lengoaiakoa.

3.7.2 Lengoaia erregularren adibideak

$A = \{a, b, c\}$ alfabetoa hartuko dugu.

1. b-rik eta c-rik ez duten hitzez osatutako L_1 lengoaia, lengoaia erregularra da. L_1 erregularra dela frogatzeko, oinarrizko lengoaia erregularrak, hau da, $\varnothing$, $\{\varepsilon\}$, $\{a\}$, $\{b\}$ eta $\{c\}$ lengoaiak eta kateaketa, bilketa eta itxidura erabiliz definitu behar da. L_1 lengoaia erregularra da $\{a\}$ lengoaia eta itxidura erabiliz defini daitekeelako: $L_1 = \{a\}^*$. Gauza bera egiteko beste era bat, hasteko $H = \{a\}$ lengoaia erregularra definitzea eta gero L_1 lengoaia H erabiliz definitzea izango litzateke: $L_1 = H^*$. L_1 era laburtuan definitzeko era honako hau izango litzateke: $L_1 = a^*$.

2. a siboloaren errepikapen bat edo gehiagoz osatutako hitz guztiez eratutako L_2 lengoaia, lengoaia erregularra da. L_2 definitzeko aukera bat honako hau da: $L_2 = L_1 \setminus \{\varepsilon\}$. Baina definizio horrek ez du balio L_2 erregularra dela frogatzeko. Izan ere, $\setminus$, hau da, kenketa, ez da onartzen lengoaia bat erregularra dela frogatzean. Bestalde, L_2 ezin da $\{a\}^*$ gisa definitu, $\{a\}^*$ lengoaian ε ere badagoelako baina ε hitza ez da L_2 lengoaiakoa. L_2 lengoaiako hitzek gutxienez a bat izan behar dutenez, L_2 lengoaia $\{a\}$ lengoaia erregularraren eta $\{a\}^*$ lengoaia erregularraren arteko kateadura bezala defini daiteke. Beraz, $L_2 = \{a\}\{a\}^*$, edo aurreko adibideko lengoaiak erabiliz, $L_2 = HL_1$. L_2 era laburtuan honela defini daiteke: $L_2 = aa^*$.

3. A alfabetoko hiru sinboloez osatutako $L_3 = \{a, b, c\}$ lengoaia erregularra da. Izan ere, L_3 lengoaia $\{a\}$, $\{b\}$ eta $\{c\}$ lengoaia erregularren bildura gisa defini daiteke. Hau da, $L_3 = \{a\} \cup \{b\} \cup \{c\}$. Gauza bera egiteko beste era bat, hasteko $H_1 = \{a\}$, $H_2 = \{b\}$ eta $H_3 = \{c\}$ lengoaia erregularrak definitzea eta gero, L_3 honela definitzea izango litzateke: $L_3 = H_1 \cup H_2 \cup H_3$. Azkenik, L_3 era laburtuan honela defini daiteke: $L_3 = a + b + c$.

4. A alfabetoaren gainean defini daitezkeen hitz guztiez eratutako A^* lengoaia unibertsala ere erregularra da. Izan ere, L_3 lengoaia erregularraren itxidura gisa defini daiteke. Hau da, $A^* = (L_3)^*$. Gauza bera adierazteko beste era bat honako hau litzateke: $A^* = (\{a\} \cup \{b\} \cup \{c\})^*$. Era laburtuan A^* honela defini daiteke: $A^* = (a + b + c)^*$.

5. b-rik eta c-rik ez duten eta a kopuru bikoitia duten hitzez osatutako lengoaia (ε, aa, $aaaa$, $aaaaaa$, eta era horretako hitzak dituen lengoaia) erregularra da. L_4 erregularra dela frogatzeko, hasteko, $L_5 = \{aa\}$ lengoaia hartuko dugu. L_5 erregularra da $\{a\}$ eta $\{a\}$ lengoaiak kateatuz defini daitekeelako: $L_5 = \{a\}\{a\}$. Edo beste era batera idatziz, $L_5 = HH$, hor H lengoaia $\{a\}$ lengoaia erregularra izanda. L_4 lengoaia L_5 lengoaiaren itxidura gisa defini daiteke: $L_4 = (L_5)^*$. L_4 definitzeko beste era bat honako hau litzateke: $L_4 = (\{a\}\{a\})^*$. Eta L_4 era laburtuan honela defini daiteke: $L_4 = (aa)^*$.

6. a sinboloaz hasten diren hitz denez osatutako L_6 lengoaia ere lengoaia erregularra da. Lengoaia hau era formalean honela defini daiteke:

$$L_6 = \{w | w \in A^* \wedge \exists u(u \in A^* \wedge w = au)\}$$

   Definizio horrek ez du balio L_6 erregularra dela frogatzeko. L_6 erregularra dela frogatzeko, oinarrizko lengoaia erregularrak, hau da, $\varnothing$, $\{\varepsilon\}$, $\{a\}$, $\{b\}$ eta $\{c\}$ lengoaiak eta kateaketa, bilketa eta itxidura erabiliz definitu behar da. L_6 definitzeko aukera bat honako hau litzateke: $L_6 = \{a\}(\{a\} \cup \{b\} \cup \{c\})^*$. Gauza bera idazteko beste era bat $H = \{a\}$ eta $L_3 = \{a, b, c\}$ lengoaia erregularrak erabiliz izango litzateke: $L_6 = H(L_3)^*$. Azkenik, L_6 era laburtuan honela defini daiteke: $L_6 = a(a + b + c)^*$.

7. Jarraian, b sinboloaz bukatzen diren hitz denez osatutako L_7 lengoaia, lengoaia erregularra dela frogatuko dugu. Lengoaia hori era formalean definitzeko aukera bat honako hau da:

$$L_7 = \{w | w \in A^* \wedge \exists u(u \in A^* \wedge w = ub)\}$$

   Baina definizio horrek ez du balio L_7 erregularra dela frogatzeko. L_7 erregularra dela frogatzeko, oinarrizko lengoaia erregularrak, hau da, $\varnothing$, $\{\varepsilon\}$, $\{a\}$, $\{b\}$ eta $\{c\}$ lengoaiak eta kateaketa, bilketa eta itxidura erabiliz definitu behar da. Elementu horiek erabiliz L_7 honela defini daiteke: $L_7 = (\{a\} \cup \{b\} \cup \{c\})^*\{b\}$. Gauza bera adierazteko beste era bat $H = \{b\}$ eta $L_3 = \{a, b, c\}$ lengoaia erregularrak erabiliz izango litzateke: $L_7 = (L_3)^*H$. Azkenik, L_7 era laburtuan honela defini daiteke: $L_7 = (a + b + c)^*b$.

8. Orain, a sinboloaz hasi eta b sinboloaz bukatzen diren hitzez osatutako L_8 lengoaia erregularra dela frogatuko dugu. L_8 era formalean definitzeko aukera bat honako hau da:

$$L_8 = \{w | w \in A^* \wedge \exists u(u \in A^* \wedge w = aub)\}$$

Baina definizio horrek ez du balio L_8 erregularra dela frogatzeko. L_8 erregularra dela frogatzeko, oinarrizko lengoaia erregularrak, hau da, $\varnothing$, $\{\varepsilon\}$, $\{a\}$, $\{b\}$ eta $\{c\}$ lengoaiak eta kateaketa, bilketa eta itxidura erabiliz definitu behar da. Elementu horiek erabiliz, L_8 honela defini daiteke: $L_8 = \{a\}(\{a\} \cup \{b\} \cup \{c\})^*\{b\}$. Gauza bera adierazteko beste era bat $H_1 = \{a\}$, $H_2 = \{b\}$ eta $L_3 = \{a, b, c\}$ lengoaia erregularrak erabiliz izango litzateke: $L_8 = H_1(L_3)^*H_2$. L_8 era laburtuan honela defini daiteke: $L_8 = a(a+b+c)^*b$. Bestalde, lehenago L_6 eta L_7 erregularrak direla frogatu dugunez, L_8 lengoaia L_6L_7 gisa definitzeak ere L_8 erregularra dela frogatzeko balio du.

9. c-rik ez duten eta a sinboloaren agerpen denak (a-rik baldin badago) ezkerreko aldean eta b sinboloaren agerpen denak (b-rik baldin badago) eskuin aldean dituzten hitz denez osatutako L_9 lengoaia erregularra dela frogatuko dugu orain. L_9 lengoaian ε, $aaaa$, bbb eta $aabbb$ hitzak izango genituzke adibidez. Lengoaia hori era formalean definitzeko era bat honako hau da:

$$L_9 = \{w | w \in A^* \wedge \exists u, v(u \in A^* \wedge v \in A^* \wedge |u| = |u|_a \wedge |v| = |v|_b \wedge w = uv)\}$$

Baina definizio horrek ez du balio L_9 erregularra dela frogatzeko. L_9 erregularra dela frogatzeko, oinarrizko lengoaia erregularrak, hau da, $\varnothing$, $\{\varepsilon\}$, $\{a\}$, $\{b\}$ eta $\{c\}$ lengoaiak eta kateaketa, bilketa eta itxidura erabiliz definitu behar da. Elementu horiek erabiliz, L_9 honela defini daiteke: $L_9 = \{a\}^*\{b\}^*$. Gauza bera adierazteko beste era bat $H_1 = \{a\}$ eta $H_2 = \{b\}$ lengoaiak erabiliz izango litzateke: $L_9 = (H_1)^*(H_2)^*$. Bestalde, L_9 era laburtuan honela defini daiteke: $L_9 = a^*b^*$.

10. c-rik ez duten eta a-ren agerpenak eta b-ren agerpenak nahastu gabe dituzten hitzez osatutako L_{10} lengoaia erregularra dela frogatuko dugu jarraian. L_{10} lengoaian, adibidez, ε, $aaaa$, bbb, $aabbb$ eta $bbbaaaa$ hitzak izango ditugu. Lengoaia hori era formalean definitzeko era bat honako hau da:

$$L_{10} = \{w | w \in A^* \wedge \exists u, v(u \in A^* \wedge v \in A^* \wedge |u| = |u|_a \wedge |v| = |v|_b \wedge (w = uv \vee w = vu))\}$$

Baina definizio horrek ez du balio L_{10} erregularra dela frogatzeko. L_{10} erregularra dela frogatzeko, oinarrizko lengoaia erregularrak, hau da, $\varnothing$, $\{\varepsilon\}$, $\{a\}$, $\{b\}$ eta $\{c\}$ lengoaiak eta kateaketa, bilketa eta itxidura erabiliz definitu behar da. Elementu horiek erabiliz, L_{10} honela defini daiteke: $L_{10} = (\{a\}^*\{b\}^*) \cup (\{b\}^*\{a\}^*)$. Gauza bera adierazteko beste era bat $H_1 = \{a\}$ eta $H_2 = \{b\}$ lengoaiak erabiliz da: $L_{10} = (H_1)^*(H_2)^* \cup (H_2)^*(H_1)^*$. Bestalde, L_{10} era laburtuan honela defini daiteke: $L_{10} = (a^*b^*) + (b^*a^*)$.

11. Zehazki bi a eta, bestetik, b eta c edozein kopurutan dituzten hitzez osatutako L_{11} lengoaia erregularra dela frogatuko dugu jarraian. L_{11} lengoaian ε, aa, $abacc$, $bbbaabbb$, $cabbbabb$ eta $babbbca$ hitzak izango ditugu adibidez. Lengoaia hori era formalean definitzeko era bat honako hau da:

$$L_{11} = \{w | w \in A^* \wedge |w|_a = 2\}$$

Baina definizio horrek ez du balio L_{11} erregularra dela frogatzeko. L_{11} erregularra dela frogatzeko, oinarrizko lengoaia erregularrak, hau da, $\varnothing$, $\{\varepsilon\}$, $\{a\}$, $\{b\}$ eta $\{c\}$ lengoaiak eta kateaketa, bilketa eta itxidura erabiliz definitu behar da. Elementu horiek erabiliz, L_{11} honela defini daiteke:

$$L_{11} = (\{b\} \cup \{c\})^*\{a\}(\{b\} \cup \{c\})^*\{a\}(\{b\} \cup \{c\})^*$$

Bestalde, L_{11} era laburtuan honela defini daiteke: $L_{11} = (b+c)^*a(b+c)^*a(b+c)^*$.

12. a sinboloa kopuru bikoitian duten hitzez osatutako L_{12} lengoaia erregularra dela frogatuko dugu orain. Adibidez, ε, aa, $abacc$, $aaaa$ eta $ababaabbbcaa$ hitzak L_{12} lengoaiakoak dira. Lengoaia hori era formalean definitzeko era bat honako hau da:

$$L_{12} = \{w | w \in A^* \wedge |w|_a \bmod 2 = 0\}$$

Hala ere, definizio horrek ez du balio L_{12} erregularra dela frogatzeko. L_{12} erregularra dela frogatzeko, oinarrizko lengoaia erregularrak, hau da, $\varnothing$, $\{\varepsilon\}$, $\{a\}$, $\{b\}$ eta $\{c\}$ lengoaiak eta kateaketa, bilketa eta itxidura erabiliz definitu behar da. Elementu horiek eta bi a dituzten hitzez osatutako L_{11} lengoaia erregularra erabiliz, L_{12} definitzeko era bat honako hau izango litzateke: $L_{12} = (L_{11})^* \cup (\{b\} \cup \{c\})^*$. Hor, * eragilea L_{11} lengoaia erregularrari eta $\{b\} \cup \{c\}$ lengoaia erregularrari aplikatu zaio. $(L_{11})^*$ espresioak berak bakarrik ez du balio L_{12} definitzeko. Izan ere, L_{12}-koak diren hitz batzuk (b, $bccb$, ccb, ccc, $bbbb$ eta abar) ez dira $(L_{11})^*$ lengoaiakoak.

3.8.

Lengoaia erregularrei dagozkien automata finituen kalkulua

Lengoaia erregular bat emanda, lengoaia erregular horri dagokion automata finitua era sistematikoan lortzeko balio duen metodo bat aurkeztuko da atal honetan.

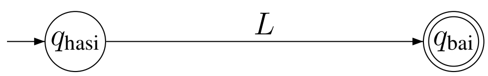
Automata finitua esatean, AFDak, AFEDak eta ε-AFEDak sartzen dira hor. Beraz, aurkeztuko den metodoari jarraituz, AFDa, AFEDa edo ε-AFEDa izan daitekeen automata finitu bat lortuko da. Gehienetan AFED bat edo ε-AFED bat lortuko da.

Metodoa aplikatzean, emango diguten lengoaia erregularra oinarrizkoak diren lengoaia erregularren eta kateaketa, itxidura ($*$) eta bilketa ($+$) eragiketen bidez adierazita egongo da. Esate baterako, alfabetoa $A = \{a, b, c\}$ baldin bada, oinarrizko lengoaiak $\varnothing$ lengoaia hutsa, $\{\varepsilon\}$ hitz hutsa bakarrik duen lengoaia eta $\{a\}$, $\{b\}$ eta $\{c\}$ izango lirateke. Giltzak ez dira erabiliko. Adibidez, lengoaia erregularrak $aa(b^*) + (bb)^*$ erako espresioen bidez adieraziko dira $\{a\}\{a\}(\{b\}^*) \cup (\{b\}\{b\})^*$ erako espresioen bidez adierazi ordez. Ahal den gehienetan parentesien erabilera ere saihestu egingo da lehentasun handieneko eragilea $*$ dela (itxidura), bigarren lehentasun handieneko eragilea kateaketa dela eta hirugarrean lekuan $+$ (bilketa) eragilea dugula kontuan hartuz. Beraz, $(aa(b^*)) + ((bb)^*)$ idatzi beharrean $aab^* + (bb)^*$ idatziko dugu.

3.8.1 Egoera berriak sortuz, lengoaia erregularrei dagozkien AFak eraikitzen dituen metodoa

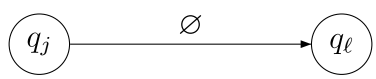
Lengoaia erregular batetik abiatuz AF bat eraikitzeko prozedura zehaztuko da ondoren.

Goian esan den formatuan adierazitako L lengoaia erregularra emanda, lehenengo urratsa honako egitura duen AF bat sortzea izango da:



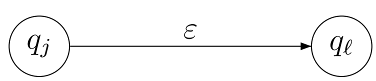
Egitura hori duen AFa sortu ondoren, jarraian zehaztuko diren eskema orokorrak aplikatu beharko dira arku edo trantsizio bakoitzean A alfabetoko sinbolo bakar bat edo ε sinboloa duen AF bat lortu arte:

- L lengoaia hutsa baldin bada, hau da, $L = \varnothing$ baldin bada:



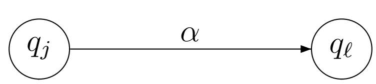
orduan q_j eta q_ℓ egoeren arteko arkua kendu egin beharko da.

- L bakarrik hitz hutsaz osatutako lengoaia baldin bada, hau da, $L = \varepsilon$ baldin bada:



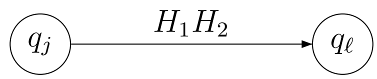
orduan dagoen bezala laga beharko da.

- L lengoaia A alfabetoko α sinbolo bakar batez osatutako oinarrizko lengoaia bat baldin bada, hau da, $L = \alpha$ baldin bada:

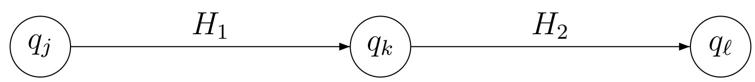


orduan dagoen bezala laga beharko da.

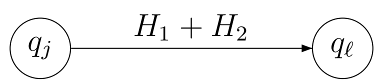
- L lengoaia H_1 eta H_2 beste bi lengoaia erregularren kateadura bezala definituta baldin badago, hau da, $L = H_1H_2$ baldin bada:



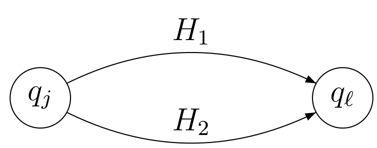
orduan egoera berri bat sortu beharko da:



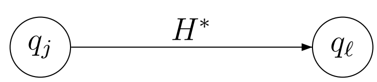
- L lengoaia H_1 eta H_2 beste bi lengoaia erregularren bildura bezala definituta baldin badago, hau da, $L = H_1 + H_2$ baldin bada:



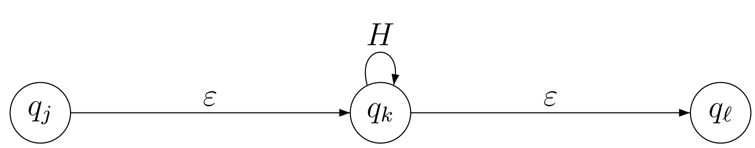
orduan arku berri bat sortu beharko da:



- L lengoaia H beste lengoaia erregular baten itxidura bezala definituta baldin badago, hau da, $L = H^*$ baldin bada:

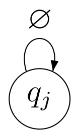


orduan egoera berri bat sortu beharko da:

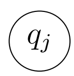


q_j eta q_ℓ egoerak egoera bera direnean eskema horiek nola aplikatu behar diren azalduko da jarraian:

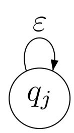
- L lengoaia hutsa baldin bada, hau da, $L = \varnothing$ baldin bada:



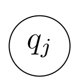
  orduan q_j egoerako begizta kendu egin beharko da:



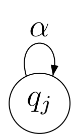
- L lengoaia hitz hutsaz bakarrik osatutako lengoaia baldin bada, hau da, $L = \varepsilon$ baldin bada:



  orduan ere begizta kendu egin beharko da [1]:



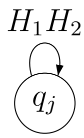
- L lengoaia A alfabetoko α sinbolo bakar batez osatutako oinarrizko lengoaia bat baldin bada, hau da, $L = \alpha$ baldin bada:



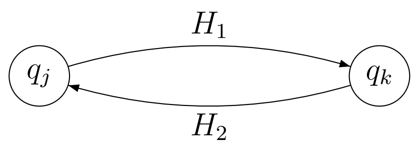
  orduan dagoen bezala laga beharko da.

[1] Kontuan izan kasu honetan q_j eta q_ℓ egoera desberdinak direnean gezia mantendu egin behar dela esan dugula.

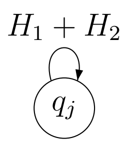
- L lengoaia H_1 eta H_2 beste bi lengoaia erregularren kateadura gisa definituta baldin badago, hau da, $L = H_1H_2$ baldin bada:



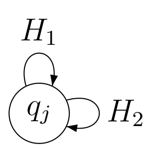
orduan q_k egoera berri bat sortu beharko da:



- L lengoaia H_1 eta H_2 beste bi lengoaia erregularren bildura gisa definituta baldin badago, hau da, $L = H_1 + H_2$ baldin bada:

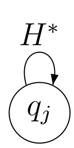


orduan begizta berri bat sortu beharko da:

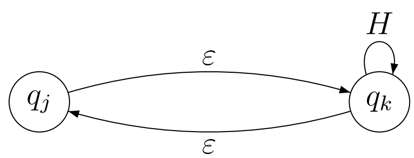


Automata finituetan horrelako bi arku edo begizta ipini beharrean, H_1 eta H_2 komaz bereizita dituen arku bakar bat ipini ohi dugu, baina hori horrela egin ahal izateko, H_1 eta H_2 espresioak A multzoko elementuak izan beharko lukete. H_1 eta H_2 espresioak A multzoko elementuak ez direnean, bi begizta ipini beharko dira espresio bakoitza, H_1 eta H_2, bere aldetik garatzeko.

- L lengoaia H beste lengoaia erregular baten itxidura baldin bada, hau da, $L = H^*$ baldin bada:



orduan egoera berri bat sortu beharko da:

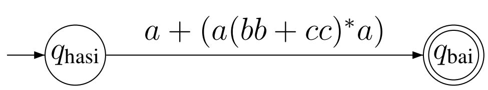


3.8.2 Lengoaia erregular bati dagokion AFaren kalkuluaren adibidea

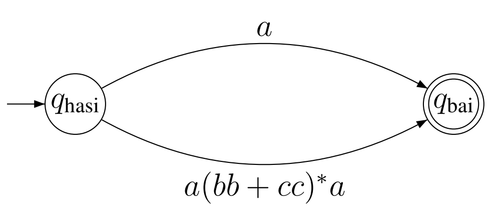
3.8.2.1 Adibidea: $a + (a(bb + cc)^*a)$ lengoaia erregularrari dagokion AFaren kalkulua

$L = a + (a(bb + cc)^*a)$ lengoaia erregularrari dagokion AFa diseinatu nahi da.

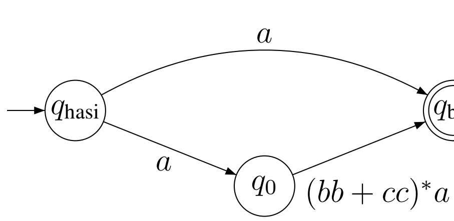
Lehenengo urratsa, q_{hasi} eta q_{bai} egoerak sortu eta bien artean L etiketa duen arku bat ipintzea izango da:



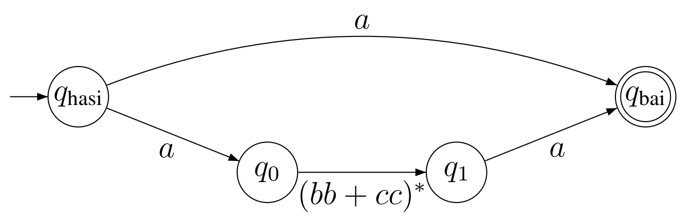
Orain $H_1 = a$ eta $H_2 = a(bb+cc)^*a$ lengoaien arteko bildura daukagu. Beraz, q_{hasi} egoeratik q_{bai} egoerara bi arku ipini beharko dira:



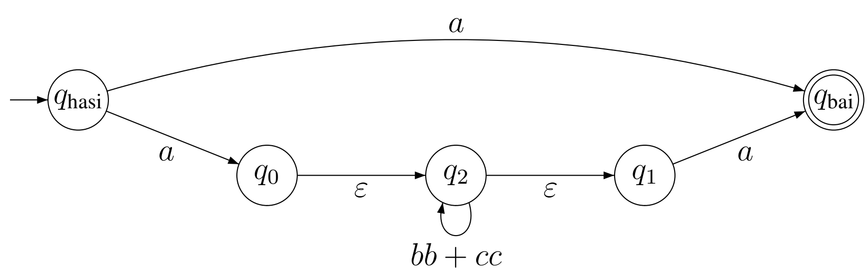
H_1 espresioari dagokion arkua ez da gehiago garatu behar; izan ere, A alfabetoko sinbolo bat da. Beraz, orain H_2 espresioari dagokion arkua garatu beharko da. H_2 espresioa $G_1 = a$ eta $G_2 = (bb + cc)^*a$ lengoaia erregularren kateaketa da. Ondorioz, egoera berri bat sortu beharko da: q_0.



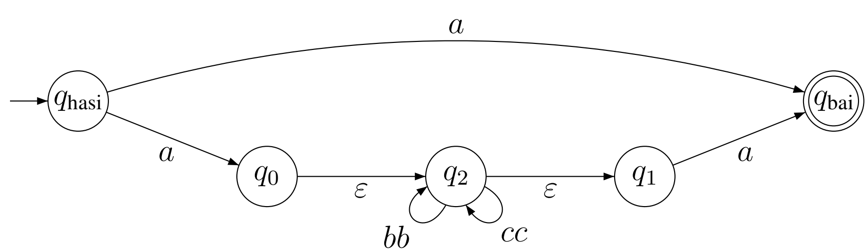
G_1 espresioari dagokion arkua ez da gehiago garatu behar, arku horren etiketa A alfabetoko sinbolo bat baita. Beraz, G_2 espresioari dagokion arkua garatu beharko da orain. G_2 espresioa $F_1 = (bb + cc)^*$ eta $F_2 = a$ lengoaia erregularren kateadura da eta, ondorioz, egoera berri bat sortu beharko da: q_1.



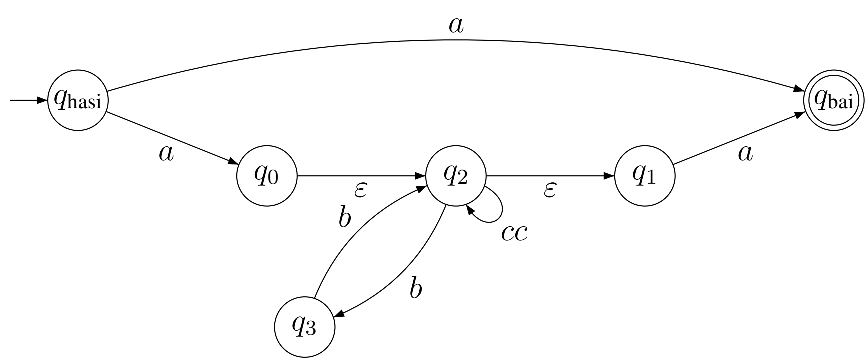
F_2 espresioari dagokion arkua ez da gehiago garatu behar, arku horren etiketa A alfabetoko sinbolo bat baita. Baina F_1 espresioari dagokion arkua garatuz jarraitu beharra dago. F_1 espresioa $I = (bb + cc)$ lengoaia erregularraren itxidura da eta, ondorioz, egoera berri bat sortu beharko da: q_2.



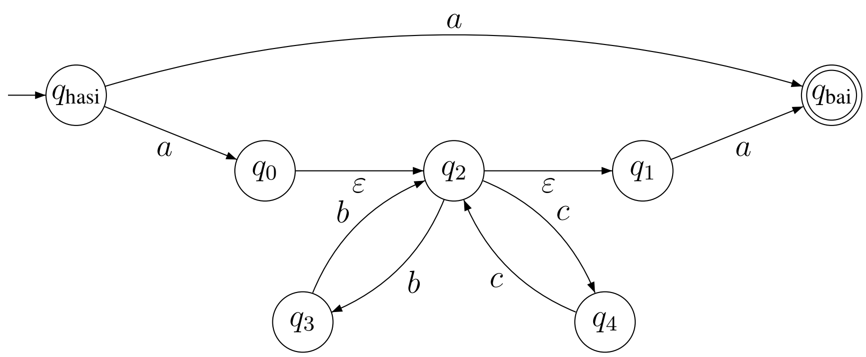
Orain q_2 egoerako begizta garatu behar da. $I = bb + cc$ lengoaia $Z_1 = bb$ eta $Z_2 = cc$ lengoaien bildura da eta kasu horri dagokion eskemari jarraituz, bi begizta ipini beharko dira:



Z_1 lengoaia $Y_1 = b$ eta $Y_2 = b$ lengoaien kateadura denez, egoera beri bat, q_3, ipiniko dugu:



Z_2 lengoaia $X_1 = c$ eta $X_2 = c$ lengoaien kateadura denez, kasu honetan ere beste egoera berri bat, q_4, gehituko dugu.



Arku guztietan A alfabetoko sinbolo bat edo ε sinbolo berezia dugunez, $a + (a(bb + cc)^*a)$ lengoaia erregularrari dagokion automata finitua eraikitzeko prozesua bukatu da.

3.9.

Automata finitu eta lengoaia erregularren arteko baliokidetasuna

Automata finitu (determinista edo ez-determinista) bakoitzak lengoaia bat definitzen du. Hitz baten aurrean automatak "Bai" erantzuten badu, orduan hitz hori lengoaiakoa da eta "Ez" erantzuten badu, ez da lengoaiakoa.

Edozein automata finitu hartuz, automata horri dagokion lengoaia erregularra izango da. Eta edozein lengoaia erregular hartuz, lengoaia hori onartzen duen automata finitua beti existituko da. Beraz lengoaia erregularrak automata finituen bidez defini daitezkeen lengoaiak dira.

3.10.

Lengoaia ez-erregularrak

Automata finitu eta lengoaia erregularren arteko baliokidetasunaren aurrean honako galdera hau sortzen da: Lengoaia denak erregularrak al dira? edo beste era batera esanda, 2^{A^*} multzoko edozein L lengoaia hartzen badugu, L onartzen duen automata finitua kasu denetan existitzen al da? Erantzuna ezezkoa da.

3.10.1 Erregularrak ez diren lengoaiak badaude

Edozein A alfabeto finkatuta ere, A-ren gainean definitutako lengoaia ez-erregularrak badaudela frogatuko dugu orain.

A alfabetoa finkatu ondoren, har dezagun $B = A \cup \{\varnothing, \varepsilon, {}^*, +, (,)\}$ alfabetoa. B^* multzkoko elementuen (hitzen) bidez A-ren gaineko lengoaia erregular denak adieraz ditzakegu. Badakigu B^* zenbagarria dela eta 2^{A^*} zenbaezina dela. Beraz, B^* multzoan 2^{A^*} multzoan baino elementu gutxiago daude. Ondorioz, 2^{A^*} multzoko lengoaia batzuk ezin dira adierazi B^* multzoko hitzen bidez. B^* multzoko hitzen bidez adierazi ezin diren 2^{A^*} multzoko lengoaiak ez dira erregularrak.

3.10.2 Kontesturik gabeko lengoaiak eta piladun automatak

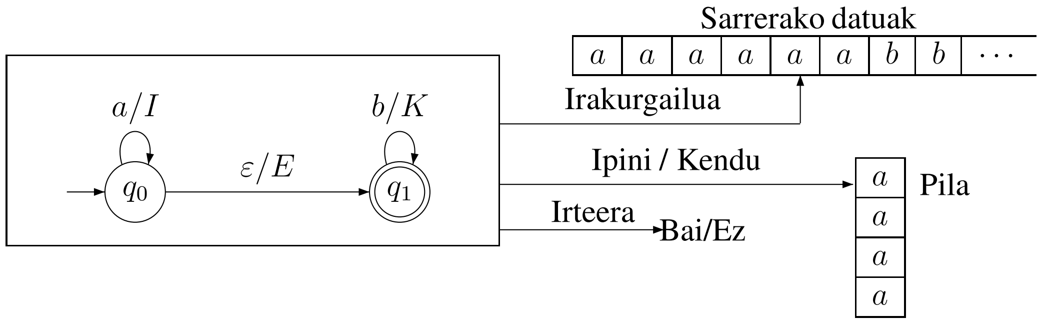
Jarraian datorren lengoaiarentzat ez dago automata finiturik:

$$G_1 = \{w \in A^* \mid w = a^n b^n \wedge n \geq 0\}$$

a eta b kopuru bera duten eta gainera a-ren eta b-ren agerpenak nahastu gabe dituzten hitzez osatuta dago G_1 lengoaia. Beraz, $G_1 = \{\varepsilon, ab, aabb, aaabbb, aaaabbbb, \ldots\}$. G_1 lengoaia L_9 lengoaiaren azpilengoaia da. L_9 lengoaia erregularra izan arren, G_1 ez da erregularra. Kontuan hartu a kopurua eta b kopurua edozein izan daitekeela (0, 1, 2, 3, eta abar). Automata finituetan memoria egoeren bidez lortzen da eta automata finituak gauza batzuk gogoratzeko gai dira. Adibidez, a kopurua bikoitia al den gogoratzeko gai den automata finitu bat defini dezakegu.

Baina G_1 lengoaiaren kasuan lehenengo a kopurua zenbatu behar da eta gero kopuru hori b kopuruarekin alderatu behar da, baina automata finitu batek ezin dezake egin hori, ezin baitezake zenbatu hasieratik ez bada ezagutzen zenbatu beharreko kopurua.

Beraz, lengoaia batzuk ez dira erregularrak eta automata finituen bidez ezin dira lengoaia denak definitu. Esate baterako G_1 lengoaiarentzat jarraian erakusten den piladun automata ez-determinista beharko genuke.



Automata hau pila hutsarekin abiatzen da eta hasierako a denak pilan gordez joaten da b lehenengo aldiz agertu arte. Gero, irakurtzen den b bakoitzeko a bat kentzen da pilatik. Hitza bukatzean pila hutsik badago, badakigu a kopurua eta b kopurua berdinak izan direla. Hitza bukatzean pila hutsik ez badago, orduan badakigu a kopurua b kopurua baino handiagoa izan dela. Bestalde, b-ren agerpen denak irakurri baino lehen pila hutsik gelditzen bada, badakigu b kopurua a kopurua baino handiagoa dela. Gainera, uneren batean c agertzen bada, hitza ez da onartuko. Era berean, b agertu ondoren berriro a agertzen bada, hitza ez da onartuko. Piladun automata hau determinista ez denez, egoeratatik era desberdinetan mugi gaitezke hitza irakurtzean. Hitza irakurriz goazenean, egoeratan zehar mugitzeko era horietako bat ondo bukatuko da makina gelditzean onarpen egoeran baldin bagaude, hitz dena irakurri bada, pilatik elementu bat kendu behar zen bakoitzean kendu ahal izan bada eta azkenean pila hutsik gelditu bada. Pilatik a bat kendu behar denean pila hutsik baldin badago, automata blokeatu egingo da. Gainera, pilatik a bat kendu ahal izateko sarrerako hitzean b bat irakurri behar da. Hitz bat irakurtzean egoerak zeharkatzeko era bat baino gehiago daudenean, nahikoa da era bat ondo bukatzearekin hitza G_1 lengoaiakoa dela jakiteko. Era denak gaizki bukatzen badira, orduan hitza ez da G_1 lengoaiakoa.

Trantsizioetan α/β erako etiketak ditugu. Etiketa horietan α elementua alfabetoko sinbolo bat edo hitz hutsa da (beraz, α elementua $A \cup \{\varepsilon\}$ multzoko sinbolo bat da) eta β elementua $\{E, I, K\}$ multzoko sinbolo bat da. E sinboloak pilan ez dela ezer egin behar adierazten du, I sinboloak pilaren gainean (gailurrean) a sinboloa ipini behar dela adierazten du eta K sinboloak pilaren gailurrean dagoen a sinboloa kendu behar dela adierazten du. Adibidez, $aabb$ hitzarentzat egoeratatik zehar mugitzeko era bat baino gehiago daude:

1. $(aabb, q_0, [\,]) \to (abb, q_0, [a]) \to (bb, q_0, [a, a]) \to (bb, q_1, [a, a]) \to (b, q_1, [a]) \to (\varepsilon, q_1, [\,])$. Bide hau ondo bukatu da. Hitza G_1 lengoaiakoa dela ziurtatzeko beharrezkoak direm baldintzak bete dira.

2. $(aabb, q_0, []) \to (abb, q_0, [a]) \to (abb, q_1, [a])$. Kasu honetan ez da ondo bukatu. Azkarregi pasatu da q_1 egoerara eta jarraian irakurri beharreko sinboloa a da eta pilan ere a dago

eta ondorioz ezin da pilako a hori ezabatu eta automata blokeatu egingo da eta hitza G_1 lengoaiakoa dela ziurtatzeko baldintzak ez dira beteko. Izan ere, pilatik a bat ezabatzeko, irakurtzeko gelditzen den hitz zatiko lehenengo sinboloak b izan behar du q_1 egoeran gaudenean. Gainera, q_1 egoeran a-rik ezin da irakurri.

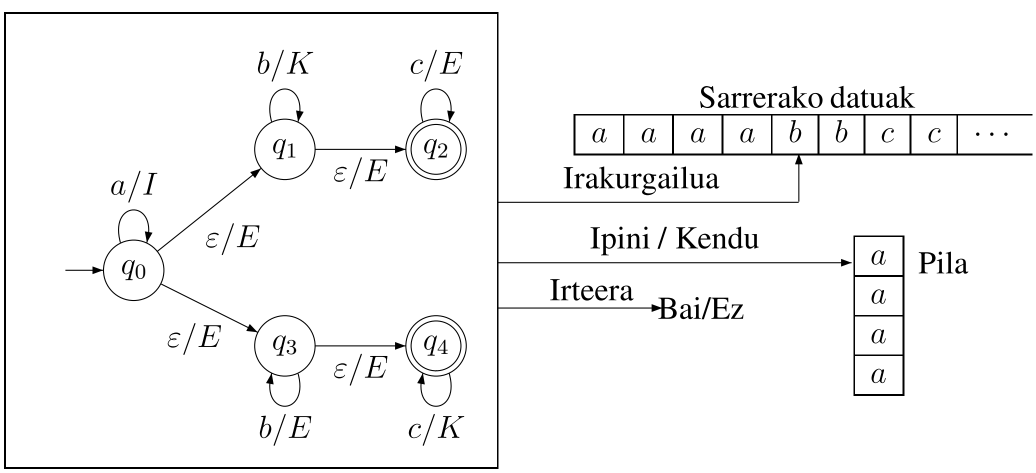
3. $(aabb, q_0, []) \rightarrow (aabb, q_1, [])$. Kasu honetan ere ez da ondo bukatu, izan ere azkarregi pasatu da q_1 egoerara eta jarraian irakurri beharreko sinboloa a da, baina q_1 egoeran a agertzen bada, automata blokeatu egiten da eta hitza G_1 lengoaiakoa dela ziurtatzeko baldintzak ez dira beteko.

Kasu batean ondo bukatu denez, badakigu $aabb$ hitza G_1 lengoaiakoa dela. Esate baterako $aaab$ hitza hartzen badugu, kasu denak gaizki bukatuko dira eta ondorioz badakigu hitz hori ez dela G_1 lengoaiakoa.

Piladun automata ez-deterministen bidez definigarraiak diren lengoaiak **kontesturik gabeko lengoaiak dira**.

Jarraian piladun automata ez-determinista behar duten kontesturik gabeko beste bi lengoaia emango dira: G_2 eta G_3.

- $G_2 = \{w \in A^* \mid w = a^i b^j c^k \wedge i \geq 0 \wedge j \geq 0 \wedge k \geq 0 \wedge (i = j \vee i = k)\}$. Beraz, G_2 lengoaian ε, $aaabbb$, $aaabbbc$, $aaaccc$, $aaabccc$ eta $aabbcc$ erako hitzak daude. Baina, adibidez, $abbccc$ eta aab ez dira G_2 lengoaiakoak.



Hasieran pila hutsik egongo da eta automata sarrerako hitzeko a-ren agerpenak pilan ipiniz joango da b edo c agertu arte. b edo c agertzen denean bi aukera daude:

(a) q_1, q_2 bidea jarraitu eta hitzean irakurtzen den b bakoitzeko pilatik a bat kendu. Irakurtzen ari den hitzeko b denak bukatzean pilako a denak ere kentzea lortu bada, ez dago c kopurua kontrolatu beharrik, bai baitakigu a kopurua eta b kopurua berdinak direla.

(b) q_3, q_4 bidea jarraitu eta hitzeko b-ak irakurri pilatik a-rik kendu gabe eta c lehenengo aldiz agertzen denean, hitzean irakurtzen den c bakoitzeko pilatik a bat kendu. Irakurtzen ari den hitza bukatzean (hau da, c denak bukatzean) pilako a denak ere kentzea lortu bada, badakigu a kopurua eta c kopurua berdinak direla.

Piladun automata hau determinista ez denez, egoeratatik era desberdinetan mugi gaitezke hitza irakurtzean. Hitza irakurriz goazenean, egoeratan zehar mugitzeko era horietako bat ondo bukatuko da makina gelditzean onarpen egoeran baldin bagaude, hitz dena irakurri bada, pilatik elementu bat kendu behar zen bakoitzean kendu ahal izan bada eta azkenean pila hutsik gelditu bada. Pilatik a bat kendu behar denean pila hutsik baldin badago, automata blokeatu egingo da eta hitza G_2 lengoaiakoa dela ziurtatzeko baldintzak ez dira beteko. Gainera, pilatik a bat kendu ahal izateko sarrerako hitzean b bat irakurri behar da q_1 egoeran baldin bagaude eta c bat irakurri behar da q_4 egoeran baldin bagaude. Hitz bat irakurtzean egoerak zeharkatzeko era bat baino gehiago baldin badaude, nahikoa da era bat ondo bukatzearekin hitza G_2 lengoaiakoa dela ziurtatu ahal izateko. Era denak gaizki bukatzen badira, orduan hitza G_2 lengoaiakoa ez dela ziurtatu ahal izango da. E sinboloak pilan ez dela ezer egin behar adierazten du, I sinboloak pilaren gainean (gailurrean) a sinboloa ipini behar dela adierazten du eta K sinboloak pilaren gailurrean dagoen a sinboloa kendu behar dela adierazten du.

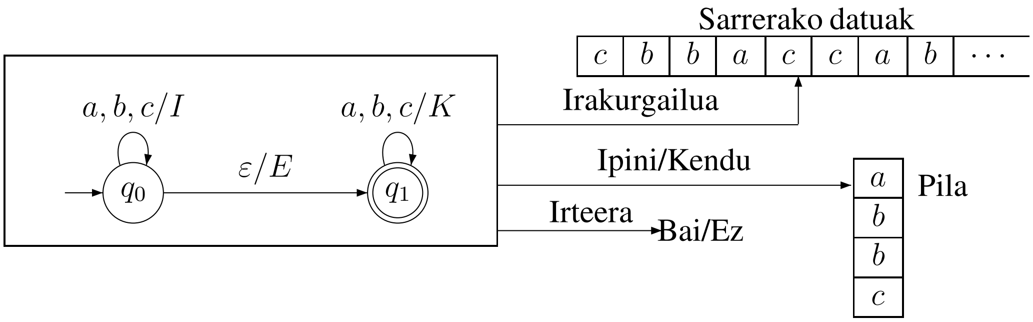
Adibidez, $aabcc$ hitzarentzat aukera bat baino gehiago daude:

1. $(aabcc, q_0, [\,]) \rightarrow (abcc, q_0, [a]) \rightarrow (bcc, q_0, [a, a]) \rightarrow (bcc, q_3, [a, a]) \rightarrow (cc, q_3, [a, a]) \rightarrow (cc, q_4, [a, a]) \rightarrow (c, q_4, [a]) \rightarrow (\varepsilon, q_4, [\,])$. Kasu honetan ondo bukatu da. a kopurua eta c kopurua konparatu dira eta berdinak dira. $aabcc$ hitza G_2 lengoaiakoa dela ziurtatu ahal izateko beharrezkoak diren baldintzak bete dira.
2. $(aabcc, q_0, [\,]) \rightarrow (abcc, q_0, [a]) \rightarrow (bcc, q_0, [a, a]) \rightarrow (bcc, q_1, [a, a]) \rightarrow (cc, q_1, [a]) \rightarrow (cc, q_2, [a]) \rightarrow (c, q_2, [a]) \rightarrow (\varepsilon, q_2, [a])$. Kasu honetan gaizki joan da. Automata q_1 egoerara pasatu da a kopurua eta b kopurua konparatzeko asmoarekin, baina bi kopuru horiek ez dira berdinak. b-ren agerpenak bukatu direnean pila ez dago hutsik, a sinbolo bat dago. Jarraian c-ak irakurtzen dira pilan ezer aldatu gabe eta hitza bukatzean pila ez dago hutsik. $aabcc$ hitza G_2 lengoaiakoa dela ziurtatu ahal izateko bete beharreko baldintzak ez dira betetzen.
3. $(aabcc, q_0, [\,]) \rightarrow (abcc, q_0, [a]) \rightarrow (abcc, q_3, [a]) \rightarrow (abcc, q_4, [a])$. Kasu honetan gaizki joan da q_3 egoerara azkarregi pasatzeagatik. q_3 egoeran a-rik ez denez onartzen, aukera bakarra ezer irakurri gabe q_4-ra pasatzea da. Baina q_4 egoeran a-rik ez denez onartzen automata blokeatu egin da. $aabcc$ hitza G_2 lengoaiakoa dela ziurtatu ahal izateko bete beharreko baldintzak ez dira betetzen.

Gaizki bukatzen diren kasu gehiago ere badaude, baina kasu batean ondo bukatu denez, $aabcc$ hitza G_2 lengoaiakoa dela ziurta daiteke.

Baina $aaab$ hitza hartzen badugu, kasu denak gaizki bukatuko dira eta hitza G_2 lengoaiakoa ez dela ziurtatu ahal izango dugu.

- $G_3 = \{w \in A^* \mid \exists v(v \in A^* \wedge w = vv^R)\}$. G_3 lengoaiako hitz batzuk ε, $\underline{aa}aa$, $\underline{abc}cba$, $\underline{ab}ba$ eta $\underline{ccbaba}ababcc$ dira. Bestalde, adibidez $aaca$ eta aab ez dira G_3 lengoaiakoak.



Hitz batek vv^R egitura al duen erabakitzeko, v-ren sinboloak pilan gordez joango gara eta v^R irakurtzean pilako sinboloak kenduz joango gara. Piladun automata hau determinista ez denez, egoeratatik era desberdinetan mugi gaitezke hitza irakurtzean. Hitza irakurriz goazenean, egoeratan zehar mugitzeko era horietako bat ondo bukatuko da makina gelditzean onarpen egoeran baldin bagaude, hitz dena irakurri bada, pilatik elementu bat kendu behar zen bakoitzean kendu ahal izan bada eta azkenean pila hutsik gelditu bada. Pilatik sinbolo bat kendu behar denean pila hutsik baldin badago, automata blokeatu egingo da eta hitza G_3 lengoaiakoa dela ziurtatu ahal izateko baldintzak ez dira beteko. Gainera, pilatik sinbolo bat kendu ahal izateko, sinbolo horrek irakurtzeko gelditzen den sarrerako hitzaren zatiko lehenengo sinboloaren berdina izan behar du. Hitz bat irakurtzean, egoerak zeharkatzeko era bat baino gehiago baldin badaude, nahikoa da era bat ondo bukatzea hitza G_3 lengoaiakoa dela ziurtatu ahal izateko. Era denak gaizki bukatzen badira, orduan hitza G_3 lengoaiakoa ez dela ziurtatu ahal izango dugu. E sinboloak pilan ez dela ezer egin behar adierazten du, I sinboloak pilaren gainean (gailurrean) azkena irakurri den sinboloa ipini behar dela adierazten du eta K sinboloak pilaren gailurrean dagoen sinboloa kendu behar dela adierazten du, sinbolo hori eta irakurtzeko gelditzen den hitz zatiko lehenengo sinboloa berdinak badira.

Adibidez, $acca$ hitzarentzat aukera bat baino gehiago daude:

1. $(acca, q_0, [\,]) \rightarrow (cca, q_0, [a]) \rightarrow (ca, q_0, [c, a]) \rightarrow (ca, q_1, [c, a]) \rightarrow (a, q_1, [a]) \rightarrow (\varepsilon, q_1, [\,])$. Kasu honetan ondo bukatu da. $acca$ hitza G_3 lengoaiakoa dela ziurtatzeko baldintzak bete egin dira.
2. $(acca, q_0, []) \rightarrow (cca, q_0, [a]) \rightarrow (cca, q_1, [a])$. Kasu honetan gaizki joan da q_1 egoerara azkarregi pasatu delako. Pilatik sinbolo bat kentzeko hitzeko lehenengo sinboloaren berdina izan behar duenez, automata blokeatu egingo da hitzeko lehenengo sinboloa c baita eta pilaren gailurrean a baitago. Beraz, $acca$ hitza G_3 lengoaiakoa dela ziurtatu ahal izateko baldintzak ez dira betetzen.
3. $(acca, q_0, []) \rightarrow (cca, q_0, [a]) \rightarrow (ca, q_0, [c, a]) \rightarrow (a, q_0, [c, c, a]) \rightarrow (a, q_1, [c, c, a])$. Kasu honetan gaizki joan da q_1 egoerara beranduegi pasatu delako. q_1 egoerara igarotakoan, hitzeko lehenengo sinboloa a da eta pilaren gailurrean c dago. Pilatik

elementu bat kentzeko elementu horrek hitzeko lehenengo elementuaren berdina izan behar duenez, automata blokeatu egingo da. Beraz, $acca$ hitza G_3 lengoaiakoa dela ziurtatu ahal izateko baldintzak ez dira betetzen.

4. ...

Gaizki bukatzen diren kasu gehiago ere badaude, baina kasu batean ondo bukatzen denez, $acca$ hitza G_3 lengoaiakoa da. Baina $bbba$ hitza hartzen badugu, kasu denak gaizki bukatuko dira eta hitza lengoaiakoa ez dela ziurtatu ahal izango dugu.

3.10.2.1 Piladun automaten bidez definigarriak ez diren lengoaiak

Piladun automatetan bakarrik bi eragiketa buru daitezke pilan: elementu bat ipini gailurrean eta gailurreko elementua kendu. Beraz, gailurrean ez dauden elementuak ezin dira hartu. Pilako gailurrean ez dagoen x elementu bat hartzeko era bakarra lehenengo bere gainean daudenak kentzea da. Pilak duten muga hori dela eta, piladun automatak ere kalkulu-gaitasun mugatua dute.

Jarraian erakusten diren lengoaiak ezin dira definitu piladun automaten bidez eta ondorioz ez dira kontesturik gabeko lengoaiak:

- $F_1 = \{w \in A^* \mid w = a^n b^n c^n \wedge n \geq 0\}$. Adibidez, ε, abc, $aabbcc$ eta $aaabbbccc$ hitzak F_1 lengoaiakoak dira. Baina, esate baterako, $abbccc$ eta aab ez dira F_1 lengoaiakoak. Lengoaia hau G_2 lengoaiaren antzekoa da: G_2 lengoaian a kopurua b kopuruarekin edo c kopuruarekin konparatu behar zen, baina ez biekin. Baina F_1 lengoaian a kopurua b kopuruarekin eta c kopuruarekin konparatu behar da, biekin. Beraz, a sinboloaren agerpenak pilan gordetzen baditugu eta gero b-ak irakurtzean pilako a horiek kenduz joaten bagara, c-ak iristean a denak ezabatuta egongo dira pilatik eta a kopurua zein izan den ez dugu jakingo.

- $F_2 = \{w \in A^* \mid w = a^i b^j c^k \wedge 0 \leq i \leq j \leq k\}$. Adibidez, ε, abc, $aabbcc$, $bbccc$ eta $abbccc$ hitzak F_2 lengoaiakoak dira, baina $aaabbccc$ eta aab ez. Lengoaia honen arazoa F_1 lengoaiako arazoaren antzekoa da.

- $F_3 = \{w \in A^* \mid \exists v(v \in A^* \wedge w = vv)\}$. Adibidez,$\varepsilon$, $\underline{aa}aa$, $\underline{abc}abc$, $\underline{ab}ab$ eta $\underline{ccbaba}ccbaba$ hitzak F_3 lengoaiakoak dira baina $aaca$, $acca$ eta aab ez. Pilan sinboloak gordetzeko ordenagatik eta pilako gailurreko elementua bakarrik irakur daitekeenez, hitz bat bi azpihitz berdinez osatuta al dagoen ala ez jakiteko erarik ez dago piladun automaten bidez.

3.11.

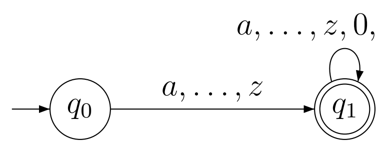
Lengoaia erregularren eta kontesturik gabeko lengoaien aplikazioak

Lengoaia erregularrak eta kontesturik gabeko lengoaiak, eta ondorioz, automata finituak eta piladun automatak, konputagarritasuna aztertzeko erabil daitezke teoria mailan, baina beste arlo batzuetan ere erabil daitezke.

Lengoaia erregularrak (eta automata finituak) programazio-lengoaietan erabiltzen diren aldagaien eta izenen egitura kontrolatzeko erabil daitezke (analisi lexikoa). Adibidez, programazio-lengoaia bateko aldagaien izenek letra minuskula batez hasi behar badute eta gero letra minuskulez eta digituz osatuta egon badaitezke, aldagaien izenek $A = \{a, b, \ldots, z, 0, 1 \ldots, 9\}$ alfabetoaren gainean definitutako honako V lengoaia hau osatzen dute:

$$\begin{aligned} &H_1 = \{w \in A^* \mid w = a \vee w = b \vee \cdots \vee w = z\} \\ &H_2 = A^* \\ &V = H_1 H_2 \end{aligned}$$

V lengoaiarentzat honako AFEDa izango genuke:

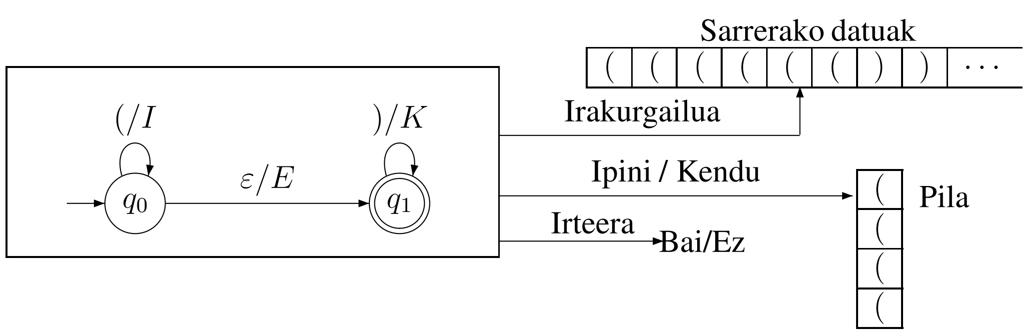


Bestalde, kontesturik gabeko lengoaiak konpiladoreak eraikitzeko erabiltzen dira (analisi sintaktikoa). Adibide bat parentesiak ondo al dauden kontrolatzen duen piladun automatarena izango litzateke. Alfabetoa $A = \{(,)\}$ izango litzateke eta kontesturik gabeko lengoaia honako hau izango litzateke:

$$L = \{w \in A^* \mid w = (^n)^n \wedge n \geq 0\}$$

Beraz, adibidez $\varepsilon, (), (()), ((()))$ eta $(((())))$ hitzak lengoaia horretakoak dira eta $((,)(, ()()$ eta $(()))$ hitzak ez dira lengoaia horretakoak.

Lengoaia horri dagokion piladun automata honako hau izango litzateke:



'(' sinboloak pilan ipini edo gordeko dira eta gero ')' sinbolo bakoitzeko '(' bat kenduko da pilatik.