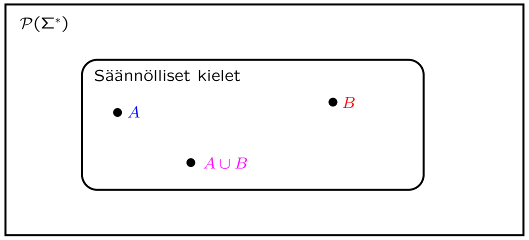
Yllä osoitettiin, että säännöllisten kielten joukko on suljettu yhdisteen suhteen, eli jos kielet A ja B ovat säännöllisiä, niin myös $A \cup B$ on. Tätä voi havainnollistaa seuraavalla kuvalla:



Tätä ei pidä sekoittaa edellisen kalvon kuvaan. Tärkeimmät erot ovat:

- Kaavion *pisteet* ovat kieliä (eivät merkkijonoja).
- *Alueet* ovat kieliluokkia (eivät yksittäisiä kieliä).
- *Koko laatikko* on kaikkien kielten (ei merkkijonojen) joukko. ($\mathcal{P}(X)$ on joukon X *potenssijoukko* eli kaikkien osajoukkojen joukko.)

Jatkossa osoitetaan myös

Lause 1.2: [Sipser Thm. 1.26] Säännöllisten kielten joukko on suljettu konkatenaation suhteen; ts. jos kielet A ja B ovat säännöllisiä, niin myös $A \circ B$ on.

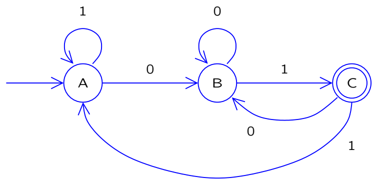
Todistaminen on kuitenkin edellistä lausetta hankalampaa. Oletaan $A = L(M_1)$ ja $B = L(M_2)$, ja halutaan muodostaa M, jolla $L(M) = A \circ B$. Luonnollinen ajatus olisi pistää automaatit M_1 ja M_2 peräkkäin. Tässä tulee ongelmaksi, milloin pitäisi siirtyä automaatista M_1 automaattiin M_2.

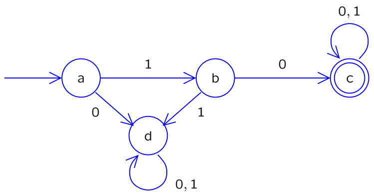
Erityisesti jollain $w \in A \circ B$ voi olla kaksi esitystä $w = uv = u'v'$, missä $u \in A$ ja $u' \in A$, ja $v \in B$ mutta $v' \notin B$. Jos tässä vielä u' on merkkijonon u alkuosa (eli $u = u'x$ jollain x), niin emme voi ilman muuta siirtyä automaattiin M_2 heti, kun M_1 pääsee hyväksyvään tilaan.

Esimerkki. Tarkastellaan aakkoston $\{0,1\}$ kieliä

$$\begin{aligned} A &= \{ v01 \mid v \in \{0,1\}^* \} \\ B &= \{ 10w \mid w \in \{0,1\}^* \} \end{aligned}$$

jotka sisältävät 01-loppuiset (A) ja 10-alkuiset (B) merkkijonot. Vastaavat automaatit ovat





A:n ja B:n konkatenaatio on selvästi kieli

$$A \circ B = \{ v0110w \mid v, w \in \{0,1\}^* \}$$

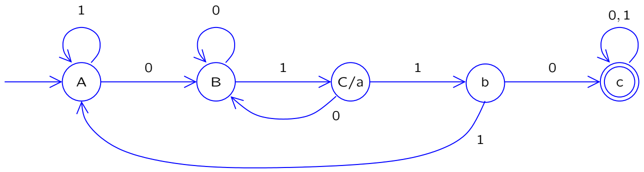
eli merkkijonon 0110 osajononaan sisältävät merkkijonot.

$A \circ B$:n tunnistaminen vaihtamalla automaatista toiseen ei toimi:

- Syötteellä 0110 oikea hetki vaihtaa on alkuosan 01 jälkeen.
- Syötteellä 01110110 taas sama vaihtohetki olisi virhe ja johtaisi väärään hylkäykseen.

Oikeaa vaihtohetkeä ei tiedetä siinä vaiheessa, kun vaihto pitäisi tehdä.

Kielen $A \circ B$ tunnistava automaatti

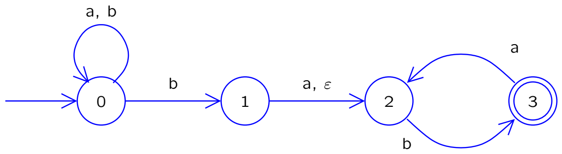


muistuttaa toki peräkkäin asetettuja alkuperäisiä automaatteja mutta erojakin on.

Yleisen säännön antaminen konkatenaatioautomaatin muodostamiselle tätä kautta on hankalaa. Pääsemme paljon helpommalla, kun yleistämme äärellistä automaattia sallimalla *epädeterminismin*.

Epädeterministiset äärelliset automaatit [Sipser s. 47–54]

Tarkastellaan esimerkkinä seuraavaa automaattia:



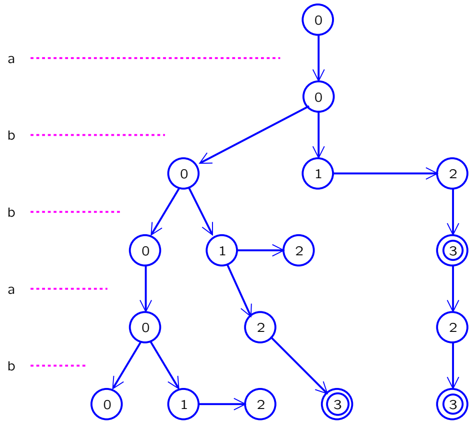
Tämä ei noudata edellä esitettyä formalismia:

- tilasta 0 pääsee merkillä b sekä tilaan 0 että tilaan 1,
- tilasta 1 on ε-siirtymä, jolla pääsee tilaan 2 "käyttämättä" yhtään syötemerkkiä ja
- joitain siirtymiä ei ole määritelty.

Aiottu tulkinta on, että annetulla syötteellä "kokeillaan" *kaikkia mahdollisia* siirtymävaihtoehtoja. Automaatti hyväksyy, jos *yksikin* näistä vaihtoehdoista päätyy hyväksyvään tilaan.

Tällaisen automaatin laskentaa voidaan havainnollistaa ryhmittämällä vaihtoehdot puuksi:

- esimerkkinä syöte `abbab`
- vaakanuolet kuvaavat ε-siirtymiä
- viimeisellä rivillä esiintyy (kahteenkin kertaan) hyväksyvä tila 3, joten automaatti hyväksyy merkkijonon `abbab`
- merkkijonon `abba` automaatti hylkää, koska puun syvyydellä 4 ei ole yhtään hyväksyvää tilaa



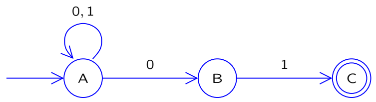
Kutsumme tällaista yleisempää automaattia *epädeterministiseksi äärelliseksi automaatiksi* (nondeterministic finite automaton, NFA). Aluksi esitetty perusversio on vastaavasti *deterministinen äärellinen automaatti* (deterministic finite automaton, DFA).

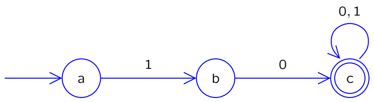
NFA:lle ei ole samalla lailla ilmeistä fyysistä toteutusmallia kuin DFA:lle. Se kannattaa mieltää lähinnä *kuvausformalismiksi*.

Näemme jatkossa, että mille tahansa NFA:lle voidaan muodostaa DFA, joka tunnistaa saman kielen. Siis epädeterminismi *ei* anna tässä mielessä lisää *ilmaisuvoimaa*.

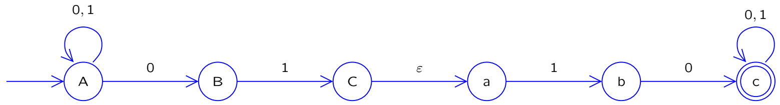
Epädeterminismiä käyttämällä esitystä voidaan kuitenkin usein selkeyttää ja yksinkertaistaa.

Esimerkki. Kielet $A = \{ v01 \mid v \in \{0,1\}^* \}$ ja $B = \{ 10w \mid w \in \{0,1\}^* \}$ tunnistavat NFA:t ovat yksinkertaisempia kuin aiemmin nähdyt DFA:t





ja konkatenaation $A \circ B = \{ v0110w \mid v, w \in \{0,1\}^* \}$ tunnistava NFA saadaan helposti yhdistämällä automaatit ε-siirtymällä:



Myöhemmin nähdään, että yleinen tapaus yhtä yksinkertainen.

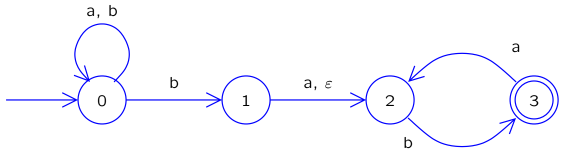
Mille tahansa aakkostolle Σ merkitsemme $\Sigma_\varepsilon = \Sigma \cup \{\,\varepsilon\,\}$. Muistetaan, että joukon A *potenssijoukkoa* eli kaikkien osajoukkojen joukkoa merkitään $\mathcal{P}(A)$.

Muodollisesti NFA on viisikko $(Q, \Sigma, \delta, q_0, F)$, missä

1. Q on äärellinen tilajoukko,
2. Σ on äärellinen aakkosto,
3. $\delta\colon Q \times \Sigma_\varepsilon \to \mathcal{P}(Q)$ on siirtymäfunktio,
4. $q_0 \in Q$ on alkutila ja
5. $F \subseteq Q$ on hyväksyvien tilojen joukko.

Erotukseksi DFA:sta siirtymäfunktio siis antaa yhden seuraajatilan sijaan *joukon* tiloja. Nämä tulkitaan "mahdollisiksi" seuraajatiloiksi.

Esimerkkiautomaattimme



formaali esitys on siis $(\{\,0,1,2,3\,\}, \{\,a,b\,\}, \delta, 0, \{\,3\,\})$, missä δ saadaan taulukosta

δ	a	b	ε
0	$\{\,0\,\}$	$\{\,0,1\,\}$	$\emptyset$
1	$\{\,2\,\}$	$\emptyset$	$\{\,2\,\}$
2	$\emptyset$	$\{\,3\,\}$	$\emptyset$
3	$\{\,2\,\}$	$\emptyset$	$\emptyset$

Määrittelemme nyt, että epädeterministinen äärellinen automaatti $(Q, \Sigma, \delta, q_0, F)$ *hyväksyy* merkkijonon w, jos jollain n voidaan valita jonot $(y_1, \ldots, y_n) \in \Sigma_\varepsilon^n$ ja $(r_0, \ldots, r_n) \in Q^{n+1}$, joilla

- $w = y_1 \ldots y_n$,
- $r_0 = q_0$,
- $r_{i+1} \in \delta(r_i, y_{i+1})$ kun $i = 0, \ldots, n-1$ ja
- $r_n \in F$.

Edellisen sivun automaatilla ja merkkijonolla abbab voidaan valita esim.

$$\begin{aligned} n &= 6 \\ (y_1, y_2, y_3, y_4, y_5, y_6) &= (\mathrm{a}, \mathrm{b}, \varepsilon, \mathrm{b}, \mathrm{a}, \mathrm{b}) \\ (r_0, r_1, r_2, r_3, r_4, r_5, r_6) &= (0, 0, 1, 2, 3, 2, 3), \end{aligned}$$

mistä nähdään, että automaatti hyväksyy merkkijonon.

Determinismi vs. epädeterminismi

Deterministisessä automaatissa siirtymä $\delta(q, a) = r$ tarkoittaa, että tilasta q syötteellä a mennään aina tilaan r.

Epädeterministisessä automaatissa siirtymä $r \in \delta(q, a)$ tarkoittaa, että tilasta q syötteellä a eräs mahdollisuus on mennä tilaan r.

Epädeterministinen automaatti hyväksyy, jos *on mahdollista* päätyä hyväksyvään tilaan. Huomaa epäsymmetria: jos jollain syötteellä voi päätyä sekä hyväksyvään että ei-hyväksyvään tilaan, niin syöte *hyväksytään*.

Epädeterminismi voidaan mieltää korkean tason kontrollirakenteeksi. Jos DFA-formalismia ajatellaan primitiivisenä ohjelmointikielenä, niin NFA-formalismi on sama kieli laajennettuna operaatiolla "kokeile kaikki vaihtoehdot". Seuraavaksi tarkastelemme, miten tämä kontrollirakenne voidaan toteuttaa.

Lähtökohtana on NFA:n simuloiminen. Simulaation ideana on laskentapuun (ks. sivu 61) läpikäynti leveyssuuntaisesti (taso kerrallaan).

Algoritmi pitää muuttujassa $\mathit{CurrentStates}$ listaa niistä tiloista, joissa se tällä hetkellä voisi olla (laskentapuun taso).

Kullakin aika-askelella $\mathit{CurrentStates}$ päivitetään:

1. haetaan saatua syötemerkkiä vastaavat seuraajatilat ja
2. lisätään mahdolliset ε-siirtymät.

Kun $R \subseteq Q$ on joukko tiloja, olkoon $E(R)$ niiden tilojen joukko, joihin päästään jostain joukon R tilasta suorittamalla 0 tai useampia ε-siirtymiä:

$$q \in E(R) \quad \Leftrightarrow \quad \begin{array}{l}\text{on olemassa } k \in \mathbb{N} \text{ ja tilat } r_0, r_1, \ldots, r_k \in Q \\ \text{joilla } r_0 \in R,\ r_k = q \text{ ja } r_i \in \delta(r_{i-1}, \varepsilon) \text{ kun } i = 1, \ldots, k.\end{array}$$

Siis erityisesti aina $R \subseteq E(R)$.

Saadaan seuraava algoritmi:

SimulateNFA$(Q, \Sigma, \delta, q_0, F)$
$\quad \mathit{CurrentStates} \leftarrow E(\{\, q_0 \,\})$
$\quad$ **while** syötettä riittää
$\qquad$ **do** lue seuraava syötemerkki a
$\qquad\quad \mathit{NextStates} \leftarrow \emptyset$
$\qquad\quad$ **for** $q \in \mathit{CurrentStates}$
$\qquad\qquad$ **do** $\mathit{NextStates} \leftarrow \mathit{NextStates} \cup \delta(q, a)$
$\qquad\quad \mathit{CurrentStates} \leftarrow E(\mathit{NextStates})$
$\quad$ **if** $\mathit{CurrentStates} \cap F \neq \emptyset$
$\qquad$ **then return** "accept"
$\qquad$ **else return** "reject"

Algoritmi SimulateNFA osoittaa, miten epädeterminismi voidaan implementoida "tulkkaamalla".

Algoritmi antaa myös idean, miten epädeterminismi voidaan implementoida "kääntämällä": annetusta NFA:sta voidaan muodostaa DFA, joka tunnistaa saman kielen.

(DFA:ta ei enää tarvitse kääntää mihinkään, sen toteutus "laitteistotasolla" on ilmeinen.)

Havainto: muuttujan $\mathit{CurrentStates}$ arvot kuuluvat potenssijoukkoon $\mathcal{P}(Q)$

$\Rightarrow$ muuttujalla $\mathit{CurrentStates}$ on korkeintaan $2^{|Q|}$ mahdollista arvoa

$\Rightarrow$ algoritmin vaatima muistitila on äärellinen

$\Rightarrow$ DFA-toteutus on mahdollinen.

Esitämme muunnoksen täsmällisemmin:

Lause 1.3: [Sipser Thm. 1.39] Mille tahansa NFA:lle on olemassa DFA, joka tunnistaa saman kielen.

Todistus: On siis annettu NFA $N = (Q, \Sigma, \delta, q_0, F)$, ja halutaan muodostaa DFA $M = (Q', \Sigma, \delta', q'_0, F')$, jolla $L(M) = L(N)$.

Edellä esitetyn mukaisesti idea on, että *yksi DFA:n tila* esittää *joukkoa NFA:n tiloja*. Siis valitsemme $Q' = \mathcal{P}(Q)$. Automaatin M tila annetulla hetkellä tulkitaan automaatin N mahdollisten tilojen joukoksi. Huomaa, että $|Q'| = 2^{|Q|}$.

Jos $r \in Q$ on mahdollinen tila nykyhetkellä, seuraavaksi luetaan merkki a, ja $s \in \delta(r, a)$, niin s on eräs mahdollinen tila seuraavalla hetkellä.
Siis jos $\delta'(R, a) = S$ (missä siis $R, S \subseteq Q$), niin joukossa S pitää olla kaikki ne tilat, jotka voivat seurata jotain joukon R tilaa merkillä a.

Esitetään sama formaalisti olettaen ensin, että automaatissa N ei ole ε-siirtymiä (eli $\delta(r, \varepsilon) = \emptyset$ kaikilla $r \in Q$). Siis $M = (Q', \Sigma, \delta', q_0', F')$, missä

1. $Q' = \mathcal{P}(Q)$,
2. jos $R \in Q'$ (eli $R \subseteq Q$) ja $a \in \Sigma$, niin

$$\begin{aligned}\delta'(R, a) &= \bigcup_{r \in R} \delta(r, a) \\ &= \{\, s \in Q \mid s \in \delta(r, a) \text{ jollain } r \in R, \,\}\end{aligned}$$

3. $q_0' = \{\, q_0 \,\}$ ja
4. $F' = \{\, R \in Q' \mid R \cap F \neq \emptyset \,\}$.

Sallitaan nyt automaatissa N myös ε-siirtymät. Kuten edellä, joukko $E(R)$ koostuu niistä tiloista, joihin pääsee jostain tilasta $r \in R$ tekemällä nolla tai useampia ε-siirtymiä.

Siirtymäfunktion määritelmä muutetaan muotoon

$$\delta'(R, a) = E\left(\bigcup_{r \in R} \delta(r, a)\right).$$

Alkutilaksi asetetaan

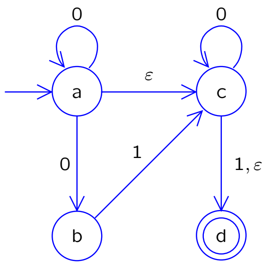
$$q_0' = E(\{\, q_0 \,\}).$$

Konstruktion perusteella on selvää, että $L(M) = L(N)$. □

Käänteinen suunta eli DFA:n muuntaminen NFA:ksi on triviaali. Siis DFA:t ja NFA:t ovat yhtä ilmaisuvoimaisia:

Korollaari 1.4: [Sipser Cor. 1.40] Kieli on säännöllinen, jos ja vain jos se voidaan tunnistaa *epädeterministisellä* äärellisellä automaatilla. □

Esimerkki 1.5: Muodostetaan DFA, joka vastaa seuraavaa NFA:ta:



Koska DFA:n alkutila on a, NFA:n alkutilaksi tulee $E(\{\,\mathsf{a}\,\}) = \{\,\mathsf{a}, \mathsf{c}, \mathsf{d}\,\}$.

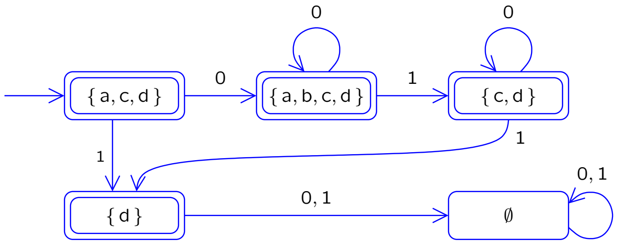
DFA:n tiloja ovat periaatteessa kaikki joukon $\{\,\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}\,\}$ osajoukot, joita on $2^4 = 16$ kappaletta. Meidän ei kuitenkaan kannata välittää tiloista, joihin ei koskaan päästä alkutilasta.

Ruvetaan siis purkamaan siirtymäfunktiota alkutilasta alkaen (seur. sivu).

$$\begin{aligned}
\delta'(\{\,\mathsf{a},\mathsf{c},\mathsf{d}\,\}, 0) &= E(\delta(a,0) \cup \delta(c,0) \cup \delta(d,0)) \\
&= E(\{\,\mathsf{a},\mathsf{b}\,\} \cup \{\,\mathsf{c}\,\} \cup \emptyset) = \{\,\mathsf{a},\mathsf{b},\mathsf{c},\mathsf{d}\,\} \\
\delta'(\{\,\mathsf{a},\mathsf{c},\mathsf{d}\,\}, 1) &= E(\delta(a,1) \cup \delta(c,1) \cup \delta(d,1)) \\
&= E(\emptyset \cup \{\,\mathsf{d}\,\} \cup \emptyset) = \{\,\mathsf{d}\,\} \\
\delta'(\{\,\mathsf{a},\mathsf{b},\mathsf{c},\mathsf{d}\,\}, 0) &= E(\delta(a,0) \cup \delta(b,0) \cup \delta(c,0) \cup \delta(d,0)) \\
&= E(\{\,\mathsf{a},\mathsf{b}\,\} \cup \emptyset \cup \{\,\mathsf{c}\,\} \cup \emptyset) = \{\,\mathsf{a},\mathsf{b},\mathsf{c},\mathsf{d}\,\} \\
\delta'(\{\,\mathsf{a},\mathsf{b},\mathsf{c},\mathsf{d}\,\}, 1) &= E(\delta(a,1) \cup \delta(b,1) \cup \delta(c,1) \cup \delta(d,1)) \\
&= E(\emptyset \cup \{\,\mathsf{c}\,\} \cup \{\,\mathsf{d}\,\} \cup \emptyset) = \{\,\mathsf{c},\mathsf{d}\,\} \\
\delta'(\{\,\mathsf{c},\mathsf{d}\,\}, 0) &= E(\delta(c,0) \cup \delta(d,0)) = E(\{\,\mathsf{c}\,\} \cup \emptyset) = \{\,\mathsf{c},\mathsf{d}\,\} \\
\delta'(\{\,\mathsf{c},\mathsf{d}\,\}, 1) &= E(\delta(c,1) \cup \delta(d,1)) = E(\{\,\mathsf{d}\,\} \cup \emptyset) = \{\,\mathsf{d}\,\} \\
\delta'(\{\,\mathsf{d}\,\}, 0) &= E(\delta(d,0)) = E(\emptyset) = \emptyset \\
\delta'(\{\,\mathsf{d}\,\}, 1) &= E(\delta(d,1)) = E(\emptyset) = \emptyset
\end{aligned}$$

Voidaan lopettaa, koska uusia tiloja ei enää löydy. (Määritelmistä seuraa, että aina $\delta'(\emptyset, a) = \emptyset$.)

DFA:n hyväksyviä tiloja ovat kaikki, jotka sisältävät NFA:n hyväksyvän tilan d, eli tässä tapauksessa kaikki muut kuin $\emptyset$.



□

Epädeterminismin sovelluksia

Näemme pian, että epädeterminismi yksinkertaistaa huomattavasti monia automaattikonstruktioita. Sillä on käyttöä tietojenkäsittelytieteessä laajemminkin, esim.

Rinnakkaisjärjestelmät: Asynkronisessa laskennassa eri prosessorien suoritusjärjestys ei ole kontrolloitavissa. Ajattelemme siis, että suoritusjärjestys määräytyy "epädeterministisesti" ja vaadimme esim. että järjestelmä ei saa lukkiutua *millään* suoritusjärjestyksellä.

Laskennan vaativuus: Monille tärkeille etsintä- ja optimointiongelmille on helppo esittää epädeterministinen ratkaisu, mutta epädeterminismin *tehokas* simuloiminen deterministisellä (=oikealla) tietokoneella on avoin ongelma.

Vrt. edellä: NFA:n muuntaminen DFA:ksi saattaa aiheuttaa tilojen lukumäärän eksponentiaalisen kasvun.

Säännöllisten kielten sulkeumaominaisuudet [Sipser s. 58–63]

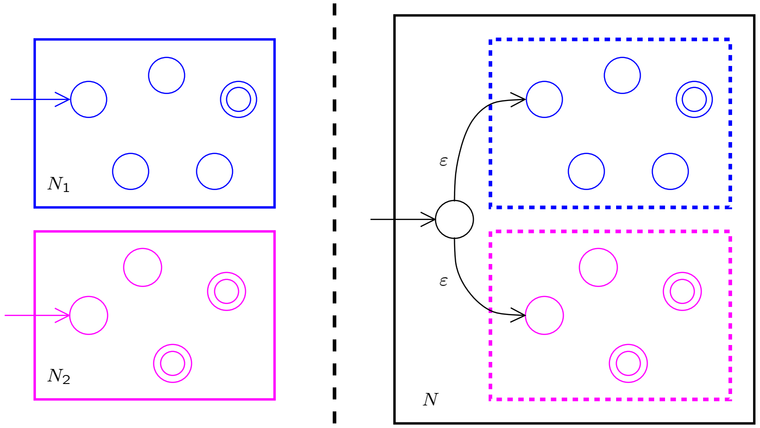
Osoitamme nyt, että säännöllisten kielten joukko on suljettu yhdisteen, konkatenaation ja tähtioperaation suhteen. Toisin sanoen jos A ja B ovat säännöllisiä, niin myös $A \cup B$, $A \circ B$ ja A^* ovat.

Kieli on määritelmän mukaan säännöllinen, jos se voidaan tunnistaa deterministisellä äärellisellä automaatilla. Todistimme juuri (korollaari 1.4), että voimme yhtä hyvin käyttää epädeterministisiä automaatteja. Katsotaan ensin, miten tämä yksinkertaistaa jo esitetyn tapauksen $A \cup B$ (lause 1.1) todistusta.

Lause 1.6: [Sipser Thm. 1.45] Jos A ja B ovat säännöllisiä, myös $A \cup B$ on.

Todistus: Oletetaan siis annetuksi NFA:t N_1 ja N_2, joilla $A = L(N_1)$ ja $B = L(N_2)$. Muodostetaan NFA N, jolla $L(N) = A \cup B$.

Konstruktion periaate on seuraava:



Muodollisemmin olkoon $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ ja $N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$. Voidaan tietysti olettaa, että $Q_1 \cap Q_2 = \emptyset$. Nyt $N = (Q, \Sigma, \delta, q_0, F)$, missä

1. $q_0 \notin Q_1 \cup Q_2$ on uusi tila
2. $Q = \{ q_0 \} \cup Q_1 \cup Q_2$
3. δ toteuttaa

$$\begin{aligned}
\delta(q, a) &= \delta_1(q, a) && \text{kun } q \in Q_1 \text{ ja } a \in \Sigma_\varepsilon \\
\delta(q, a) &= \delta_2(q, a) && \text{kun } q \in Q_2 \text{ ja } a \in \Sigma_\varepsilon \\
\delta(q_0, \varepsilon) &= \{ q_1, q_2 \} \\
\delta(q_0, b) &= \emptyset && \text{kun } b \neq \varepsilon
\end{aligned}$$

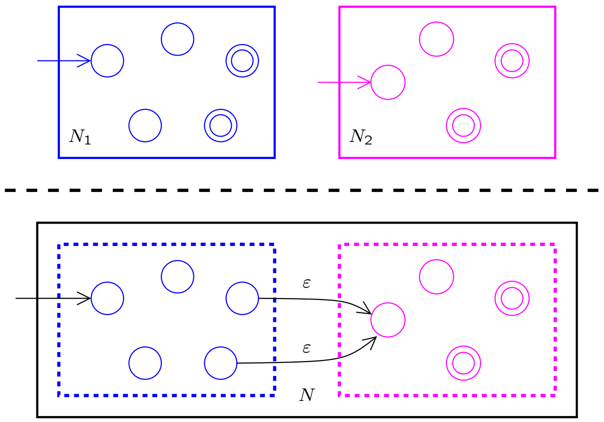
4. $F = F_1 \cup F_2$.

Selvästi N tunnistaa kielen $A \cup B$. □

Sovelletaan samaa ideaa konkatenaatioon.

Lause 1.7: [Sipser Thm. 1.47] Jos A ja B ovat säännöllisiä, myös $A \circ B$ on.

Todistus: Periaatepiirros on nyt seuraava:



Muodollisesti olkoon taas $A = L(N_1)$ ja $B = L(N_2)$, missä $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$, $N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ ja $Q_1 \cap Q_2 = \emptyset$. Muodostetaan $N = (Q, \Sigma, \delta, q_0, F)$, missä

1. $Q = Q_1 \cup Q_2$,
2. $q_0 = q_1$,
3. $F = F_2$ ja
4. kun $q \in Q$ ja $a \in \Sigma_\varepsilon$, niin $\delta(q, a)$ on seuraava:

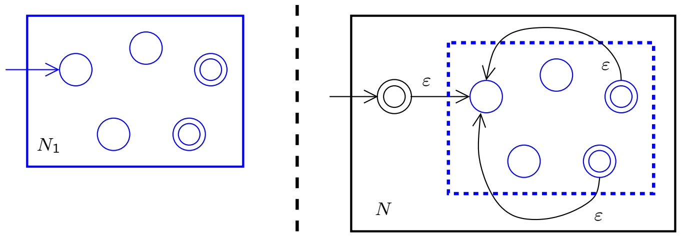
$$\begin{aligned}
\delta(q, a) &= \delta_1(q, a) && \text{kun } q \in Q_1 - F_1 \\
\delta(q, a) &= \delta_1(q, a) && \text{kun } q \in F_1 \text{ ja } a \neq \varepsilon \\
\delta(q, \varepsilon) &= \delta_1(q, \varepsilon) \cup \{ q_2 \} && \text{kun } q \in F_1 \\
\delta(q, a) &= \delta_2(q, a) && \text{kun } q \in Q_2.
\end{aligned}$$

Selvästi $L(N) = A \circ B$. □

Vielä viimeiseksi tähtioperaatio:

Lause 1.8: [Sipser Thm 1.49] Jos A on säännöllinen, niin A^* on myös.

Todistus: Oletetaan taas $A = L(N_1)$, ja muodostetaan N jolla $L(N) = A^*$. Periaate:



(Huomaa, että alkutilan vaihtaminen hyväksyväksi ei välttämättä toimisi.)

Olkoon siis $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$. Nyt $N = (Q, \Sigma, \delta, q_0, F)$, missä

1. $q_0 \notin Q_1$ on uusi tila,
2. $Q = Q_1 \cup \{ q_0 \}$,
3. $F = F_1 \cup \{ q_0 \}$ ja
4. δ määritellään

$$\begin{aligned}
\delta(q_0, \varepsilon) &= \{ q_1 \} \\
\delta(q_0, a) &= \emptyset \qquad \text{kun } a \neq \varepsilon \\
\delta(q, a) &= \delta_1(q, a) \qquad \text{kun } q \in Q_1 - F_1 \\
\delta(q, a) &= \delta_1(q, a) \qquad \text{kun } q \in F_1 \text{ ja } a \neq \varepsilon \\
\delta(q, \varepsilon) &= \delta_1(q, \varepsilon) \cup \{ q_1 \} \qquad \text{kun } q \in F_1.
\end{aligned}$$

Selvästi $L(N) = A^*$. □

Säännölliset lausekkeet [Sipser luku 1.3]

Kiinnitetään jokin aakkosto Σ. *Säännöllinen lauseke* on kaava, jonka "arvona" on jokin aakkoston Σ kieli. Kun R on säännöllinen lauseke, sen "arvoa" eli sen *esittämää kieltä* merkitään $L(R)$. Perusajatuksena on muodostaa uusia kieliä edellä esitettyjen operaatioiden $\cup$, $\circ$ ja * avulla.

Esimerkki Säännöllisen lausekkeen $\mathsf{a}(\mathsf{b} \cup \mathsf{c})^*$ esittämään kieleen kuuluvat ne merkkijonot, joiden alussa on a ja sen jälkeen nolla tai useampia b- ja c-merkkejä. Siis

$$\begin{aligned}
L(\mathsf{a}(\mathsf{b} \cup \mathsf{c})^*) &= \{ \mathsf{a} \} \circ (\{ \mathsf{b}, \mathsf{c} \})^* \\
&= \{ \mathsf{a}, \mathsf{ab}, \mathsf{ac}, \mathsf{abb}, \mathsf{abc}, \mathsf{acb}, \mathsf{acc}, \mathsf{abbb}, \ldots \}.
\end{aligned}$$

Osoitamme jatkossa, että tälla tavoin voidaan *esittää* (tai kuvata) tasan ne kielet, jotka voidaan *tunnistaa* äärellisellä automaatilla, eli säännölliset kielet (mistä nimi).

Määrittelemme säännölliset lausekkeet seuraavasti:

1. $\emptyset$ ja ε ovat säännöllisiä lausekkeita, ja $L(\emptyset) = \emptyset$ ja $L(\varepsilon) = \{ \varepsilon \}$.
2. a on säännöllinen lauseke kaikilla $a \in \Sigma$, ja $L(a) = \{ a \}$.
3. Jos R_1 ja R_2 ovat säännöllisiä lausekkeita, niin myös $(R_1 \cup R_2)$ ja $(R_1 \circ R_2)$ ovat, ja $L((R_1 \cup R_2)) = L(R_1) \cup L(R_2)$ ja $L((R_1 \circ R_2)) = L(R_1) \circ L(R_2)$.
4. Jos R on säännöllinen lauseke, niin myös (R^*) on, ja $L((R^*)) = (L(R))^*$; ja
5. ei ole muita säännöllisiä lausekkeita, kuin mitä edelläolevasta seuraa.

Siis säännöllinen lauseke muodostetaan lähtemällä liikkeelle peruslausekkeista $\emptyset$, ε ja a, $a \in \Sigma$, ja soveltamalla yhdistettä, konkatenaatiota ja tähteä äärellinen määrä kertoja.

Yksinkertaistamme säännöllisten lausekkeiden kirjoitusasua seuraavilla konventioilla:

- Konkatenaatio jätetään merkitsemättä; esim. $\mathsf{a} \circ \mathsf{b} \circ \mathsf{a}$ kirjoitetaan aba.
- Operaatiot evaluoidaan presedenssijärjestyksessä "tähti, konkatenaatio, yhdiste" ja turhat sulut jätetään pois. Siis $a \cup bc^*$ tarkoittaa $(a \cup (b \circ (c^*)))$.
- Jos $\Sigma = \{ a_1, \ldots, a_k \}$, niin lauseketta $a_1 \cup \ldots \cup a_k$ merkitään lyhyesti Σ.
- Lausekkeen R konkatenaatiota itsensä kanssa k kertaa

  $$\underbrace{R \circ \ldots \circ R}_{k \text{ kertaa}}$$

  merkitään R^k.
- Lauseke RR^* lyhennetään R^+; siis $R^* = \varepsilon \cup R^+$.