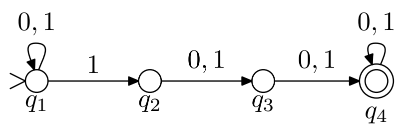
**T-79.148** **Kevät 2002**
**Tietojenkäsittelyteorian perusteet**
**Harjoitus 4**
**Demonstraatiotehtävien ratkaisut**

4. **Tehtävä**: Laadi epädeterministinen äärellinen automaatti, joka testaa onko annetun binäärijonon kolmanneksi viimeinen merkki 1, ja determinisoi se.

   **Vastaus**: Kielen $L = \{w \in \{0,1\}^* \mid w\text{:n kolmanneksi viimeinen merkki on 1 }\}$ tunnistaa epädeterministinen automaatti $M = (Q, \Sigma, \delta, q_1, F)$, missä

$$Q = \{q_1, q_2, q_3, q_4\}$$
$$\Sigma = \{a, b\}$$
$$F = \{q_4\},$$

ja siirtymäfunktio $\delta$ on määritelty kuten allaolevassa kuvassa:

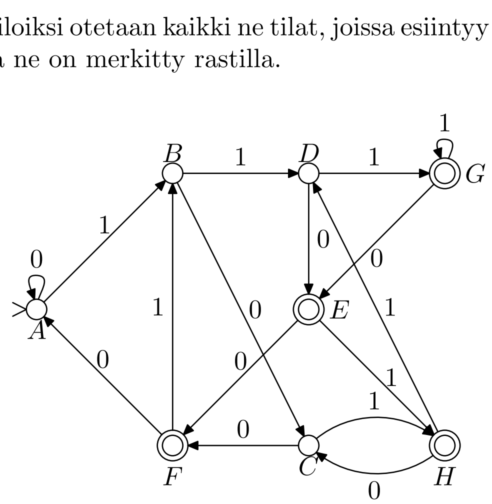



Konetta $M$ vastaava deterministinen automaatti $M'$ muodostetaan siten, että $M'$:n tiloiksi otetaan kaikki $Q$:n osajoukot ($Q' = \mathcal{P}(Q)$). Tilajoukkoihin koodataan kaikki mahdolliset $M$:n laskennat. Esimerkiksi kun $M$ on lukenut syötteen 010, voi se olla joko tilassa $q_1$ tai $q_3$. Niinpä koneen $M'$ täytyy samalla syötteellä päätyä tilaan $\{q_1, q_3\}$.

Muodostetaan tilansiirtofunktio $\delta'$:

| $q$ | 0 | 1 | uusi nimi |
|---|---|---|---|
| $\{q_1\}$ | $\{q_1\}$ | $\{q_1, q_2\}$ | $A$ |
| $\{q_1, q_2\}$ | $\{q_1, q_3\}$ | $\{q_1, q_2, q_3\}$ | $B$ |
| $\{q_1, q_3\}$ | $\{q_1, q_4\}$ | $\{q_1, q_2, q_4\}$ | $C$ |
| $\{q_1, q_2, q_3\}$ | $\{q_1, q_3, q_4\}$ | $\{q_1, q_2, q_3, q_4\}$ | $D$ |
| $\{q_1, q_3, q_4\}$ | $\{q_1, q_4\}$ | $\{q_1, q_2, q_4\}$ | $E\times$ |
| $\{q_1, q_4\}$ | $\{q_1\}$ | $\{q_1, q_2\}$ | $F\times$ |
| $\{q_1, q_2, q_3, q_4\}$ | $\{q_1, q_3, q_4\}$ | $\{q_1, q_2, q_3, q_4\}$ | $G\times$ |
| $\{q_1, q_2, q_4\}$ | $\{q_1, q_3\}$ | $\{q_1, q_2, q_3\}$ | $H\times$ |

Automaatin $M'$ lopputiloiksi otetaan kaikki ne tilat, joissa esiintyy jokin $M$:n lopputiloista. Ylläolevassa taulukossa ne on merkitty rastilla.



5. **Tehtävä**: Osoita, että jos aakkoston $\Sigma = \{a, b\}$ kielet $A$ ja $B$ voidaan tunnistaa äärellisillä automaateilla, niin samoin voidaan tunnistaa myös kielet $\bar{A} = \Sigma^* - A$, $A \cup B$ ja $A \cap B$.

   **Vastaus**: Olkoon $A$ ja $B$ aakkoston $\Sigma = \{a, b\}$ kieliä, jotka voidaan tunnistaa äärellisillä automaateilla. Halutaan osoittaa, että myös kielet $\bar{A} = \Sigma^* - A$, $A \cup B$ ja $A \cap B$ voidaan tunnistaa äärellisillä automaateilla.

$\bar{A}$: Olkoon $M_A = (Q, \Sigma, \delta, q_0, F)$ deterministinen tilakone[1], joka tunnistaa kielen $A$. Muodostetaan tästä kielen komplementin tunnistava automaatti $M_{\bar{A}}$:

$$M_{\bar{A}} = (Q, \Sigma, \delta, q_0, Q - F) \ .$$

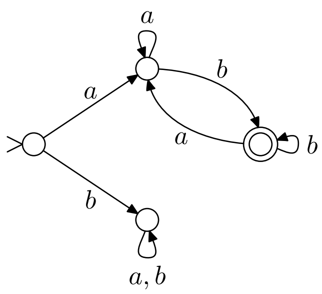
Kone $M_{\bar{A}}$ toimii muuten täsmälleen samalla tapaa kuin $M_A$, mutta hyväksyvät tilat on muutettu hylkääviksi ja päinvastoin. Näin ollen $M_{\bar{A}}$ hyväksyy ne sanat, jotka $M_A$ hylkää ja hylkää ne, jotka $M_A$ hyväksyy, joten $L(M_{\bar{A}}) = \bar{A}$.

Tarkastellaan esimerkiksi automaattia, joka tunnistaa kielen:

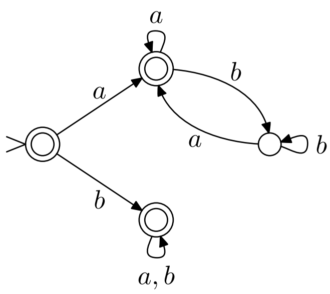
$$A = \{w \in \Sigma^* \mid \ w \text{ on muotoa } axb, \text{ missä } x \in \Sigma^*\} \ .$$

Kieleen $A$ kuuluvat kaikki sanat, jotka alkavat $a$-kirjaimella ja päättyvät $b$-kirjaimeen. Alla esitetään automaatit $M_A$ ja $M_{\bar{A}}$:

$M_A$:




$M_{\bar{A}}$:




Tässä on huomattava, että esitetty konstruktio toimii vain, jos $M_A$ on deterministinen. (Yritä etsiä yksinkertainen vastaesimerkki epädeterministiselle tapaukselle.)

$A \cup B$: Olkoot $M_A = (Q_A, \Sigma, \delta_A, s_A, F_A)$ ja $M_B = (Q_B, \Sigma, \delta_B, s_B, F_B)$ äärelliset automaatit, jotka tunnistavat kielet $A$ ja $B$. Oletetaan lisäksi, että tilajoukot ovat erilliset, eli $Q_A \cap Q_B = \emptyset$. Tämä oletus voidaan tehdä, sillä tarvittaessa voidaan toisen koneen tilat nimetä uudelleen. Muodostetaan epädeterministinen tilakone $M_{A\cup B}$ seuraavasti:

$$M_{A\cup B} = (Q, \Sigma, \delta, s, F) \ ,$$

missä

$$\begin{aligned} Q &= Q_A \cup Q_B \cup \{s\} \\ F &= F_A \cup F_B \\ \delta &= \delta_A \cup \delta_B \cup \{(s, \varepsilon, s_a), (s, \varepsilon, s_b)\} \ . \end{aligned}$$

Kone $M_{A\cap B}$ muodostetaan siis yhdistämällä koneet $M_A$ ja $M_B$. Tila $s$ on uusi alkutila, josta voidaan siirtyä tyhjällä siirtymällä joko $M_A$:n tai $M_B$:n alkutilaan.

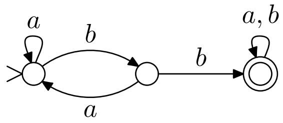
Jos sana $x$ kuuluu kieleen $A$, $M_{A\cup B}$ hyväksyy sen siirtymällä aluksi tilaan $s_A$ ja suorittamalla sen jälkeen samat siirrot kuin kone $M_A$ olisi suorittanut. Mikäli $x \in B$, siirrytään tilaan $s_B$ ja toimitaan kuten $M_B$.

Tarkastellaan edellisessä kohdassa esiteltyä automaattia $M_A$ sekä uutta automaattia $M_B$, joka tunnistaa kielen:

$$B = \{w \in \Sigma^* \mid w : \text{ssä esiintyy osajono } bb\} \ .$$

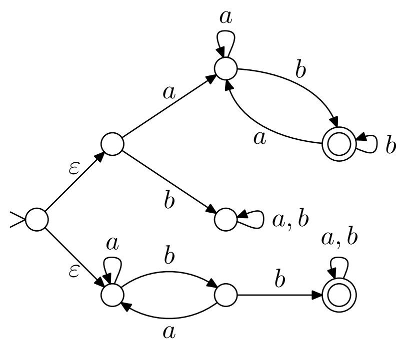
[1] $M_A$ on välttämättä olemassa, sillä mitä tahansa epädeterminististä automaattia kohden voidaan muodostaa saman kielen tunnistava deterministinen automaatti.

$M_B$:




Kielen $A \cup B$ hyväksyvä automaatti on seuraavanlainen:

$M_{A \cup B}$:



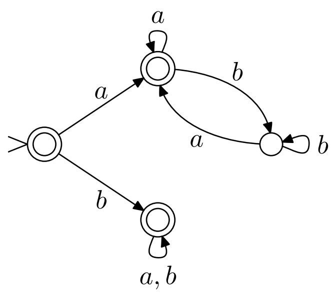

Usein lisätään $M_{A \cup B}$:hen myös uusi lopputila $f$, ja lisätään sinne tyhjä siirtymä kaikista alkuperäisistä lopputiloista $q \in F_A \cup F_B$. Tällöin $F = \{f\}$.

$A \cap B$: Väite seuraa suoraan kahdesta edellisestä kohdasta, sillä DeMorganin sääntöjen perusteella:

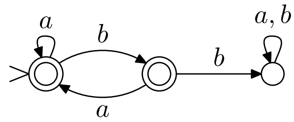
$$A \cap B = \overline{\overline{A} \cup \overline{B}} \ .$$

Tarkastellaan vielä yllä esiteltyjä koneita $M_A$ ja $M_B$, ja muodostetaan kone $M_{A \cap B}$ käyttäen DeMorganin sääntöä:

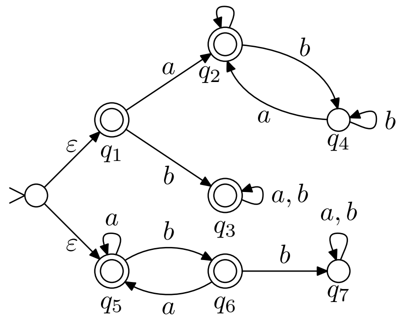
$M_{\bar{A}}$:




$M_{\bar{B}}$:

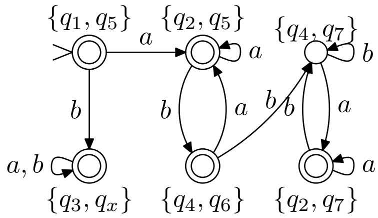



$M_{\bar{A} \cup \bar{B}}$:



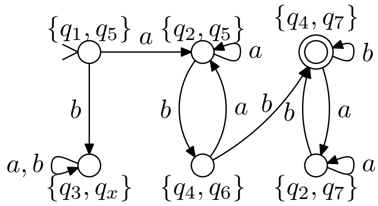

Koneen $M_{\bar{A} \cup \bar{B}}$ komplementointia varten se täytyy ensin determinisoida (kone on jo minimoitu, yksityiskohdat liitteenä):

$M'_{\bar{A}\cup\bar{B}}$:




Nyt saadaan haluttu kone vaihtamalla ylläolevan koneen hyväksyvät tilat hylkääviksi ja päinvastoin:

$M_{A\cap B}$:




Kahden automaatin leikkaus voidaan määritellä myös suoraan käyttäen samantapaista menetelmää kuin seuraavassa tehtävässä.

6. **Tehtävä**: (*soveltava*) Monet tiedonsiirtoprotokollien analysointiin käytettävät menetelmät muodostavat järjestelmän tila-avaruuden, jota tutkimalla etsitään ongelmia, esimerkiksi lukkiumia. Yksi tapa muodostaa tila-avaruus on mallintaa kutakin protokollan osapuolta erikseen tilakoneella ja yhdistää nämä yhdeksi isoksi tilakoneeksi.

   Olkoon $M_1 = (K_1, \Sigma_1, \Delta_1, s_1, \emptyset)$ ja $M_2 = (K_2, \Sigma_2, \Delta_2, s_2, \emptyset)$ epädeterministisiä tilakoneita. Yhdistetty tilakone $M = (K, \Sigma, \Delta, s, \emptyset)$ muodostetaan seuraavasti:

   – $K = K_1 \times K_2$
   – $\Sigma = \Sigma_1 \cup \Sigma_2$
   – $s = (s_1, s_2)$
   – Siirtymä $(p_1, p_2) \xrightarrow{a} (q_1, q_2)$ kuuluu relaatioon $\Delta$ mikäli jokin seuraavista ehdoista toteutuu:
     1. $a \in \Sigma_1 \cap \Sigma_2$, $(p_1, a, q_1) \in \Delta_1$ ja $(p_2, a, q_2) \in \Delta_2$.
     2. $a \in \Sigma_1$, $a \notin \Sigma_2$, $(p_1, a, q_1) \in \Delta_1$ ja $p_2 = q_2$.
     3. $a \notin \Sigma_1$, $a \in \Sigma_2$, $(p_2, a, q_2) \in \Delta_2$ ja $p_1 = q_1$.

     Olkoot $M_1$ ja $M_2$ kuten alla. Muodosta yhdistetty tilakone $M$ ja osoita, että järjestelmässä on lukkiuma (eli tila, josta ei lähde yhtään siirtymää).

**Vastaus**: Yhdistetyn tilakoneen tiloiksi otetaan kaikki mahdolliset parit, jotka voidaan muodostaa alkuperäisten tilakoneiden tiloista. Esim, mikäli $K_1 = \{q_1, q_2, q_3\}$ ja $K_2 = \{p_1, p_2\}$, niin yhdistetyn koneen tilat ovat:

$$K = \{(q_1, p_1), (q_2, p_1), (q_3, p_1), (q_1, p_2), (q_2, p_2), (q_3, p_2)\} \ .$$

Konseptuaalisesti koneiden siirtymät jaetaan kahteen luokkaan, sisäisiin ja ulkoisiin. Siirtymä on sisäinen, mikäli siihen liittyvä symboli ei kuulu toisen tilakoneen aakkostoon.

Tehtävänannossa ainoa $M_1$:n sisäinen siirtymä on $q_3 \xrightarrow{int_1} q_0$. Tilakoneella $M_2$ on neljä sisäistä siirtymää, ne jotka tehdään symbolilla $int_2$. Mikäli siirtymän symboli esiintyy molemmissa tilakoneissa, on siirtymä ulkoinen.

Yhdistetyn tilakoneen siirtymärelaation muodostamissäännöt tarkoittavat käytännössä sitä, että tilakoneet synkronoivat toimintansa ulkoisten siirtymien kautta. Tilakone voi tehdä sisäisiä siirtymiä milloin vain, mutta ulkoinen siirtymä voidaan tehdä vain, kun molemmat koneet tekevät saman siirtymän samaan aikaan.

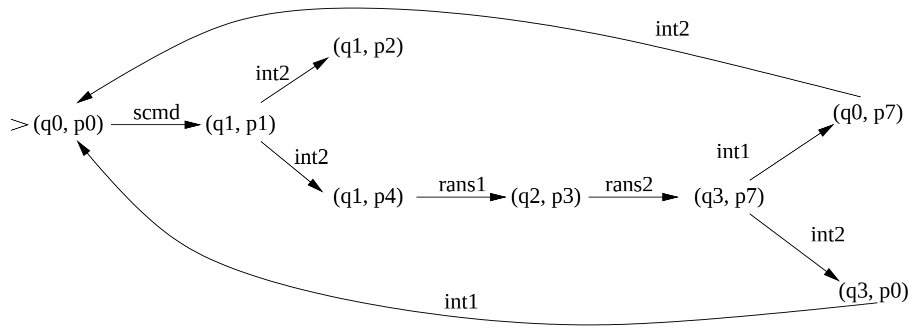
Tehtävän tilakoneiden tilajoukot ovat

$$K_1 = \{q_0, q_1, q_2, q_3\}$$
$$K_2 = \{p_0, p_1, p_2, p_3, p_4, p_5, p_6, p_7\} \ ,$$

joten yhdistetyssä tilakoneessa on $4 \times 8 = 32$ tilaa. Huomattavan monia näistä tiloista ei kuitenkaan voida koskaan saavuttaa, joten tila-avaruus supistuu muotoon:

$$K = \{(q_0, p_0), (q_1, p_1), (q_1, p_2), (q_1, p_4), (q_2, p_3), (q_3, p_7), (q_0, p_7), (q_3, p_0)\}$$

Tilakoneen siirtymät on esitetty alla olevassa kuvassa.




Kuvasta huomataan, että tilasta $(q_1, p_2)$ ei johda yhtään kaarta eteenpäin. Kyseiseen tilaan on mahdolista päästä alkutilasta suorituksella

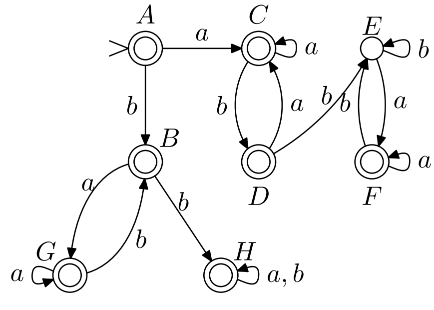
$$(q_0, p_0) \stackrel{scmd}{\rightarrow} (q_1, p_1) \stackrel{int_2}{\rightarrow} (q_1, p_2) \ ,$$

joten systemissä on saavutettavissa oleva lukkiuma.

Tehtävässä ei ole määritelty lopputiloja tilakoneille, koska tiedonsiirtoprotokollia analysoitaessa ei yleensä olla kiinnostuneista yksittäisistä suorituksita vaan käsitellään äärettömän mittaisia viestijonoja. Tavalliset tilakoneet sallivat ainoastaan äärellisen (tosin mielivaltaisen pitkän) syötteen, ja äärettömiä sanoja tunnistamaan käytetäänkin vahvempia automaattiluokkia, yleensä Büchi-automaatteja. Nämä automaatit eivät kuitenkaan kuulu tämän kurssin asioihin, ja niihin pääsee tutustumaan kurssilla T-79.179 Rinnakkaiset ja hajautetut järjestelmät.

### Liite: tilakoneen minimointi

Tilakoneen determinisointialgoritmilla saadaan tehtävän 5. tilakone $M_{\bar{A} \cup \bar{A}}$ muutettua seuraavaan muotoon:



Nyt halutaan löytää pienin deterministinen tilakone, joka tunnistaa saman kielen. Luennolla esitetty algoritmi toimii siten, että automaatin tilojen välille määritellään ekvivalenssirelaatio, jota vaiheittain tarkennetaan, kunnes haluttu lopputulos saavutetaan.

Algoritmin ensimmäisessä vaiheessa poistetaan kaikki tilat, joita ei voida saavuttaa alkutilassa. Tässä automaatissa sellaisia ei ole, joten automaatti pysyy vielä ennallaan.

Seuraavaksi muodostetaan ensimmäinen ekvivalenssiositus siten, että automaatin lopputiloista tehdään yksi luokka ja kaikista muista tiloista toinen:

0-ekvivalenssi:

| Luokka | Tila | $a$ | $b$ |
|---|---|---|---|
| I | $A$ | $C$ (I) | $B$ (I) |
| | $B$ | $G$ (I) | $H$ (I) |
| | $C$ | $C$ (I) | $D$ (I) |
| | $D$ | $C$ (I) | $E$ (II) |
| | $F$ | $F$ (I) | $E$ (II) |
| | $G$ | $G$ (I) | $B$ (I) |
| | $H$ | $H$ (I) | $H$ (I) |
| II | $E$ | $F$ (I) | $E$ (II) |

Kaaviosta huomataan, että I-luokan tiloista $D$ ja $F$ siirrytään $b$:llä II-luokan tilaan $E$, kun taas kaikista muista tiloista $b$-siirtymä vie johonkin I-luokkaan kuuluvaan tilaan. Erotetaan nyt kaksi erilaista tilaa omaksi luokakseen:

1-ekvivalenssi:

| Luokka | Tila | $a$ | $b$ |
|---|---|---|---|
| I | $A$ | $C$ (I) | $B$ (I) |
| | $B$ | $G$ (I) | $H$ (I) |
| | $C$ | $C$ (I) | $D$ (III) |
| | $G$ | $G$ (I) | $B$ (I) |
| | $H$ | $H$ (I) | $H$ (I) |
| II | $E$ | $F$ (III) | $E$ (II) |
| III | $D$ | $C$ (I) | $E$ (II) |
| | $F$ | $F$ (III) | $E$ (II) |

Tällä kertaa tilat $C$ ja $F$ eivät sovi luokkiinsa, ja ne täytyy erottaa omiksi luokikseen. Näin jatketaan, kunnes lopulta kaikki luokat ovat konsistentteja:

2-ekvivalenssi:

| Luokka | Tila | $a$ | $b$ |
|---|---|---|---|
| I | $A$ | $C$ (IV) | $B$ (I) |
| | $B$ | $G$ (I) | $H$ (I) |
| | $G$ | $G$ (I) | $B$ (I) |
| | $H$ | $H$ (I) | $H$ (I) |
| II | $E$ | $F$ (V) | $E$ (II) |
| III | $D$ | $C$ (IV) | $E$ (II) |
| IV | $C$ | $C$ (IV) | $D$ (III) |
| V | $F$ | $F$ (V) | $E$ (II) |

3-ekvivalenssi:

| Luokka | Tila | $a$ | $b$ |
|---|---|---|---|
| I | $A$ | $C$ (IV) | $B$ (VI) |
| II | $E$ | $F$ (V) | $E$ (II) |
| III | $D$ | $C$ (IV) | $E$ (II) |
| IV | $C$ | $C$ (IV) | $D$ (III) |
| V | $F$ | $F$ (V) | $E$ (II) |
| VI | $B$ | $G$ (VI) | $H$ (VI) |
| | $G$ | $G$ (VI) | $B$ (VI) |
| | $H$ | $H$ (VI) | $H$ (VI) |

Kaikki luokat ovat nyt konsistentit, ja voidaan muodostaa tilakone, jonka tiloina ovat syntyneet ekvivalenssiluokat. Minimoitu kone on esitetty kaaviona tehtävän 5. vastauksen yhteydessä.

Terminä $k$-ekvivalenssi tarkoittaa sitä, että kaikki samaan luokkaan kuuluvat tilat käsittelevät samalla tapaa kaikkia korkeintaan $k$ merkkiä pitkiä syötteitä. Jos $p \overset{k}{\equiv} q$ ja tilasta $p$ lähtevä $k$:n pituinen laskenta päätyy lopputilaan, niin myös $q$:sta lähtevä samalla syötteellä tehty laskenta päätyy hyväksyvään tilaan, ja päinvastoin.