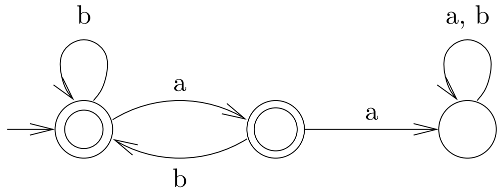
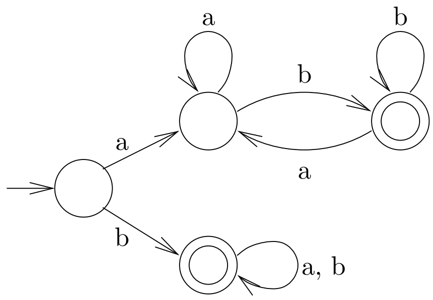
582206 Laskennan mallit
Erilliskoe 14.09.2010. Ratkaisuja

1. [*4+4 pistettä*] Anna seuraaville aakkoston $\{a, b\}$ kielille sekä säännöllinen lauseke että (deterministinen tai epädeterministinen) äärellinen automaatti.

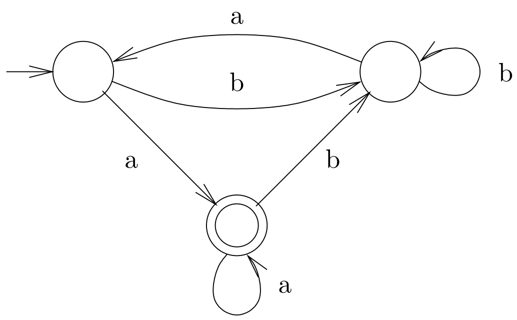
   (a) Kieli A koostuu kaikista merkkijonoista, joissa ei ole kahta a:ta peräkkäin.
   Vastaus: Säännöllinen lauseke: $b^* \cup b^*a(bb^*a)^*b^*$
   Deterministinen äärellinen automaatti:



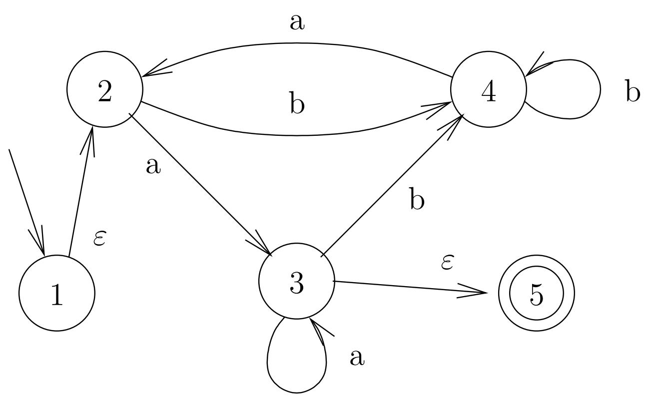
   (b) Kieli B koostuu kaikista merkkijonoista, jotka alkavat b:llä tai loppuvat b:hen.
   Vastaus: Säännöllinen lauseke: $b(a \cup b)^* \cup (a \cup b)^*b$
   Deterministinen äärellinen automaatti:



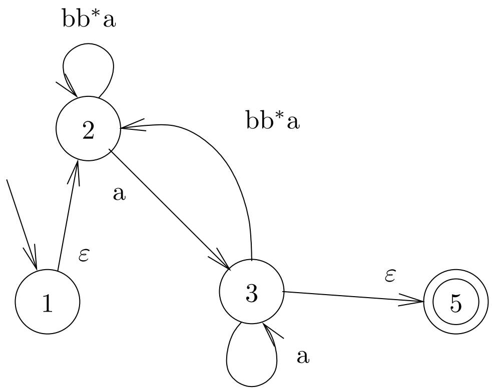
2. [*12 pistettä*]
   Muodosta oheisen äärellisen automaatin tunnistamalle kielelle säännöllinen lauseke kurssilla esitetyllä menetelmällä. Anna kielelle lyhyt sanallinen kuvaus.



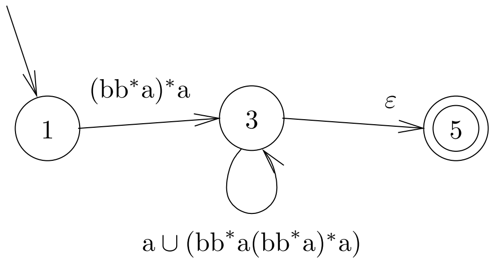
   Vastaus: Lisätään erillinen alku- ja lopputila ja numeroidaan tilat esityksen selkeyttämiseksi:



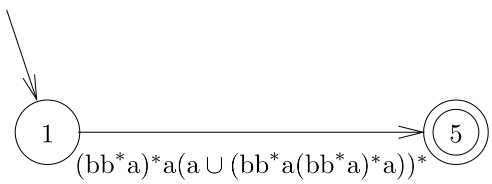
Eliminoidaan tila 4:



Eliminoidaan tila 2:



Eliminoidaan tila 3:



Saadaan säännöllinen lauseke

$$(\mathrm{bb}^*\mathrm{a})^*\mathrm{a}(\mathrm{a} \cup (\mathrm{bb}^*\mathrm{a}(\mathrm{bb}^*\mathrm{a})^*\mathrm{a}))^*.$$

Alkuperäisestä automaatista nähdään, että se hyväksyy merkkijonon a ja lisäksi sellaiset merkkijonot, jotka loppuvat aa. Hieman miettimällä nähdään, että säännöllinen lauseke antaa saman kielen.

3. [*12 pistettä*] Kun $a = a_1 \ldots a_n$ ja $b = b_1 \ldots b_n$ ovat kaksi aakkoston Σ yhtä pitkää merkkijonoa, missä $a_i \in \Sigma$ ja $b_i \in \Sigma$ kaikilla i, määritellään aakkoston $\Sigma \times \Sigma$ merkkijono $\textsc{Parit}(a, b) = (a_1, b_1) \ldots (a_n, b_n)$. Todista, että jos A ja B ovat säännöllisiä aakkoston Σ kieliä, niin kieli

$$\{\, \textsc{Parit}(a, b) \mid a \in A, b \in B, |a| = |b| \,\}$$

on säännöllinen aakkoston $\Sigma \times \Sigma$ kieli.

Vastaus: Olkoot $M_A = (Q_A, \Sigma, \delta_A, q_A, F_A)$ ja $M_B = (Q_B, \Sigma, \delta_B, q_B, F_B)$ deterministisiä äärellisiä automaatteja, jotka hyväksyvät kielet A ja B. Muodostetaan deterministinen äärellinen automaatti $M = (Q, \Sigma \times \Sigma, \delta, q_0, F)$ seuraavasti:

- $Q = Q_A \times Q_B$
- $q_0 = (q_A, q_B)$
- $F = F_A \times F_B$

- $\delta((r,s),(a,b)) = (\delta_A(r,a), \delta_B(s,b))$ kaikilla $(r,s) \in Q$ ja $(a,b) \in \Sigma \times \Sigma$.

Tarkastellaan automaatin M laskentaa syötteellä $w = (a_1, b_1) \ldots (a_n, b_n) = \textsc{Parit}(a,b)$, missä $a = a_1 \ldots a_n$ ja $b = b_1 \ldots b_n$. Merkitään laskennan aikana vierailtavia tiloja $(r_0, s_0), \ldots, (r_n, s_n)$. Siis automaatissa M suoritetaan laskenta

$$(r_0, s_0) \overset{(a_1,b_1)}{\to} (r_1, s_1) \overset{(a_2,b_2)}{\to} \ldots \overset{(a_n,b_n)}{\to} (r_n, s_n).$$

Automaatin M määritelmästä nähdään, että tällöin automaatissa M_A on laskenta

$$r_0 \overset{a_1}{\to} r_1 \overset{a_2}{\to} \ldots \overset{a_n}{\to} r_n$$

ja automaatissa M_B laskenta

$$s_0 \overset{b_1}{\to} s_1 \overset{b_2}{\to} \ldots \overset{b_n}{\to} s_n.$$

Siis $r_n \in F_A$, jos ja vain jos $a \in A$, ja $s_n \in F_B$, jos ja vain jos $b \in B$. Tästä seuraa, että $(r_n, s_n) \in F$, jos ja vain jos $w = \textsc{Parit}(a,b)$, missä $a \in A$ ja $b \in B$. Siis automaatti M tunnistaa kielen C. $\square$

4. [*2+4 pistettä*] Olkoon G seuraavaa yhteydetön kielioppi:

$$\begin{array}{ll} S \to RST \mid U & T \to \mathrm{c} \\ R \to \mathrm{a} \mid \varepsilon & U \to U\mathrm{b} \mid \varepsilon. \end{array}$$

   (a) Anna lyhyt sanallinen kuvaus kieliopin G tuottamalle kielelle.

   (b) Osoita, että G on moniselitteinen.

Kielioppi tuottaa kielen, jonka merkkijonoissa on a-, b- ja c-merkkejä aakkosjärjestyksessä siten, että c-merkkejä on ainakin yhtä monta kuin a-merkkejä.

Kielioppi on moniselitteinen, sillä esim. merkkijonolla acc on kaksi vasenta johtoa:

$$\begin{array}{c} S \Rightarrow RST \Rightarrow ST \Rightarrow RSTT \Rightarrow \mathrm{a}STT \Rightarrow \mathrm{a}UTT \Rightarrow \mathrm{a}TT \Rightarrow \mathrm{ac}T \Rightarrow \mathrm{acc} \\ S \Rightarrow RST \Rightarrow \mathrm{a}ST \Rightarrow \mathrm{a}RSTT \Rightarrow \mathrm{a}STT \Rightarrow \mathrm{a}UTT \Rightarrow \mathrm{a}TT \Rightarrow \mathrm{ac}T \Rightarrow \mathrm{acc} \quad . \end{array}$$

5. [*10 pistettä*] Selvitä CYK-algoritmia soveltamalla, kuuluuko merkkijono abbaa kieliopin

$$\begin{array}{ll} S \to AX \mid YA & A \to AA \mid \mathrm{a} \\ X \to AB \mid AY & B \to BB \mid \mathrm{b} \\ Y \to BA \mid BX & \end{array}$$

tuotamaan kieleen.

Vastaus: CYK-algoritmi laskee seuraavan taulukon:

	1	2	3	4	5
1	A	X	X	X	X
2	—	B	B	Y	Y, S
3	—	—	B	Y	Y, S
4	—	—	—	A	A
5	—	—	—	—	A

Taulukon oikeassa ylänurkassa on ei ole lähtösymbolia S, joten merkkijono abbaa ei kuulu kieleen.

6. [*12 pistettä*] Ohjelmointikurssin yhtenä harjoitustehtävänä opiskelijoiden pitää laatia Java-ohjelma, joka tulostaa luvut $1, \ldots, 10$ ja pysähtyy.

   (a) Olisiko mahdollista laatia tehtävät automaattisesti tarkastava ohjelma, joka saa syötteenä kaikki opiskelijoiden palauttamat ratkaisut ja kertoo, mitkä niistä toimivat oikein ja mitkä eivät?

   (b) Olisiko mahdollista laatia tarkastamista helpottava apuohjelma, joka erottelee joukosta ne opiskelijoiden ohjelmat, jotka menevät ikuiseen silmukkaan eivätkä tulosta mitään?

   Perustele vastauksesi.

   Kummassakin kohdassa tarkastusohjelmalta vaaditaan, että se ei koskaan tee virhettä, vaikka opiskelija tuntisi sen lähdekoodin ja yrittäisi tahallaan harhauttaa sitä kumpaan tahansa suuntaan.

   Vastaus:

   Kumpaakaan kuvattua ohjelmaa ei ole mahdollista laatia.

   Tarkastellaan ensin (b)-kohtaa. Haluttu apuohjelma ratkaisisi oleellisesti Java-ohjelmien pysähtymisongelman. Koska Java-ohjelmat ovat yhtä ilmaisuvoimaisia kuin Turingin koneet ja Turingin koneiden pysähtymisongelma tiedetään ratkeamattomasti, tämä on mahdotonta.

   Esitetään (b)-kohdan perustelu hieman yksityiskohtaisemmin. Tehdään vastaoletus, että haluttu apuohjelma olisi olemassa ja koodattu esim. Javalla. Osoitamme alla, että sen avulla voidaan ratkaista mielivaltaisesta Java-ohjelmasta P, pysähtyykö se syötteellä x. Siis osaisimme ratkaista Javalla Java-kielisten ohjelmien pysähtymisongelman. Koska Java on tasan yhtä ilmaisuvoimainen kuin Turingin koneet, tämä on sama, kuin että Turingin koneiden pysähtymisongelma voitaisiin ratkaista Turingin koneella. Tämä kuitenkin tiedetään mahdottomaksi. Siis oletuksen apuohjelman olemassaolosta on oltava väärä.

   Olkoon siis annettu ohjelma P ja syöte w. Muodostetaan ohjelma P', joka toimii muuten kuten P toimisi syötteellä w, mutta ei tulosta mitään. Tämä muunnos on selvästi mahdollista toteuttaa algoritmisesti. Nyt

   - jos P pysähtyy syötteellä x, niin P' pysähtyy tulostamatta mitään
   - jos P ei pysähdy syötteellä x, niin P' ei pysähdy.

   Siis kysymys "pysähtyykö P syötteellä x" voitaisiin ratkaista muodostamalla ensin P' ja soveltamalla tehtävänannon mukaista apuohjelmaa, kuten edellä väitettiin.

   Tarkastellaan nyt (a)-kohtaa. Perusajatus on sama kuin edellä: on mahdotonta tietää, milloin ohjelma ei lainkaan pysähdy. Tehdään taas yksityiskohtaisempi tarkastelu.

   Tehdään vastaoletus, että automaattitarkastaja olisi olemassa. Ratkaisemme Java kielen pysähtymisongelman seuraavasti:

   (a) Olkoon annettu Java-ohjelma P ja syöte x.

   (b) Muodostetaan Java-ohjelma P', joka toimii seuraavasti:

      i. Simuloi ohjelmaa P syötteellä x, mutta älä tulosta mitään.

      ii. Jos simuloitava suoritus pysähtyy, tulosta luvut 1,. . . , 10 ja pysähdy.

   (c) Nyt P pysähtyy syötteellä x, jos ja vain jos P' kelpaa automaattitarkastajalle.