

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

Esim. reaalitylukuautomaatin tilataulu:

	<i>digit</i>	.	<i>exp</i>	+	-
→ q_0	q_1	q_7			
q_1	q_1	q_2	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				



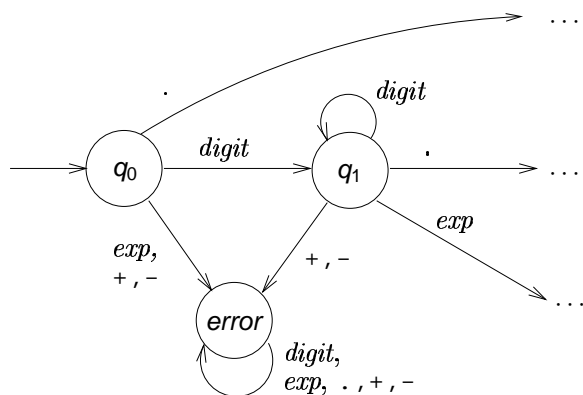
K: Mitä tilataulun tyhjät paikat tarkoittavat?

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 merkitä näkyviin.



Esim. reaalitylukuautomaatin täydellinen kaavioesitys olisi:



ja reaalitylukuautomaatin täydellinen tauluesitys olisi:

	<i>digit</i>	.	<i>exp</i>	+	-
→ q_0	q_1	q_7	<i>error</i>	<i>error</i>	<i>error</i>
q_1	q_1	q_2	q_4	<i>error</i>	<i>error</i>
⋮	⋮	⋮	⋮	⋮	⋮
← q_6	q_6	<i>error</i>	<i>error</i>	<i>error</i>	<i>error</i>
<i>error</i>	<i>error</i>	<i>error</i>	<i>error</i>	<i>error</i>	<i>error</i>



2.2 Äärellisiin automaateihin perustuva ohjelmointi

Annetun äärellisen automaatin pohjalta on helppo laatia automaatin toimintaa vastaava ohjelma. Esim. reaalitylukuautomaattiin 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;
```

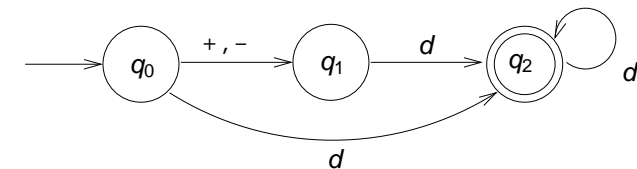


```
...
case 99:
    break;
}
if (q == 2 || q == 3 || q == 6)
    printf("SYÖTE ON REAALILUKU.\n");
else
    printf("SYÖTE EI OLE REAALILUKU.\n");
}
```



Semanttisten toimintojen liittäminen äärellisiin automaateihin

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 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';
    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;
}

```



```

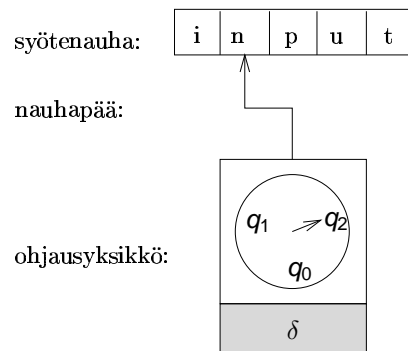
}
if (q == 2)
{ printf("LUVUN ARVO ON %d.\n", sgn*val); /*SEM*/
  exit(0); }
else
{ printf("VIRHEELLINEN SYÖTE.\n"); /* SEM */
  exit(1); }
}

```



2.3 Äärellisen automaatin käsitteen formalisointi

Mekanistinen malli:



Äärellinen automaatti M koostuu äärellistilaisesta *ohjausyksiköstä*, jonka toimintaa säätelee automaatin *siirtymäfunktio* δ , sekä merkkipaikkoihin jaetusta *syötenauhasta* ja nämä yhdistävästä *nauhapäätstä*, 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-askellessa 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*;
- ▶ $\delta : Q \times \Sigma \rightarrow Q$ on automaatin *siirtymäfunktio*;
- ▶ $q_0 \in Q$ on automaatin *alkutila*;
- ▶ $F \subseteq Q$ on automaatin (*hyväksyvien*) *lopputilojen* joukko.



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.



Esimerkki. Reaalilukuautomaatin formaali esitys:

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

missä δ on kuten aiemmin taulukossa; esim.

$$\delta(q_0, 0) = \delta(q_0, 1) = \dots = \delta(q_0, 9) = q_1,$$

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

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

jne.



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 tunnistama kieli määritellään:

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



Esimerkki: merkkijonon "0.25E2" käsittely reaalityöautomaatilla:

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

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

