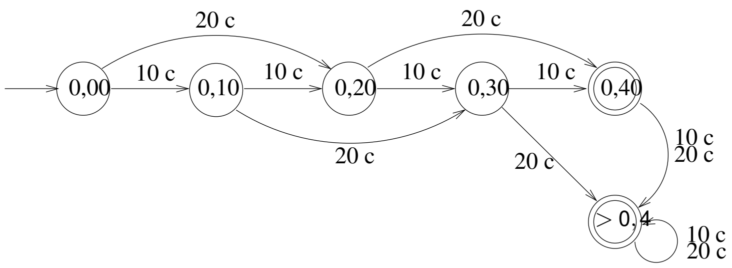
ÄÄRELLISET AUTOMAATIT JA SÄÄNNÖLLISET KIELET

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, ...

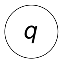
Esimerkki 1: Kahviautomaatti.



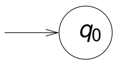
Em. tilakaavion esittämä automaatti ratkaisee päätösongelman "riittävätkö annetut rahat kahvin ostamiseen?"

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

Tilakaavioiden merkinnät:

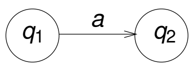


Automaatin tila nimeltä q



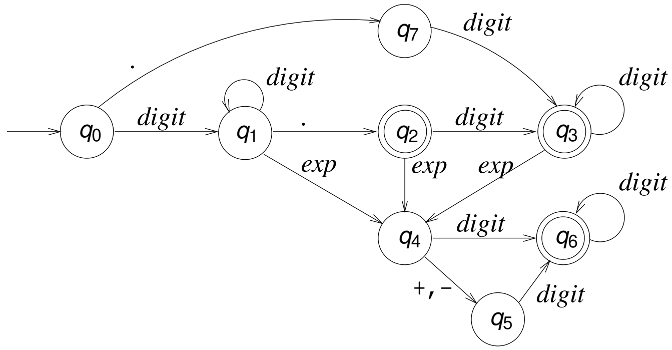
Alkutila

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



Syötemerkin a aikaansaama siirtymä tilasta q_1 tilaan q_2

Esimerkki 2: C-kielen etumerkittömät reaaliluvut.



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

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

Esim. reaalilukuautomaatin tilataulu:

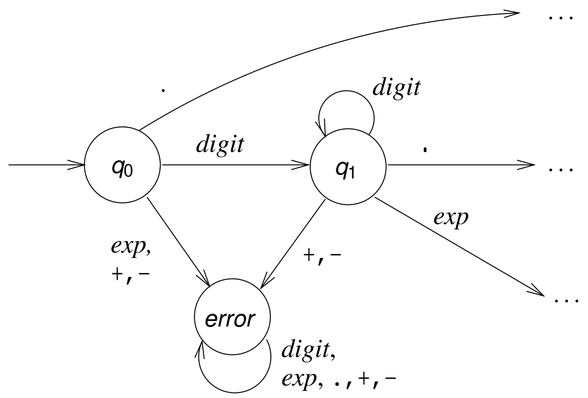
		digit	.	*exp*	$+$	$-$
$\rightarrow$	q_0	q_1	q_7			
	q_1	q_1	q_2	q_4		
$\leftarrow$	q_2	q_3		q_4		
$\leftarrow$	q_3	q_3		q_4		
	q_4	q_6			q_5	q_5
	q_5	q_6				
$\leftarrow$	q_6	q_6				
	q_7	q_3				

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 merkitä näkyviin.

Esim. reaalilukuautomaatin täydellinen kaavioesitys olisi:



ja reaalilukuautomaatin täydellinen tauluesitys olisi:

		digit	.	*exp*	$+$	$-$
$\rightarrow$	q_0	q_1	q_7	*error*	*error*	*error*
	q_1	q_1	q_2	q_4	*error*	*error*
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\leftarrow$	q_6	q_6	*error*	*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. reaalilukuautomaattiin perustuva syötejonon syntaksitestaus:

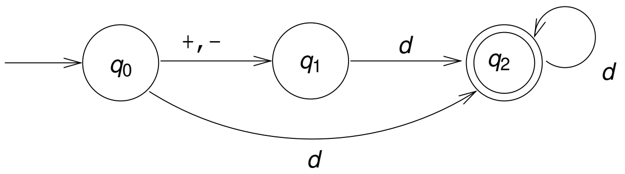

```
#include <stdio.h>
#include <ctype.h>
main() {
   int q, c;
   q = 0;
   while ((c = getchar()) != '\n')
      switch (q) {
      case 0:
         if (isdigit(c)) q = 1;
         else if (c == '.') q = 7;
         else q = 99;
         break;
      case 1:
         if (isdigit(c)) q = 1;
       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



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


Semanttisten toimintojen liittäminen äärellisiin automaatteihin

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



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

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 syöteluvun arvon laskevilla 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;                             /* SEM */
        q = 1;
      }
      else if ('0' <= c && c <= '7') {
        val = c - '0';                        /* SEM */
        q = 2;
      }
      else q = 99;
      break;
```

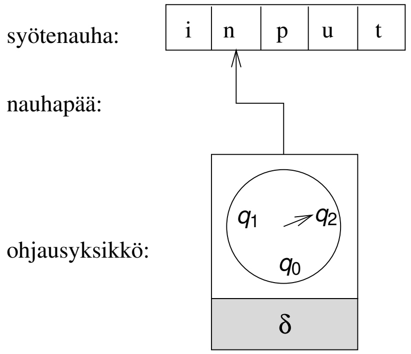
```
case 1:
  if ('0' <= c && c <= '7') {
    val = c - '0';                      /* SEM */
    q = 2;
  }
  else q = 99;
  break;
case 2:
  if ('0' <= c && c <= '7') {
    val = 8 * val + (c - '0');          /* SEM */
    q = 2;
  }
  else q = 99;
  break;
case 99:
  break;
}
```



```
  }
  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 äärellistilaisesta *ohjausyksiköstä*, jonka toimintaa säätelee automaatin *siirtymäfunktio* δ, sekä merkkipaikkoihin jaetusta *syötenauhasta* ja nämä yhdistävästä *nauhapäästä*, 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-askelessa automaatti lukee nauhapään kohdalla olevan syötemerkin, päättää ohjausyksikön tilan ja luetun merkin perusteella siirtymäfunktion mukaisesti ohjausyksikön uudesta tilasta, ja siirtää nauhapäätä yhden merkin eteenpäin.

Automaatti pysähtyy, kun viimeinen syötemerkki on käsitelty. Jos ohjausyksikön 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 *syöteaakkosto*;
- $\delta : Q \times \Sigma \to Q$ on automaatin *siirtymäfunktio*;
- $q_0 \in Q$ on automaatin *alkutila*;
- $F \subseteq Q$ on automaatin *(hyväksyvien) lopputilojen* joukko.

Esimerkki. Reaalilukuautomaatin formaali esitys:

$$\begin{aligned} M \; = \; & (\{q_0, \ldots, q_7, \mathit{error}\}, \{\mathtt{0}, \mathtt{1}, \ldots, \mathtt{9}, \mathtt{.}, \mathtt{E}, \mathtt{e}, \mathtt{+}, \mathtt{-}\}, \\ & \delta, q_0, \{q_2, q_3, q_6\}), \end{aligned}$$

missä δ on kuten aiemmin taulukossa; esim.

$$\delta(q_0, \mathtt{0}) = \delta(q_0, \mathtt{1}) = \cdots = \delta(q_0, \mathtt{9}) = q_1,$$

$$\delta(q_0, \mathtt{.}) = q_7, \quad \delta(q_0, \mathtt{E}) = \delta(q_0, \mathtt{e}) = \mathit{error},$$

$$\delta(q_1, \mathtt{.}) = q_2, \quad \delta(q_1, \mathtt{E}) = \delta(q_1, \mathtt{e}) = q_4,$$

jne.

Automaatin *tilanne* on pari $(q, w) \in Q \times \Sigma^*$; erityisesti automaatin *alkutilanne syötteellä x* on pari (q_0, x).

Intuitio: q on automaatin tila ja w on syötemerkkijonon jäljellä oleva, so. nauhapäästä oikealle sijaitseva osa.

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

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

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

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

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

$$(q, w) = (q_0, w_0) \underset{M}{\vdash} (q_1, w_1) \underset{M}{\vdash} \cdots \underset{M}{\vdash} (q_n, w_n) = (q', w')$$

Erikoistapauksena $n = 0$ saadaan $(q, w) \underset{M}{\vdash^*} (q, w)$ 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

$$(q_0, x) \underset{M}{\vdash^*} (q_f, \varepsilon) \qquad \text{jollakin } q_f \in F;$$

muuten M *hylkää* x:n.

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

Automaatin M *tunnistama kieli* määritellään:

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

Esimerkki: merkkijonon "0.25E2" käsittely reaalilukuautomaatilla:

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

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