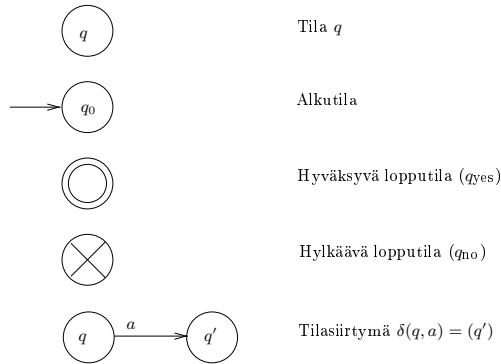


Kahviautomaatti voidaan esittää *tilasiirtymäkaaviona* (Kuva 4.1) tai seuraavanlaisena *tilasiirtymätaulukkona*:

	50 snt	1 euro
$\rightarrow q_0$	q_1	q_2
q_1	q_2	q_3
q_2	q_3	q_4
q_3	q_4	q_4
$\leftarrow q_4$	q_4	q_4

4.1.1 Äärellisen automaatin esitys

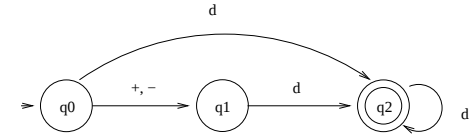
Tavallisesti äärellinen automaatti voidaan esittää tilasiirtymäkaaviona. Tilasiirtymäkaavion merkinnät on esitelty kuvassa 4.2. Huomaa, että hylkääviä lopputiloja (virhetiloja) ei ole yleensä tapana piirtää kaavioihin. Jos jostain tilasta ei löydy siirtymää annetulla syötteellä, päädytään aina hylkäävään lopputilaan.



Kuva 4.2: Äärellisen automaatin kaavioesituksen merkinnät.

Yhtä hyvin automaatti voidaan esittää tilasiirtymätaulukkona tai tietokoneohjelmana. Eri esitystapoja on havainnollistettu seuraavissa esimerkeissä.

Esimerkki 4.1 Laaditaan automaatti, joka tunnistaa etumerkilliset (10-järjestelmän) kokonaisluvut:



Tilasiirtymätaulukkona:

		d	$+, -$
$\rightarrow$	0	2	1
	1	2	
$\leftarrow$	2	2	

Tässä $d = \{0, 1, \dots, 9\}$. Taulukon puuttuvat kohdat vastaavat virhetilaa "Error", joka jätetään yleensä selkeyden vuoksi merkitsemättä.

Tämä voidaan toteuttaa yksinkertaisena tietokoneohjelmana:

```
int IsDigit(char c); /* palauttaa 1, jos c on lukumerkki, 0 muuten */
```

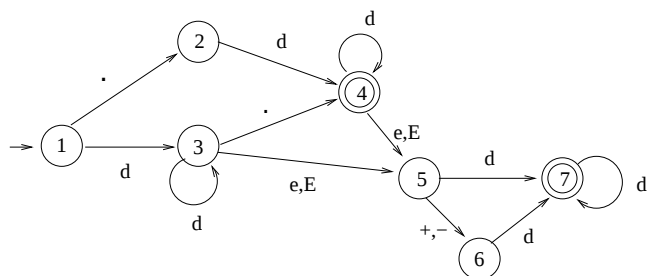
```
int q = 0;
char c;
while ((c = fgetc(stdin)) != EOF)
{
    switch (q)
    {
        case 0: if (c == '+' || c == '-')
                q = 1;
                else if (IsDigit(c))
                q = 2;
                break;
        case 1: if (IsDigit(c))
                q = 2;
                else q = 99;
                break;
        case 2: if (IsDigit(c))
                q = 2;
                else q = 99;
                break;
```

```

    case 99: break;
  }
}
if (q == 2) printf("luku OK!");
else printf("Virheellinen luku");

```

Esimerkki 4.2 C-kielen mukaisen etumerkittömän liukuluvun tunnistava automaatti:



Tilasiirtymätaulukko:

		d	.	e,E	+, -
→	1	3	2		
	2	4			
	3	3	4	5	
←	4	4		5	
	5	7			6
	6	7			
←	7	7			

Äärellisen automaatin pohjalta laadittuun ohjelmaan voidaan liittää myös semanttisia toimintoja:

Esimerkki 4.3 Laaditaan ohjelma, joka tunnistaa etumerkillisen kokonaisluvun ja lisäksi evaluoi luvun arvon. (Tämä vastaa siis string-to-integer muunnosta, jota esim. C:ssä vastaa funktio stol.)

```

int IsDigit(char c); /* palauttaa 1, jos c on numero, 0 muuten */

```

```

int q = 0;
char c;
int sign = 1;
int val = 0;
while ((c = fgetc(stdin)) != EOF)
{
  switch (q)
  {
    case 0: if (c == '+' || c == '-') {
              q = 1;
              if (c == '-') sign = -1;
            }
            else if (IsDigit(c)) {
              q = 2;
              val = c - '0';
            }
            break;
    case 1: if (IsDigit(c)) {
              q = 2;
              val = c - '0';
            }
            else q = 99;
            break;
    case 2: if (IsDigit(c)) {
              q = 2;
              val = 10 * val + (c - '0');
            }
            else q = 99;
            break;
    case 99: break;
  }
}
if (q == 2) printf("luvun arvo on %d", sign * val);
else printf("Virheellinen luku");

```