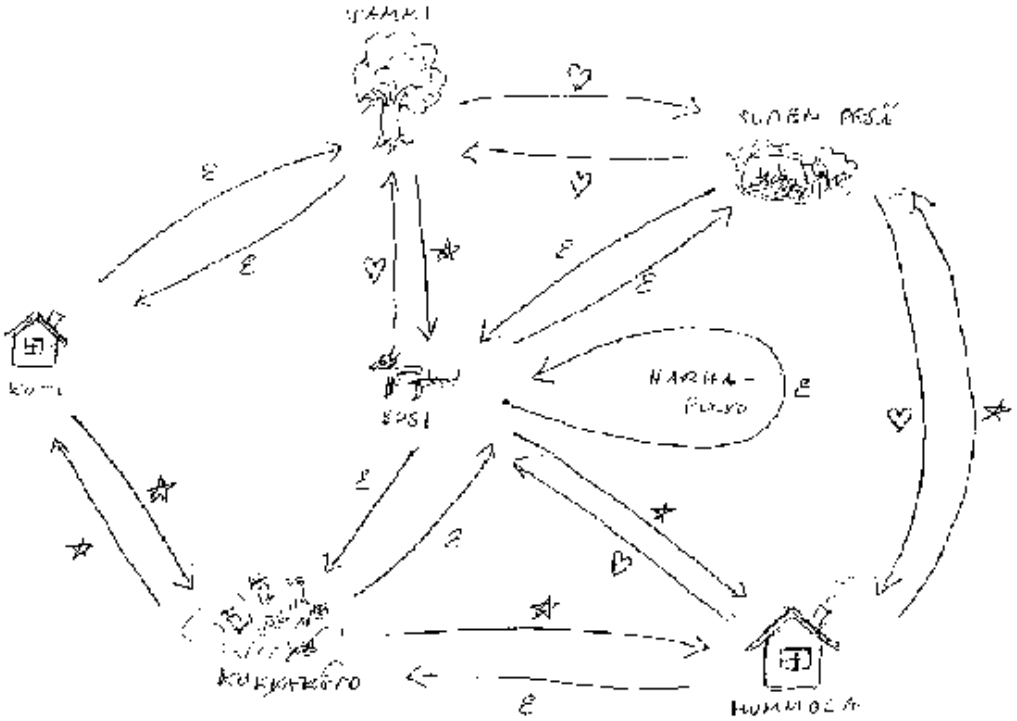
3.4.3 ϵ-automaatti

Ongelma: Punahilkkapeli

Punahilkka haluaa vierailla Mummonsa luona. Metsässä risteilee useita polkuja, joista osaa voi kulkea vapaasti, mutta osaa vartioi peikko. Peikot vaativat läpipääsystä tullia yhden piparkakun – joko täden- tai sydämenmuotoisen mieltymyksensä mukaan. Äiti on kuitenkin antanut Punahilkalle vain yhden tähtipiparkakun. Miten pitkälle Punahilkka pääsee? Entä yhdellä sydänpiparkakulla?



Kiltti susi kutsuu Punahilkan pesäänsä ja tarjoaa yhden sydänpiparkakun. Nyt Punahilkka pääsee Mummolaan, mutta miten hän pääsee takaisin kotiin?

ϵ-automaatti

- Äärellisten automaattien mallin laajennus
- Epädeterministinen äärellinen automaatti, jossa sallitaan *ϵ-siirtymät* ts. siirtymät tyhjällä merkkijonolla
- Hyödyllinen työkalu monimutkaisten järjestelmien suunnittelussa!
- Voidaan käyttää automaattien yhdistämiseen: $L(M_1) \bigcup L(M2)$ (kielten yhdiste), $L(M_1)L(M2)$ (kielten katenaatio), $(L(M_1))^*$ (kielen sulkeuma)
- Vastaa säännöllisissä lausekkeissa esiintyvää tyhjää merkkijonoa ϵ
- Huom! Kaikki, mikä voidaan kuvata ϵ-siirtymien avulla, voidaan kuvata ilmankin!

Formaali määrittely

Määritelmä: ϵ-automaatti on viisikko $M = (Q, \Sigma, \delta, q_0, F)$, missä siirtymäfunktio δ on kuvaus

$$\delta : Q \times (\Sigma \cup \{\epsilon\}) \to \mathcal{P}(Q)$$

- Tilanne (q, w) voi johtaa suoraan tilanteeseen (q', w')

  $$(q, w) \underset{M}{\vdash} (q', w'),$$

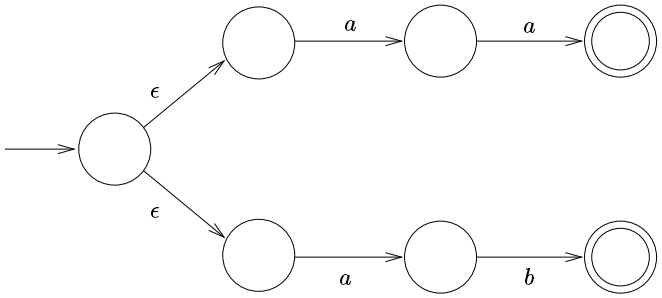
  jos

  (i) $w = aw'$ $(a \in \Sigma)$ ja $q' \in \delta(q, a)$; tai
  (ii) $w = w'$ ja $q' \in \delta(q, \epsilon)$

- Muut määritelmät kuten tavallisilla epädeterministisillä äärellisillä automaateilla
- *Lemma:* Olkoon $A = L(M)$ jollakin ϵ-automaatilla M. Tällöin on olemassa myös ϵ-siirtymätön epädeterministinen automaatti $\widehat{M}$, jolla $A = L(\widehat{M})$.
  Todistus: Olkoon $M = (Q, \Sigma, \delta, q_0, F)$ jokin ϵ-automaatti. Automaatti $\widehat{M}$ toimii muuten kuten M, mutta simuloi kunkin askelensa yhteydessä myös kaikki M:n mahdolliset ϵ-siirtymät.

  Formaalisti:

  $$\widehat{M} = (Q, \Sigma, \hat{\delta}, q_0, \widehat{F}),$$



Kuva 3.3: Kielen $\{aa, ab\}$ tunnistava ϵ-automaatti.

missä

$$\hat{\delta}(q,a) = \{q' \in Q \mid (q,a) \underset{M}{\vdash}^* (q',\epsilon)\};$$

$$\widehat{F} = \begin{cases} F \cup \{q_0\}, & \text{jos } (q_0,\epsilon) \underset{M}{\vdash}^* (q_f,\epsilon) \text{ jollain } q_f \in F; \\ F, & \text{muuten.} \end{cases}$$ □

3.4.4 ϵ-siirtymien poisto

- Muodosta siirtymä $q \xrightarrow{a} aq'$, jos $(q,a) \underset{M}{\vdash}^* (q',\epsilon)$

  esim. $q \overset{\epsilon}{\rightarrow} q_1 \overset{a}{\rightarrow} q_2 \overset{\epsilon}{\rightarrow} q'$

- Lisää alkutila q_0 lopputiloihin, jos $(q_0,\epsilon) \underset{M}{\vdash}^* (q_f,\epsilon)$ $(q_f \in F)$

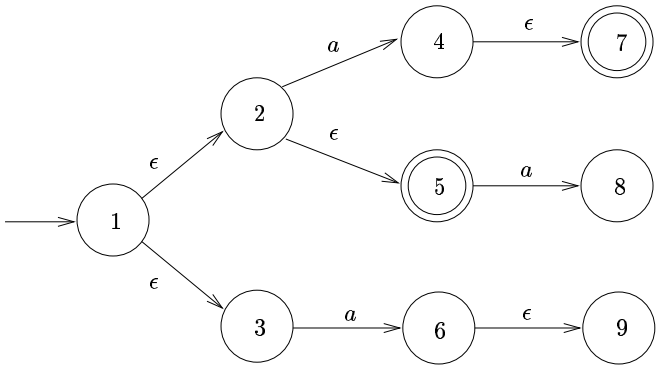
  esim. $q_0 \overset{\epsilon}{\rightarrow} q_1 \overset{\epsilon}{\rightarrow} q_f$

- Intuitiivisesti: seuraa kaikkia q:sta lähteviä polkuja, jotka sisältävät a:n, kunnes vastaan tulee jokin epätyhjä merkki (toinen a tai b). Kaikki matkan varrella olleet tilat ovat tilan q seuraajia a:lla.

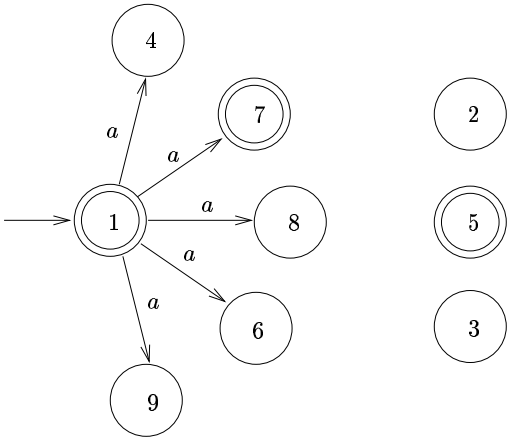
- Uusi (joukkoarvoinen) siirtymäfunktio $\delta(q,a)$ saadaan mekaanisesti kolmessa vaiheessa. Olkoon $S(q,a)$ q:n *a-sulkeuma* merkillä a (ts. tilat, joihin pääsee a:lla, s.e. polulla saa lisäksi esiintyä ϵ:ia). $S(q,\epsilon)$ on siis q:n ϵ-sulkeuma (tilat, joihin pääsee ϵ:eilla). Huom! $q \in S(q,\epsilon)$ $\forall q$.

  1. Laske $S(q,\epsilon) = \{q' | (q,\epsilon) \underset{M}{\vdash}^* (q',\epsilon)\}$; /* mihin pääsee pelkillä ϵ:eilla q:sta */
  2. Laske $S(q,a) = S(S(q,\epsilon),a)$; /* mihin 1. askeleen tulosjoukosta pääsee a:lla */

3. $S(q,a) := S(S(q,a),\epsilon)$; /* Lisää vielä ne, joihin 2. askeleen tulosjoukosta pääsee ϵ:eilla */
   return $S(q,a)$;



Kuva 3.4: ϵ-automaatti M.



Kuva 3.5: Vastaava ϵ-siirtymätön epädeterministinen automaatti $\widehat{M}$. Irralliset tilat 2, 5 ja 6 katoavat (sulautuvat tilaan 1).

- Huom! muodostettavissa ϵ-automaateissa aina yksikäsitteiset alku- ja lopputilat.

3.4.5 Lausekeautomaatit

Määritellään vielä yksi äärellisten automaattien laajennus: lausekeautomaatti:

Määritelmä: Merk. RE_Σ = aakkoston Σ säännöllisten lausekkeiden joukko. *Lausekeautomaatti* on viisikko

$$M = (Q, \Sigma, \delta, q_0, F),$$

missä siirtymäfunktio δ on äärellinen kuvaus

$$\delta : Q \times \mathrm{RE}_\Sigma \to \mathcal{P}(Q)$$

(so. $\delta(q, r) \neq \emptyset$ vain äärellisen monella parilla $(q, r) \in Q \times \mathrm{RE}_\Sigma$).

- Yhden askelen tilannejohto määritellään:

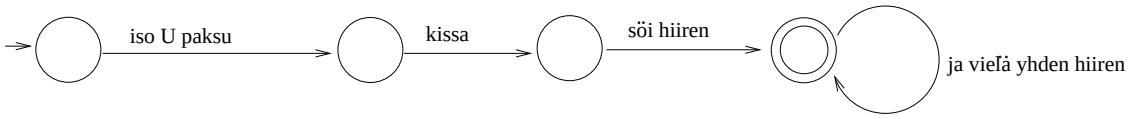
  $$(q, w) \underset{M}{\vdash} (q', w')$$

  jos on $q' \in \delta(q, r)$ jollakin sellaisella $r \in \mathrm{RE}_\Sigma$, että $w = zw'$, $z \in L(r)$. Muut määritelmät samat kuin ϵ- epädeterministisellä automaatilla

- esim. Automaatti, joka tunnistaa kielen $L(M) = \{$(iso $\bigcup$paksu) kissa söi hiiren (ja vielä yhden hiiren)*. Nyt

  $$(q_0, isokissa) \underset{M}{\vdash} (q_1, kissa)$$

  .



Kuva 3.6: Lausekeautomaatti, joka tunnistaa kielen $L(M)$ = {iso $\bigcup$paksu) kissa söi hiiren (ja viela yhden hiiren)*.

3.5 Äärelliset automaatit ja säännölliset kielet

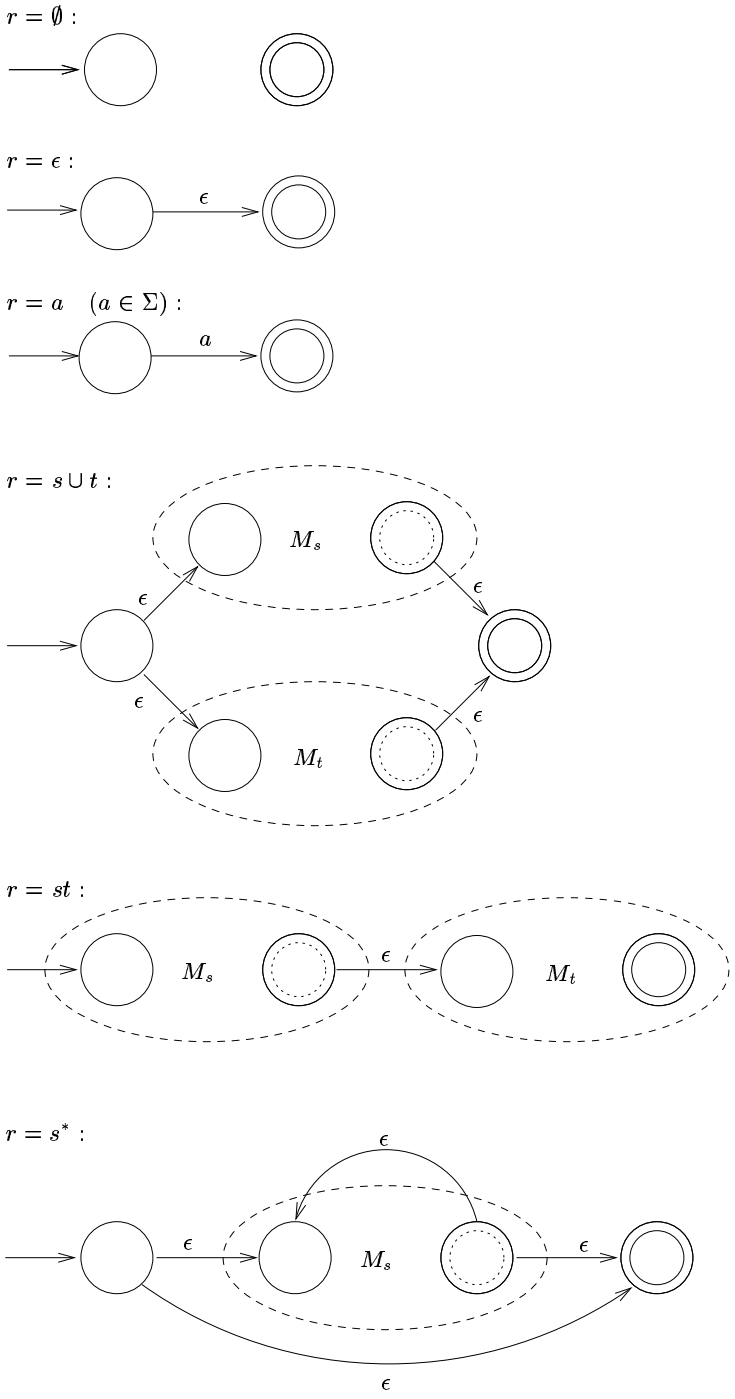
- Osoitetaan seuraava tulos:
  Kieli on säännöllinen $\Leftrightarrow$ Kieli voidaan tunnistaa äärellisellä automaatilla.

- Todistuksen idea:

  1. Kieli $L(r)$ on säännöllinen $\Rightarrow$ $L(r)$ voidaan tunnistaa äärellisellä automaatilla M:
     - Muodostetaan säännöllistä lauseketta r vastaava ϵ-automaatti
     - Tällöin on olemassa vastaava ϵ-siirtymätön epädeterministinen automaatti
     - Haluattessa epädeterministinen automaatti voidaan vielä determinisoida (ja minimoida)
  2. Kieli $L(M)$ voidaan tunnistaa äärellisellä automaatilla $M \Rightarrow L(M)$ on säännöllinen kieli:
     - Tarkastellaan äärellisten automaattien laajennosta, *lausekeautomaatteja* – jos väite pätee lausekeautomaateille, se pätee myös tavallisille äärellisille automaateille (jotka ovat niiden erikoistapaus)
     - Redusoidaan lausekeautomaatti korkeintaan 2-tilaiseksi automaatiksi, josta voidaan lukea suoraan vastaava säännöllinen lauseke

- *Lause:* Jokainen säännöllinen kieli voidaan tunnistaa äärellisellä automaatilla.
  Todistus:

  - Muodostetaan mielivaltaista säännöllistä lauseketta r vastaava ϵ-automaatti M_r, jolla $L(M_r) = L(r)$ (ks. kuva)
  - M_r:stä voidaan poistaa ϵ-siirtymät edellisen lemman mukaisesti, ja tarvittaessa voidaan syntyvä epädeterministinen automaatti determinisoida aiemmin esitetyn konstruktion avulla. □



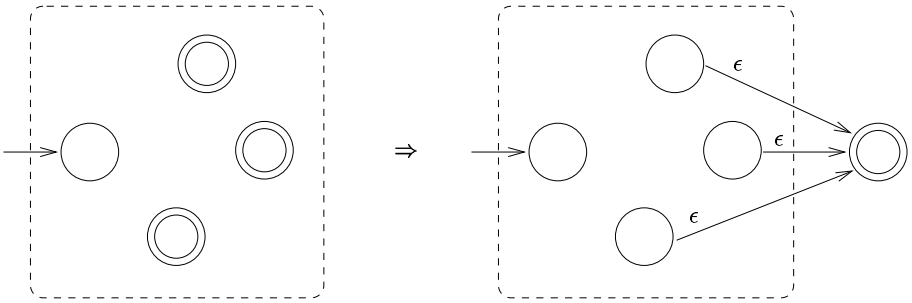
Kuva 3.7: Lauseketta r vastaavan ϵ-automaatin M_r muodostaminen.

- *Lause:* Jokainen äärellisellä automaatilla tunnistettava kieli on säännöllinen.
  Todistus: Osoitetaan, että jokainen lausekeautomaatilla tunnistettava kieli on säännöllinen.

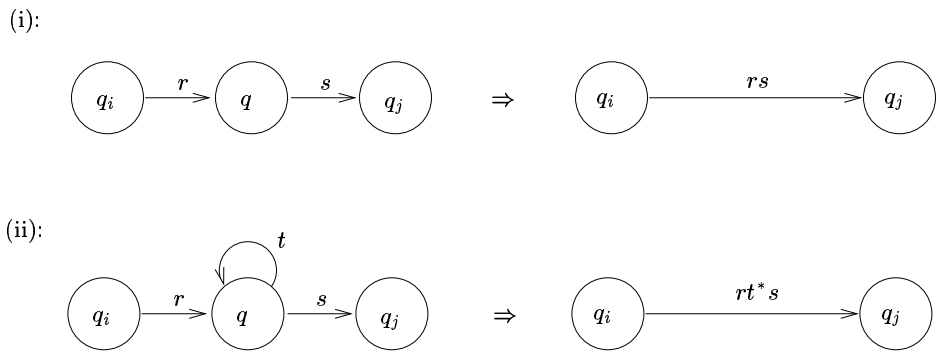
  Idea: Redusoidaan lausekeautomaatti s.e. siitä voidaan lukea vastaava säännöllinen lauseke:

  - Yhdistetään M:n lopputilat yhdeksi ϵ-siirtymillä (ks. kuva)
  - while (muita kuin alku- ja lopputiloja) {
    * valitse q, $q \neq q_0$, $q \neq q_f$ $(q_f \in F)$
    * poista q reitiltä $q_i \rightarrow q \rightarrow q_j$ $q_i = q$:n edeltäjätila ja $q_j = q$:n seuraajatila) reduktiosäännöillä (muista käydä läpi kaikki q:ta sisältävät polut!)
    * yhdistä rinnakkaiset siirtymät

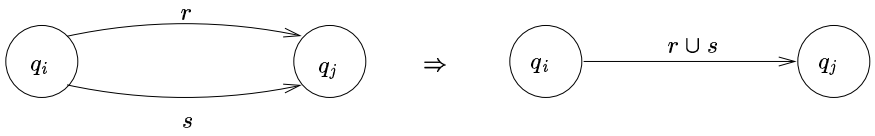
    }
    (ks. kuvat)



Kuva 3.8: Lausekeautomaatin lopputilojen yhdistäminen.

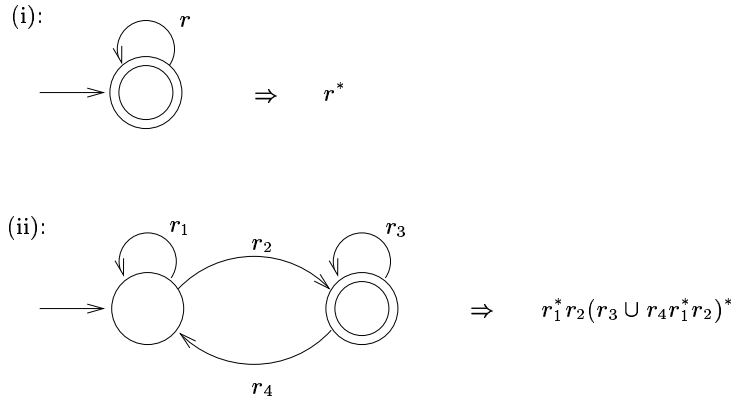


Kuva 3.9: Tilan poistaminen lausekeautomaatista.



Kuva 3.10: Rinnakkaisten siirtymien yhdistäminen lausekeautomaatissa.

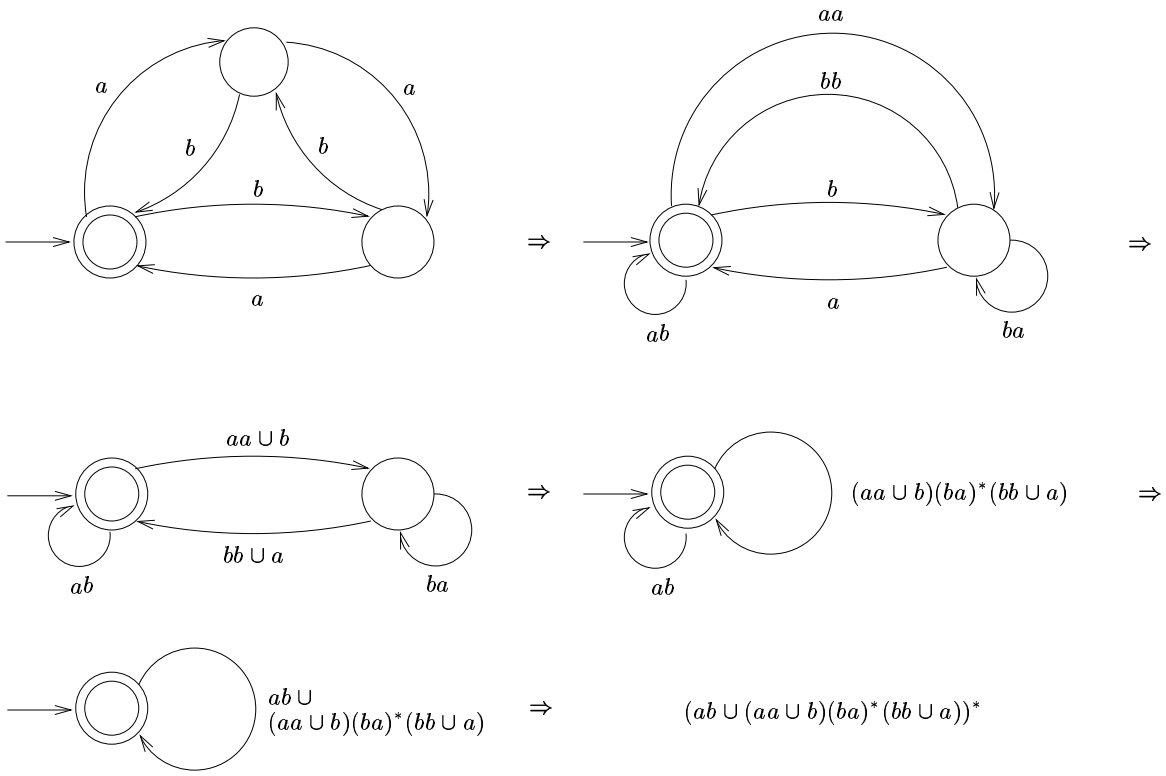
– Tiivistyksen päättyessä jäljellä olevaa enintään 2-tilaista automaattia vastaava säännöllinen lauseke muodostetaan kuten kuvassa



Kuva 3.11: Säännöllisen lausekkeen muodostaminen redusoidusta lausekeautomaatista.

□

- Esimerkki:



Kuva 3.12: Esimerkki: säännöllisen lausekkeen muodostaminen äärellisestä automaatista.

3.5.1 Hauska lisätieto: Säännöllisen kielen sulkeumaominaisuudet

Automaattien avulla voidaan osoittaa seurava tulos:

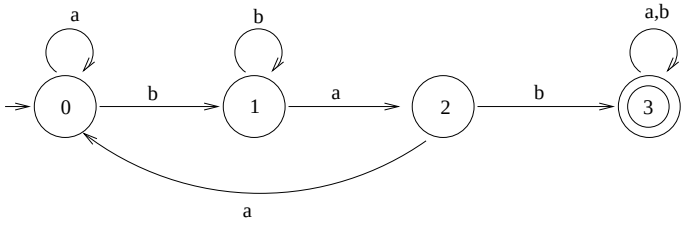
Lause: Olkoon L_1 ja L_2 säännölisiä kieliä. Tällöin myös

1. $L_1 \bigcup L_2$ (kielten yhdiste)
2. $L_1 \bigcap L_2$ (kielten leikkaus)
3. $L_1 L_2$ (kielten katenaatio)
4. $\overline{L_1} = \Sigma^* \setminus L_1$ (kielen komplementti)
5. $(L_1)^*$ (kielen sulkeuma)
6. $(L_1)^R$ (käänteiskieli, jossa kaikki L_1:n sanat on kirjoitettu takaperin)

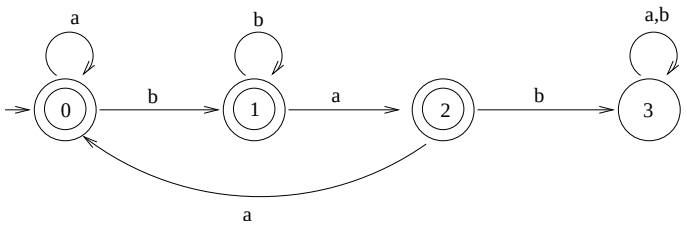
ovat säännöllisiä.

Todistus: Harjoitustehtävä!

Esim.



Kuva 3.13: Automaatti M, joka tunnistaa kielen $L(M) = \{w \in \{a,b\}^* | w$ sisältää merkkijonon $bab\}$.

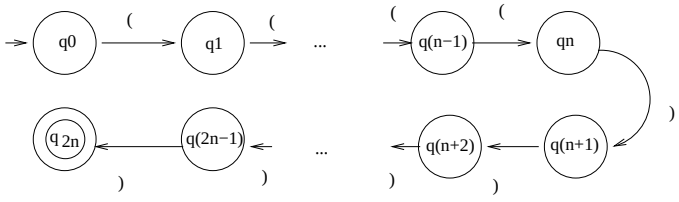


Kuva 3.14: Komplementtiautomaatti $\overline{M}$, joka tunnistaa kielen $L(\overline{M}) = \{w \in \{a,b\}^* | w$ ei sisällä merkkijonoa $bab\}$.

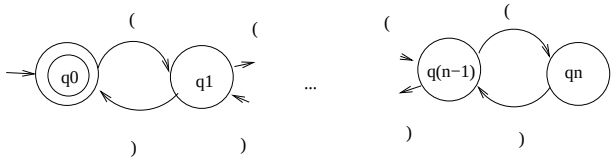
3.6 Säännöllisten kielten rajoituksista

- Minkä tahansa aakkoston formaaleja kieliä (päätösongelmia) on ylinumeroituva määrä, mutta säännöllisiä lausekkeita vain numeroituva määrä ⇒ kaikki kielet eivät voi olla säännöllisiä
- Perusrajoitus: äärellisillä automaateilla on vain rajallinen "muisti"
- Äärelliset kielet ovat aina säännöllisiä
- Milloin on ääretön kieli säännöllinen?
  - oltava jokin toistuva rakenne (sulkeuma)
  - vastaavassa automaatissa silmukka
- Esim. tasapainoisten sulkujonojen muodostama kieli $L_{\text{match}} = \{(^k)^k \mid k \geq 0\}$ ei ole säännöllinen, eli sitä ei voi tunnistaa äärellisellä automaatilla.

a)



b)



  - tunnistaa tasan n sulkuparia sisältävän merkkijonon
  - entä jos enemmän tai vähemmän sisäkkäisiä sulkuja?
  - esim. $a^{n-1} \in L_{\text{match}}$, $a^{n+1} \in L_{\text{match}}$, mutta automaatti ei tunnista niitä

- Entä kieli $\{a^k b^k \mid k \geq 0\}^*$?

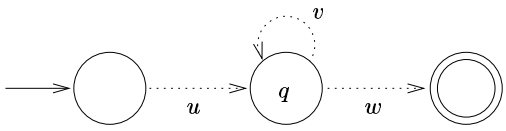
  $\Rightarrow$ ei voida tunnistaa äärellisellä automaatilla (eikä siten myöskään mv. aritmeettisia lausekkeita)

- "Pumppauslemma" formalisoi tämän rajallisen muistin idean

- Idea: mitä tahansa annetun säännöllisen kielen riittävän pitkää merkkijonoa voidaan "pumpata" keskeltä, ilman että kielen tunnistava äärellinen automaatti huomaa muutosta

- *Lemma*[Pumppauslemma]: Olkoon A säännöllinen kieli. Tällöin on olemassa $n \geq 1$ s.e. mikä tahansa $x \in A$, $|x| \geq n$, voidaan jakaa osiin $x = uvw$, $|uv| \leq n$, $|v| \geq 1$, ja $uv^i w \in A$ kaikilla $i = 0, 1, 2, \ldots$
  Todistus:

  - Olk. M jokin A:n tunnistava deterministinen äärellinen automaatti ja n = M:n tilojen määrä.
  - Tarkastellaan automaatin läpikäymiä tiloja sen tunnistaessa merkkijonoa $x \in A$, $|x| \geq n$:
    * Koska $|x| \geq n \Rightarrow$ täytyy kulkea jonkin tilan kautta (ainakin) kaksi kertaa (itse asiassa jo x:n n:ää ensimmäistä merkkiä käsitellessään)
    * Olk. q ensimmäinen tila, jonka automaatti toistaa x:ää käsitellessään.
    * Olk. $u = M$:n käsittelemä x:n alkuosa sen tullessa ensimmäisen kerran tilaan q
    * Olk. $v =$ se osa x:stä u:n jälkeen jonka M käsittelee ennen ensimmäistä paluutaan q:hun, ja
    * $w =$ loput x:stä
    * Tällöin on $|uv| \leq n$, $|v| \geq 1$, ja $uv^i w \in A$ kaikilla $i = 0, 1, 2, \ldots$ $\square$



Kuva 3.15: Merkkijonon $x = uvw \in A$ pumppaus.

- Huom! Pumppauslemma on kiinnostava vain äärettömille kielille. (Äärellisille kielille, jotka kaikki ovat säännöllisiä, voidaan valita $n = \max\{|x| \mid x \in L\}+1$, jolloin ei ole olemassa yhtään $x \in L$ s.e. $|x| \geq n$)

- Esim. Väite: Sulkulausekekieli

$$L = L_{\text{match}} = \{(^k)^k \mid k \geq 0\}.$$

on epäsäännöllinen.
Tod.: Vastaoletus: L on säännöllinen. $\Rightarrow \exists$ jokin $n \geq 1$, jonka pituisia L:n merkkijonoja voidaan pumpata.

Valitaan $x = (^n)^n$, jolloin $|x| = 2n > n$. Lemman mukaan x voidaan jakaa pumpattavaksi osiin $x = uvw$, $|uv| \leq n$, $|v| \geq 1$; siis on oltava

$$u = (^i,\ v = (^j,\ w = (^{n-(i+j)})^n, \quad \text{missä } i \leq n-1,\ j \geq 1.$$

Mutta esimerkiksi "0-kertaisesti" pumpattu merkkijono $uv^0w = (^i(^{n-(i+j)})^n = (^{n-j})^n$ ei kuulu kieleen L. Ristiriita. Siten L ei voi olla säännöllinen. □

- Pumppauslemmaa käytetään siis *epäsäännöllisyyden* osoittamiseen (Jos $\nexists n \geq 1$ s.e. ($\forall x \in A$ $|x| \geq n$ $\exists$ jako $x = uvw$, $|uv| \leq n$, $|v| \geq 1$, jolla $\forall i = 0, 1, ...$ $uv^iw \in A$), niin A ei ole säännöllinen $\Leftrightarrow$ Jos $\forall n \geq 1$ s.e. $\exists x \in A$ $|x| \geq n$ $\forall$ jaoille $x = uvw$, $|uv| \leq n$, $|v| \geq 1$ pätee $\exists i = 0, 1, ...$ $uv^iw \notin A$, niin A ei ole säännöllinen.)

- Huom: Pumppauslemma ei sano, että voimme valita u:n ja v:n haluamallamme tavalla!

- Todistusstrategia: Valitaan tarkasteltavan jonon pituudeksi esim. $2n$, jossa ensimmäiset n merkkiä muodostavat uv:n, muttei kiiinnitetä tarkkaan ottaen miten

- Esim. Väite: $L = \{a^kb^lc^{k+l} \mid k, l \geq 0\}$ ei ole säännöllinen.
Tod.: Vastaoletus: L on säännöllinen $\Rightarrow \exists n \geq 1$, jonka pituisia merkkijonoja voidaan pumpata. Valitaan $x = a^nb^lc^{n+l}$, jollakin $l \geq 0$. Nyt $|x| = 2n+2l > n$. Koska $|uv| \leq n$, uv koostuu vain a-merkeistä ja $m = |v| \geq 1$. Tällöin $uv^0w = a^{n-m}b^lc^{n+l} \notin L$. Ristiriita. Siten L ei voi olla

säännöllinen. □

- Esim. 2: Väite: $L(a^mb^nc^{m+n}$ ei ole säännöllinen.
Tod: Tarkastellaan merkkijonoja, joiden pituus vähintään $n = k + l$. Valitaan $x = a^kb^lc^{k+l}$, $|x| = 2k + 2l > n$. Nyt osa uv koostuu x:n korkeintaan $k + l$:stä ensimmäisestä merkistä (sisältää vain a:ta ja mahd. b:tä) ja v:n olt. vähintään 1-merkkinen. Kaksi jakovaihtoehtoa:

1. $u = a^i$, $v = a^{k-i}b^j$, $w = b^{l-j}c^{k+l}$, missä $i \neq k$ tai $j > 0$. Tällöin 0-kertaisesti pumpattu jono on $uv^0w{=}a^ib^{l-j}c^{k+l}$, joka kuuluu kieleen vain jos, $i = k$ ja $j = 0$, mikä ei mahd.
2. $u = a^kb^i$, $v = b^j$, $w = b^{l-i-j}c^{k+l}$, missä $j \geq 1$. $uv^0w = a^kb^ib^{l-i-j}c^{k+l} = a^kb^{l-j}c^{k+l}$, joka kuuluu kieleen vain, jos $j = 0$, mikä ei mahd.

$\Rightarrow$ Kieli ei siis säännöllinen.

- Esim. 3: Väite: Kieli $L_{pr} = \{1^p | p \text{ on alkuluku}\}$ ei ole säännöllinen.
  Tod.: Vastaoletus: L_{pr} on säännöllinen $\Rightarrow \exists$ jokin $n \geq 1$, jota pitempiä L_{pr}:n merkkijonoja voidaan pumpata.
  Valitaan $x = 1^p$, jollakin alkuluvulla $p \geq n + 2$. Pumppauslemman mukaan $x = uvw, |uv| \leq n, m = |v| \geq 1$. Nyt $|uv^{p-m}w| = |uw| + (p - m)|v| = p-m+(p-m)m = (m+1)(p-m)$. Tämä on yhdistetty luku, koska $m+1 \geq 2$ ja $p - m \geq n + 2 - m \geq 2$ (sillä $m = |v| \leq |uv| \leq n$). Siis $uv^{p-m}w \notin L_{pr}$. Ristiriita. $\square$

3.7 Ekskursio: Säännöllisten kielten sovelluksia

3.7.1 Hahmontunnistus

Tavoite: halutaan löytää tekstistä y hahmo x. Edellä tutustuimme jo yhteen tapaan ratkaista tämä ongelma äärellisten automaattien avulla. Laadimme äärellisen automaatin, joka tunnistaa hahmon x. Jos haluamme lisäksi sanan x sijainnin merkkijonossa, täytyy lisätä laskuri, joka pitää kirjaa luetuista merkeistä (siirtymistä), kunnes hahmo tunnistettu.

Äärellisiä automaatteja voidaan käyttää toisellakin tapaa hahmon tunnistukseen. Muodostetaan äärellinen automaatti tekstin y suffikseista. Automaatin kuhunkin kaareen (siirtymään) liittyy sekä kirjain että numero. Kunkin siirtymän yhteydessä katenoidaan kaaren kirjain luettujen merkkien jonoon. Lisäksi kasvatetaan sijaintilaskuria siirtymän ilmoittamalla askelmäärällä. Tuloksena saadaan pari (p, i), missä p on suffiksi i kertoo sen sijainnin (montako merkkiä luettava tekstin alusta, jotta kyseinen suffiksi löytyy).

Tällainen *suffiksiautomaatti* toimii itse asiassa tekstin hakemistorakenteena, jonka avulla voidaan etsiä mikä tahansa merkkijono tekstistä.

Muistathan myös, että säännöllisillä lausekkeilla voidaan helposti kuvata etsittäviä hahmoja, esim. tiedon haussa. Vastaava hakukone voidaan toteuttaa äärellisenä automaattina.

3.7.2 Viestinvälitysprotokollat äärellisinä automaatteina

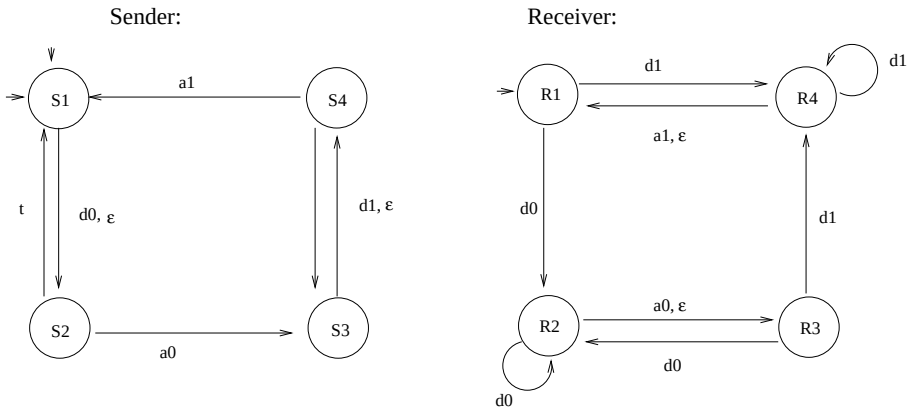
Myös erilaisia protokollia voidaan kuvata äärellisinä automaatteina. Tälla tavalla voidaan esimerkiksi verifioida, että jokin viestinvälitysprotokolla toimii oikein.

Tarkastellaan yksikertaista AB (alternating bit)-protokollaa, jossa lähettäjäprosessi S ja vastaanottajaprosessi R kommunikoivat *vuorosuuntaisen kanavan* kautta. Vuorosuuntaisessa kanavassa viestejä saa lähettää vain yhteen suuntaan kerrallaan. Kanava voi hukata tai vääristää sanomia, mutta viestien järjestys säilyy eikä kanava monista sanomia.

Koska kanavassa on kerrallaan vain yksi viesti, riittää viestien esittämiseen kaksi bittiä, 1 ja 0. Lähettäjä yrittää esnsin lähettää viestin $d0$ ja jos se saa siitä kuittauksen $a0$, se siirtyy lähettämään viestiä $d1$. Jos se saa $d1$:stäkin kuittauksen $a1$, se voi lähettää taas viestin, jota merkitään $d0$:lla jne. Lähettäjä ei kuitenkaan lähetä uutta viestiä ennen kuin se on saanut edellisestä kuittauksen. Jos kuittausta ei kuulu tietyn ajan t kuluessa se lähettää viestin uudestaan. Vastaanottaja puolestaan

lukee kaikki kanavasta tulevat viestit ja lähettää niistä kuittauksen, mahdollisesti useaankin kertaan (jos sama viesti tulee useita kertoja).

Lähettäjän ja vastaanottajan toiminta voidaan kuvata seuraavanlaisina epädeterministisinä (ϵ-) automaatteina.



3.7.3 Leksikaalinen analyysi

Leksikaalinen analyysi on kääntäjän osa, joka jakaa lähdekoodin loogisesti yhteenkuuluviin osiin (tokens). Tällaisia osia voivat olla esimerkiksi varatut sanat, tunnisteet, ym. Leksikaalinen analysaattori voidaan toteuttaa esim. UNIX-komentojen `lex` ja `flex` (edellisen GNU-versio) avulla. Lex- ja flex-komento saavat syötteenään listan säännöllisistä lausekkeista sekä ohjeet, mitä tehdä vastaavan osan kohdalla, ja tuottavat tämän pohjalta leksikaalisen analysaattorin kyseisille säännöille.

Syöte voisi olla esimerkiksi seuraavanlainen:

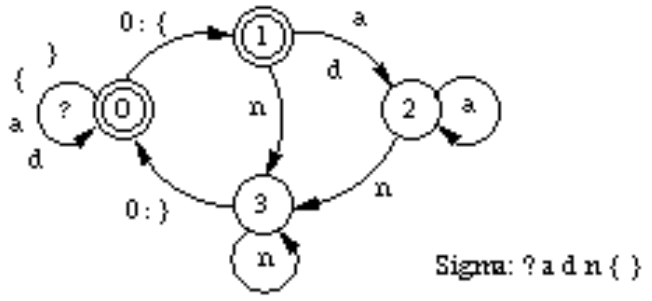
else	{return(ELSE); }
[A-Za-z][A-Za-z0-9]*	{code to enter the found identifier in the symbol table; return(ID); }
>=	{return(GE);}
=	{return(EQ); }
...	

Huom! Tällaisessa määrittelyssä varattu sana ”else” täsmää myös tunnisteen kuvaukseen. Ongelma voidaan ratkaista antamalla säännöille prioriteetijärjestys: valitaan ensimmäinen lauseke, johon merkkijono täsmää.

3.7.4 Luonnollisen kielen esiprosessointi

Luonnollisen kielen ymmärtäminen tai tiedon eristäminen tekstistä (joka ei pyrikään koko tekstin ymmärtämiseen, vaan etsimään siitä ainoastaan haluttua tietoa) on hyvin haastava tehtävä, johon säännölliset kielet eivät riitä. Niitä voidaan kuitenkin käyttää apuna tekstin esiprosessoinnissa samaan tapaan kuin ohjelmointikielen leksikaalisessa analyysissa.
Eräs lähestymistapa on muodostaa *transduktori* (engl. *transducer*), joka on eräänlainen uudelleenkirjoitava äärellinen automaatti. Laajennetaan esimerkiksi lausekeautomaattia siten, että jokaisen siirtymän yhteydessä automaatti sekä lukee merkkijonon että korvaa sen jollain toisella merkkijonolla (tai tyhjällä merkkijonolla). Tällaisella transduktorilla voidaan esimerkiksi lukea pois turhat täytesanat (kuten englannin artikkelit), tunnistaa prepositiot (esim. korvataan symbolilla PRE) tai keksiä heuristiikka erisnimien tunnistamiseen (korvataan sovitulla symbolilla).

Alla esimerkki transduktorista, joka pyrkii erottamaan (englanninkielisestä) lauseesta substantiivilmaukset (noun phrases), jotka ovat muotoa $[(d)a * n+]$, missä d on artikkeli, a on adjektiivi ja n on substantiivi. (Huom! Sanaluokat on siis jo tunnistettu jollain tapaa.)



Kuva 3.16: Transduktori, joka kirjoittaa sulut kaikkien tunnistamiensa substantiivilausekkeiden ympärille.