T-79.1001/2 **Syksy 2005**
Tietojenkäsittelyteorian perusteet (T/Y)
Harjoitus 2
Demonstraatiotehtävien ratkaisut

4. **Tehtävä:** Osoita, että mikä tahansa vähintään kaksimerkkinen aakkosto Σ on samanveroinen binääriaakkoston $\Gamma = \{0,1\}$ kanssa siinä mielessä, että Σ:n merkkijonot voidaan helposti koodata Γ:n merkkijonoiksi ja kääntäen. Miten paljon merkkijonon pituus voi muuttua suunnittelemassasi koodauksessa? (Siis jos merkkijonon $w \in \Sigma^*$ pituus on $|w| = n$ merkkiä, mikä on sen vastinjonon $w' \in \Gamma^*$ pituus?) Onnistuisiko vastaava koodaus, jos kohdeaakkostossa olisikin vain *yksi* merkki, esim. $\Gamma = \{1\}$?

   Vastaus: Olkoon Σ jokin k-merkkinen aakkosto, $k > 1$. Σ:n merkkijonot voidaan koodata $\Gamma = \{0,1\}$:n merkkijonoiksi seuraavasti:

   - Samaistetaan Σ:n aakkosto kokonaislukujen $\{1, \ldots, k\}$ kanssa.
   - Nämä luvut (eli Σ:n aakkosto) voidaan esittää $\lceil \log_2 k \rceil$:n mittaisina binäärilukuina.
   - Jokainen Σ:n merkkijono voidaan esittää Γ:n merkkijonona korvaamalla Σ:n aakkoset merkkijonossa niiden binäärikoodauksella.

   Koodauksen purku $\Gamma^* \to \Sigma^*$ onnistuu vastaavasti ottamalla merkkijonosta $\lceil \log_2 k \rceil$:n mittaiset merkkijonot ja tulkitsemalla ne Σ:n aakkosiksi.

   Jos merkkijonon $w \in \Sigma^*$ pituus on $|w| = n$ merkkiä on tällöin sen vastinjonon $w' \in \Gamma^*$ pituus $|w'| = n \cdot \lceil \log_2 k \rceil$, sillä jokaisen Σ:n merkin koodaamiseen tarvitaan $\lceil \log_2 k \rceil$ merkkiä.

   Tarkastellaan esimerkiksi aakkostoa $\Sigma = \{a,b,c,d,e,f\}$ ja merkkijonoa *aacfd*. Koska $|\Sigma| = 6$, tarvitaan symbolien esittämiseen $\lceil \log_2 6 \rceil = \lceil 2.58 \rceil = 3$ bittiä. Yksi mahdollinen koodaus on:

   $$\begin{array}{ll} a \mapsto 001 & d \mapsto 100 \\ b \mapsto 010 & e \mapsto 101 \\ c \mapsto 011 & f \mapsto 110 \end{array}$$

   Tällöin jonon *aacfd* esitys on 001001011110100.

   Vastaavaa koodausta ei voida rakentaa, mikäli $\Gamma = \{1\}$. Luvuille voidaan kyllä määrittää unaariesitys seuraavasti: $1 \mapsto 1$, $2 \mapsto 11$, $3 \mapsto 111$, ..., mutta näin syntynyttä koodia ei voida enää purkaa yksikäsitteisesti. Esim. lukujot 1 1 1, 1 2, 2 1 ja 3 koodautuvat kaikki merkkijonoksi 111.

5. **Tehtävä:** Laadi äärelliset automaatit seuraavien kielten tunnistamiseen:

   (a) $\{a^m b^n \mid m = n \mod 3\}$;

   (b) $\{w \in \{a,b\}^* \mid w$ sisältää yhtä monta a:ta ja b:tä, modulo 3$\}$.

   (Merkintä "$m = n \mod 3$" tarkoittaa, että luvut m ja n antavat kolmella jaettaessa saman jakojäännöksen.)

   Vastaus:

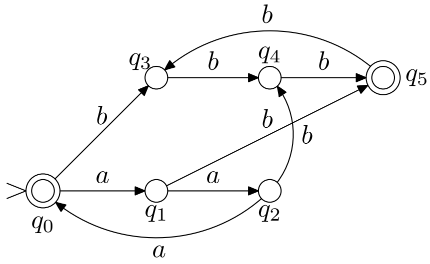
   a) Kielen $L = \{a^m b^n \mid m = n \mod 3\}$ tunnistaa seuraava äärellinen automaatti:

   $$\begin{aligned} M &= (Q, \Sigma, \delta, q_0, F) \\ Q &= \{q_0, q_1, q_2, q_3, q_4, q_5, q_6\} \\ \Sigma &= \{a, b\} \\ F &= \{q_0, q_5\} \end{aligned}$$

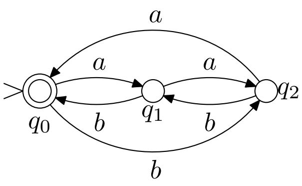
   Tilansiirtofunktio δ määritellään seuraavasti:

q	$\delta(q,a)$	$\delta(q,b)$
q_0	q_1	q_3
q_1	q_2	q_5
q_2	q_0	q_4
q_3	q_6	q_4
q_4	q_6	q_5
q_5	q_6	q_3
q_6	q_6	q_6

Tässä tilaa q_6 käytetään hylkäystilana, johon siirrytään heti, kun on varmaa että sana ei voi enää kuulua kieleen (kun sanasta löytyy alijono ba), ja jossa pysytään aina laskennan loppuun asti. Tällaiset tilat jätetään usein merkitsemättä, kun tilakone esitetään konekaaviona. Näin on tehty myös alla olevan kaavion tapauksessa:

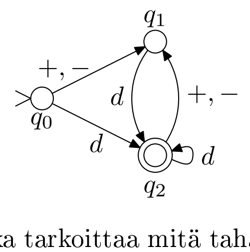


b) Kielen $L = \{w \in \{a,b\}^* \mid w$ sisältää yhtä monta a- ja b-kirjainta modulo 3$\}$ tunnistaa seuraava tilakone:



6. **Tehtävä:** Laadi äärellinen automaatti, joka tunnistaa yhteen- ja vähennyslaskumerkein toisistaan erotettujen kokonaislukujen jonoja (esim. $11 + 20 - 9$, $-5 + 8$). Toteuta automaattisi tietokoneohjelmana, joka myös laskee lukujonon arvon.

   Vastaus: Kokonaislukujen yhteen- ja vähennyslaskut voidaan tunnistaa seuraavalla automaatilla:



Tässä d on lyhennysmerkintä, joka tarkoittaa mitä tahansa numeroa joukosta $\{0, \ldots, 9\}$.

Automaatti voidaan toteuttaa varsin suoraviivaisesti millä tahansa proseduraalisella ohjelmointikielellä lisäämällä siihen summan laskenta ja virheellisen syötteen käsittely. Alla on esitetty C-kielinen toteutus:

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

int main (void)
{
  int q;    /* tila */
  int c;    /* syötemerkki */
```

```
int sgn, val, sum;  /* apumuuttujia */
sgn=1; val = 0, sum = 0;
q = 0;

while ((c = getchar()) !='\n') {
  switch (q) {
  case 0:
    if (c == '+') q = 1;
    else if (c == '-') {
      sgn = -1;
      q = 1;
    }
    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 if (c == '+')
      {
        sum = sum + val*sgn;
        val = 0;
        sgn = 1;
        q = 1;
      }
    else if (c == '-')
      {
        sum = sum + val*sgn;
        val = 0;
        sgn = -1;
        q = 1;
      }
    else q = 99;
    break;

  case 99:
    break;
  }
}
sum = sum + sgn*val;
```

```
  if (q == 2)
    printf("LUVUN ARVO ON %d.\n", sum);
  else
    printf("VIRHEELLINEN LUKU.\n");
  return;
}
```