

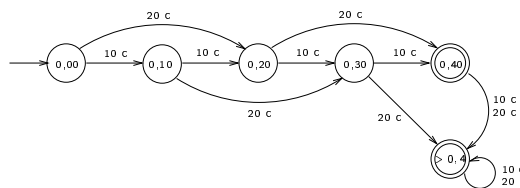
ÄÄRELLISET AUTOMAATIT JA SÄÄNNÖLLISET KIELET

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, tilataulukot, ...

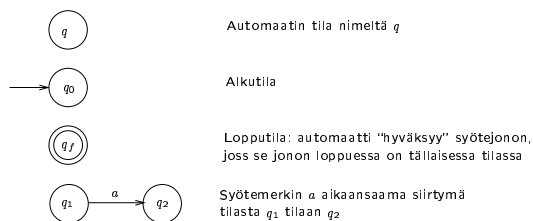
Esimerkki 1: Kahviautomaatti.



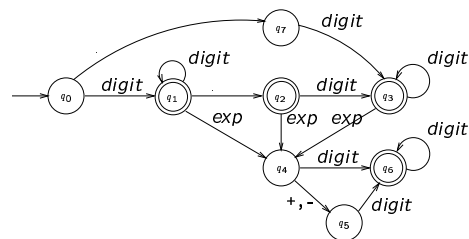
Em. tilakaavion esittämä automaatti ratkaisee päätösongelman "riittävätkö annetut rahat kahvin ostamiseen?"

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:



Esimerkki 2: C-kielen etumerkittömät reaalityyvat.



Käytetyt lyhenteet:

$$digit = \{0, 1, \dots, 9\},$$

$$exp = \{E, e\}.$$

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

Esim. reaalitylukuautomaatin tilataulu:

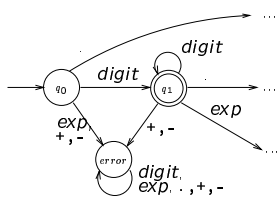
		<i>digit</i>	.	<i>exp</i>	+	-
→	<i>q</i> ₀	<i>q</i> ₁	<i>q</i> ₇			
←	<i>q</i> ₁	<i>q</i> ₁	<i>q</i> ₂	<i>q</i> ₄		
←	<i>q</i> ₂	<i>q</i> ₃		<i>q</i> ₄		
←	<i>q</i> ₃	<i>q</i> ₃		<i>q</i> ₄		
	<i>q</i> ₄	<i>q</i> ₆			<i>q</i> ₅	<i>q</i> ₅
	<i>q</i> ₅	<i>q</i> ₆				
←	<i>q</i> ₆	<i>q</i> ₆				
	<i>q</i>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

K: Mitä tilataulukon tyhjät paikat tarkoittavat?

V: Tilataulukon 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 vastaava tauluesitys:

		<i>digit</i>	.	<i>exp</i>	+	-
→	<i>q</i> ₀	<i>q</i> ₁	<i>q</i> ₇	<i>error</i>	<i>error</i>	<i>error</i>
←	<i>q</i> ₁	<i>q</i> ₁	<i>q</i> ₂	<i>q</i> ₄	<i>error</i>	<i>error</i>
⋮	⋮	⋮	⋮	⋮	⋮	⋮
←	<i>q</i> ₆	<i>q</i> 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 automaatteihin perustuva ohjelmointi

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

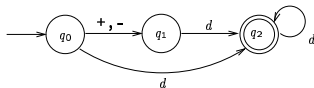
```

```
                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 == 1 || q == 2 || q == 3 || q == 6)
        printf("SYÖTE ON REAALILUKU.\n");
    else
        printf("SYÖTE EI OLE REAALILUKU.\n");
}
```

10

Semanttisten toimintojen liittämisen äärellisiin automaatteihin

Esimerkki. Kokonaisyttöluksautomaatti ja siihen perustuva syöteluvun arvonmääritys.



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

11

Täydennys syöteluvun arvon laskevilla operaatioilla:

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

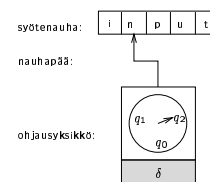
main()
{
    int q, c;
    int sgn, val;
    sgn = 1; val = 0;
    q = 0;
    while ((c = getchar()) != '\n') {
        switch (q) {
            case 0:
                if (c == '+') q = 1;
                else if (c == '-') {
                    sgn = -1;
                    q = 1;
                }
            case 99:
                break;
        }
    }
    if (q == 2)
        printf("LUVUN ARVO ON %d.\n", sgn*val);
    else
        printf("VIRHEELLINEN LUKU.\n");
}
```

12

```
else if (isdigit(c)) {
    val = c - '0';
    q = 2;
}
else q = 99;
break;
case 1:
    if (isdigit(c)) {
        val = c - '0';
        q = 2;
    }
    else q = 99;
    break;
case 2:
    if (isdigit(c)) {
        val = 10 * val + (c - '0');
        q = 2;
    }
    else q = 99;
    break;
```

2.3 Äärellisen automaatin käsitteen formalisointi

Mekanistinen malli:



Äärellinen automaatti M koostuu äärellistilaisesta ohjausyksiköstä, jonka toimintaa säätelee automaatin siirtymäfunktio δ , sekä merkki-paikkoihin 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-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.

14

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.

15

Esimerkki. Reaalilukuautomaatin formaali esitys:

$$M = (\{q_0, \dots, q_7, \text{error}\}, \{0, 1, \dots, 9, \cdot, \text{E}, \text{e}, +, -\}, \delta, q_0, \{q_1, q_2, q_3, q_6\}),$$

missä δ on kuten aiemmin taulukossa; esim.

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

16

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.

17

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

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

18

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

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

19

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:

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

20

Esimerkki: merkkijonon "0.25E2" käsittely reaalilukuautomaatilla:

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

Koska $q_6 \in F = \{q_1, q_2, q_3, q_6\}$, on siis $0.25E2 \in L(M)$.

21