

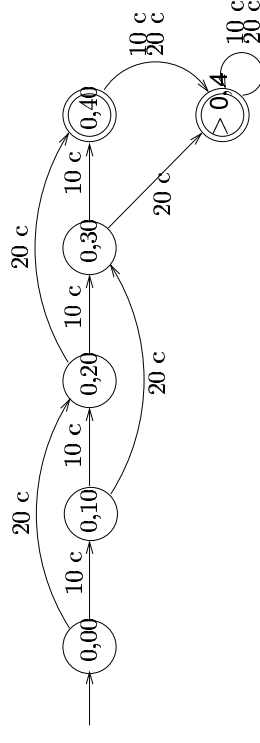
ÄÄRELLISET AUTOMAATIT JA SÄÄNNÖLLISET KIELET

2.1 Tilakaaviot ja tilataulut

Tarkastellaan aluksi tietojenkäsittelyjärjestelmiä, joilla on vain äärellisen monta mahdollista tilaa. Tällaisen järjestelmän toiminta voidaan kuvata *äärellisenä automaattina* t. *äärellisenä tilakoneena* (engl. finite automaton, finite state machine).

Äärellisillä automaateilla on useita vaihtoehtoisia esitystapoja: tilakaaviot, tilataulut, ...

Esimerkki 1: Kahviautomaatti.



Em. tilakaavion esittämä automaatti ratkaisee päätösongelman “riittävätkö annetut rahat kahvin ostamiseen?”

Äärellisiä automaatteja voidaan yleensäkin käyttää yksinkertaisten päätösongelmien ratkaisujen mallintamiseen. Automaattimallista on muitakin kuin binääriavasteisten järjestelmien kuvaamiseen tarkoitettuja versioita (ns. Moore- ja Mealy-automaatit), mutta niitä ei käsitellä tällä kurssilla.

Tilakaavioiden merkinnät:

Automaatin tila nimeltä q



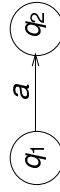
Alkutila



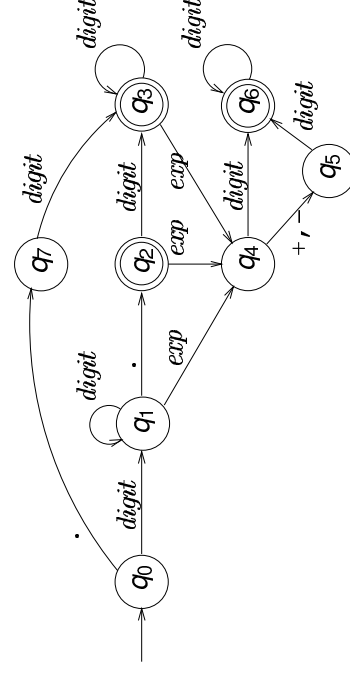
Lopputila: automaatti “hyväksyy” syötejonon, joss se jonon loppuessa on tällaisessa tilassa



Syötemerkkin a aikaansaama siirtymä tilasta q_1 tilaan q_2



Esimerkki 2: C-kielen etumerkittömät reaali-luvut.



Käytetyt lyhenteet: $digit = \{0, 1, \dots, 9\}$, $exp = \{E, e\}$.

Äärellisen automaatin esitys *tilatauluna*: automaatin uusi tila vanhan tilan ja syötemerkin funktiona.

Esim. reaalityttöautomaatin tilataulu:

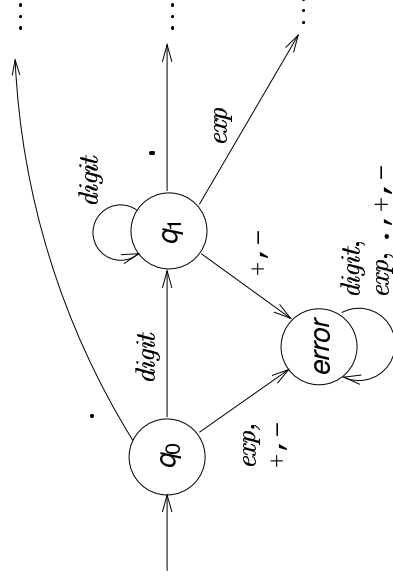
	digit	.	exp	+	-
→	q ₀	q ₁			
←	q ₁	q ₁	q ₄		
←	q ₂	q ₃	q ₄		
	q ₃	q ₃	q ₄		
	q ₄	q ₆		q ₅	q ₅
	q ₅	q ₆			
←	q ₆	q ₆			
	q ₇	q ₃			

K: Mitä tilataulun tyhjät paikat tarkoittavat?

V: Tilataulun tyhjät paikat, tai vastaavasti tilakaavion "puuttuvat" kaaret, kuvaavat automaatin virhetilanteita. Jos automaatti ohjautuu tällaiseen paikkaan, syötejono ei kuulu automaatin hyväksymään joukkoon.

Muodollisesti automaatissa ajatellaan olevan erityinen virhetila, jota ei vain selkeyden vuoksi merkittä näkyviin.

Esim. reaalityttöautomaatin täydellinen kaavioesitys olisi:



ja reaalityttöautomaatin täydellinen tauluesitys olisi:

	digit	.	exp	+	-
→	q ₀				
	q ₁	q ₇	error	error	error
	q ₁	q ₂	q ₄	error	error
	⋮	⋮	⋮	⋮	⋮
←	q ₆	error	error	error	error
	error	error	error	error	error

2.2 Äärellisiin automaatteihin perustuva ohjelmointi

Annetun äärellisen automaatin pohjalta on helppo laatia automaatin toimintaa vastaava ohjelma. Esim. reaallukuautomaattiin perustuva syötejonon syntaksitestaus:

```

...
case 99:
    break;
}
if (q == 2 || q == 3 || q == 6)
    printf("SYÖTE ON REAALILUKU.\n");
else
    printf("SYÖTE EI OLE REAALILUKU.\n");
}

```

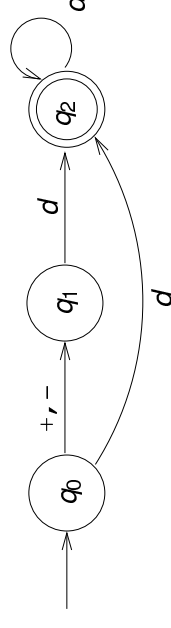
```

#include <stdio.h>
#include <ctype.h>
main() {
    int q, c;
    q = 0;
    while ((c = getchar()) != '\n')
        switch (q) {
            case 0:
                if (isdigit(c)) q = 1;
                else if (c == '.' || c == ',') q = 7;
                else q = 99;
                break;
            case 1:
                if (isdigit(c)) q = 1;
                else if (c == '.' || c == ',') q = 2;
                else if (c == 'E' || c == 'e') q = 4;
                else q = 99;
                break;

```

Semanttisten toimintojen liittäminen äärellisiin automaatteihin

Esimerkki. Kahdeksanjärjestelmän lukuja tunnistava automaatti ja siihen perustuva syöteluvun arvonmääritys ("muuttaminen kymmenjärjestelmään").



Lyhennysmerkintä $d = \{0, 1, \dots, 7\}$.

Pelkän syntaksitestin toteutus:

```
#include <stdio.h>
#include <ctype.h>

main()
{
    int q, c;
    q = 0;
    while ((c = getchar()) != '\n') {
        switch (q) {
            case 0:
                if (c == '+' || c == '-') q = 1;
                else if ('0' <= c && c <= '7') q = 2;
                else q = 99;
                break;

```

```
        case 1:
            if ('0' <= c && c <= '7') q = 2;
            else q = 99;
            break;
        case 2:
            if ('0' <= c && c <= '7') q = 2;
            else q = 99;
            break;
        case 99:
            break;
    }
}
if (q == 2)
    printf("SYÖTE OK.\n");
else
    printf("VIRHEELLINEN LUKU.\n");
}
```

Täydennys syötelluvun arvon laskevalla operaatioilla ("luvun muuttaminen kymmenjärjestelmään"):

```
#include <stdio.h>

int main(void) {
    int q, c;
    int sgn, val; /* SEM: sgn = etumerkki */
    sgn = 1; val = 0; /* SEM: val = itseisarvo */
    q = 0;

```

```
while ((c = getchar()) != '\n') {
    switch (q) {
        case 0:
            if (c == '+') q = 1;
            else if (c == '-') {
                sgn = -1;
                q = 1;
            }
            else if ('0' <= c && c <= '7') {
                val = c - '0';
                q = 2;
            }
            else q = 99;
            break;

```

```

case 1:
  if ('0' <= c && c <= '7') {
    val = c - '0';
    q = 2;
  }
  else q = 99;
  break;
case 2:
  if ('0' <= c && c <= '7') {
    val = 8 * val + (c - '0');
    q = 2;
  }
  else q = 99;
  break;
case 99:
  break;
}

```

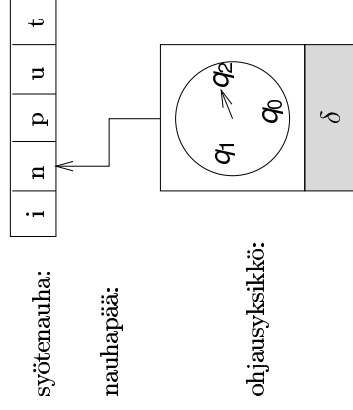
```

}
if (q == 2)
{ printf("LUVUN ARVO ON %d.\n", sgn*val); /* SEM*/
  exit(0); }
else
{ printf("VIRHEELLINEN SYÖTE.\n"); /* SEM */
  exit(1); }
}

```

2.3 Äärellisen automaatin käsitteen formalisointi

Mekanistinen malli:



Äärellinen automaatti M koostuu äärellisistä *ohjauksyksiköistä*, jonka toimintaa säätelee automaatin *siirtymäfunktio* δ , sekä merkipaikoihin jaetusta *syötenauhasta* ja nämä yhdistävästä *nauhapäädästä*, joka kullakin hetkellä osoittaa yhtä syötenauhan merkkiä.

Automaatin "toiminta":

Automaatti käynnistetään erityisessä *alkutilassa* q_0 , siten että tarkasteltava syöte on kirjoitettuna syötenauhalle ja nauhapää osoittaa sen ensimmäistä merkkiä.

Yhdessä toiminta-askelssa automaatti lukee nauhapään kohdalla olevan syötemerkin, päättää ohjauksikon tilan ja luetun merkin perusteella siirtymäfunktion mukaisesti ohjauksikon uudesta tilasta, ja siirtää nauhapäätä yhden merkin eteenpäin.

Automaatti pysähtyy, kun viimeinen syötemerkki on käsitelty. Jos ohjauksikon tila tällöin kuuluu erityiseen (*hyväksyvien lopputilojen* joukkoon, automaatti *hyväksyy* syötteen, muuten *hylkää* sen.

Automaatin *tunnistama kieli* on sen hyväksymien merkkijonojen joukko.

Täsmällinen muotoilu:

Äärellinen automaatti on viisikko

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

missä

- ▶ Q on automaatin tilojen äärellinen joukko;
- ▶ Σ on automaatin syöteakkosto;
- ▶ $\delta : Q \times \Sigma \rightarrow Q$ on automaatin siirtymäfunktio;
- ▶ $q_0 \in Q$ on automaatin alkutila;
- ▶ $F \subseteq Q$ on automaatin (hyväksyvien) lopputilojen joukko.

Esimerkki. Reaaliukuautomaatin formaali esitys:

$$M = (\{q_0, \dots, q_7, error\}, \{0, 1, \dots, 9, ., E, e, +, -, \delta, q_0, \{q_2, q_3, q_6\}\},$$

missä δ on kuten alemmin taulukossa; esim.

$$\begin{aligned} \delta(q_0, 0) &= \delta(q_0, 1) = \dots = \delta(q_0, 9) = q_1, \\ \delta(q_0, .) &= q_7, \quad \delta(q_0, E) = \delta(q_0, e) = error, \\ \delta(q_1, .) &= q_2, \quad \delta(q_1, E) = \delta(q_1, e) = q_4, \\ & \text{jne.} \end{aligned}$$

Tilanne (q, w) johtaa suoraan tilanteeseen (q', w') , merkitään

$$(q, w) \vdash_M (q', w'),$$

jos on $w = aw'$ ($a \in \Sigma$) ja $q' = \delta(q, a)$. Tällöin sanotaan myös, että tilanne (q', w') on tilanteen (q, w) välitön seuraaja.

Intuitio: automaatti ollessaan tilassa q ja lukiessaan nauhalla olevan merkkijonon $w = aw'$ ensimmäisen merkin a siirtyy tilaan q' ja siirtää nauhapäätä yhden askelen eteenpäin, jolloin nauhalle jää merkkijono w' .

Jos automaatti M on yhteydestä selvä, relaatiota voidaan merkitä yksinkertaisesti

$$(q, w) \vdash (q', w').$$

Tilanne (q, w) johtaa tilanteeseen (q', w') t. tilanne (q', w') on tilanteen (q, w) seuraaja, merkitään

$$(q, w) \vdash_M^* (q', w'),$$

jos on olemassa välitilannejono $(q_0, w_0), (q_1, w_1), \dots, (q_n, w_n)$, $n \geq 0$, siten että

$$(q, w) = (q_0, w_0) \vdash_M (q_1, w_1) \vdash_M \dots \vdash_M (q_n, w_n) = (q', w')$$

Erikoistapauksena $n = 0$ saadaan $(q, w) \vdash_M^* (q, w)$ millä tahansa tilanteella (q, w) .

Jälleen, jos automaatti M on yhteydestä selvä, merkitään yksinkertaisesti

$$(q, w) \vdash^* (q', w').$$

Automaatti M hyväksyy merkkijonon $x \in \Sigma^*$, jos on voimassa

$$(q_0, x) \vdash_M^* (q_f, \varepsilon) \quad \text{jollakin } q_f \in F;$$

muuten M hylkää x :n.

Toisin sanoen: automaatti hyväksyy x :n, jos sen alkutilanne syötteellä x johtaa, syötteen loppuessa, johonkin hyväksyvään lopputilanteeseen.

Automaatin M tunnustama kieli määritellään:

$$L(M) = \{x \in \Sigma^* \mid (q_0, x) \vdash_M^* (q_f, \varepsilon) \quad \text{jollakin } q_f \in F\}.$$

Tilanne (q, w) johtaa tilanteeseen (q', w') t. tilanne (q', w') on tilanteen (q, w) seuraaja, merkitään

$$(q, w) \vdash_M^* (q', w'),$$

jos on olemassa välitilannejono $(q_0, w_0), (q_1, w_1), \dots, (q_n, w_n)$, $n \geq 0$, siten että

$$(q, w) = (q_0, w_0) \vdash_M (q_1, w_1) \vdash_M \dots \vdash_M (q_n, w_n) = (q', w')$$

Erikoistapauksena $n = 0$ saadaan $(q, w) \vdash_M^* (q, w)$ millä tahansa tilanteella (q, w) .

Jälleen, jos automaatti M on yhteydestä selvä, merkitään yksinkertaisesti

$$(q, w) \vdash^* (q', w').$$

Esimerkki: merkkijonon "0.25E2" käsittely reaalityyppiautomaatilla:

$$\begin{array}{l} (q_0, 0.25E2) \vdash (q_1, .25E2) \vdash (q_2, 25E2) \\ \vdash (q_3, 5E2) \vdash (q_3, E2) \\ \vdash (q_4, 2) \vdash (q_6, \varepsilon). \end{array}$$

Koska $q_6 \in F = \{q_2, q_3, q_6\}$, on siis $0.25E2 \in L(M)$.