1.7 Numeroituvat ja ylinumeroituvat joukot

Määritelmä 1.10 Joukko X on *numeroituvasti ääretön*, jos on olemassa bijektio $f : \mathbb{N} \to X$. Joukko on *numeroituva*, jos se on äärellinen tai numeroituvasti ääretön. Joukko, joka ei ole numeroituva on *ylinumeroituva*.

Intuitiivisesti sanoen joukko X on numeroituva, jos sen alkiot voidaan järjestää ja indeksoida luonnollisilla luvuilla:

$$X = \{x_0, x_1, x_2, \dots, x_{n-1}\},$$

jos X on n-alkioinen äärellinen joukko ja

$$X = \{x_0, x_1, x_2, \dots\},$$

jos X on numeroituvasti ääretön.

Numeroituvan joukon kaikki osajoukot ovat myös numeroituvia (HT), mutta ylinumeroituvilla joukoilla on sekä numeroituvia että ylinumeroituvia osajoukkoja. Siten ylinumeroituvat joukot ovat jossain mielessä "isompia" kuin numeroituvat.

Lause 1.11 Minkä tahansa aakkoston Σ merkkijonojen joukko Σ^* on numeroituvasti ääretön.

Todistus. Muodostetaan bijektio $f : \mathbb{N} \to \Sigma^*$ seuraavasti. Olkoon $\Sigma = \{a_1, a_2, \dots, a_n\}$. Kiinnitetään Σ:n merkeille jokin "aakkosjärjestys"; olkoon se $a_1 < a_2 < \cdots < a_n$.

Joukon Σ^* merkkijonot voidaan nyt luetella valitun aakkosjärjestyksen suhteen *kanonisessa* t. *leksikografisessa järjestyksessä* (engl. canonical t. lexicographic order) seuraavasti:

- ensin luetellaan 0:n mittaiset merkkijonot (= ε), sitten 1:n mittaiset (= $a_1, a_2, \dots, a_n$), sitten 2:n mittaiset jne.;
- kunkin pituusryhmän sisällä merkkijonot luetellaan aakkosjärjestyksessä.

Bijektio f on siis:

$$\begin{array}{rcl}
0 & \mapsto & \varepsilon \\
1 & \mapsto & a_1 \\
2 & \mapsto & a_2 \\
\vdots & & \vdots \\
n & \mapsto & a_n \\
n+1 & \mapsto & a_1 a_1 \\
n+2 & \mapsto & a_1 a_2 \\
\vdots & & \vdots \\
2n & \mapsto & a_1 a_n
\end{array}
\qquad
\begin{array}{rcl}
2n+1 & \mapsto & a_2 a_1 \\
\vdots & & \vdots \\
3n & \mapsto & a_2 a_n \\
\vdots & & \vdots \\
n^2+n & \mapsto & a_n a_n \\
n^2+n+1 & \mapsto & a_1 a_1 a_1 \\
n^2+n+2 & \mapsto & a_1 a_1 a_2 \\
\vdots & & \vdots
\end{array}$$

□

Itse asiassa millä tahansa ohjelmointikielellä kirjoitetut ohjelmat ovat kielen perusaakkoston (esim. C-kielessä ASCII-merkistön) merkkijonoja. Lauseen 1.11 mukaan minkä tahansa aakkoston merkkijonojen joukko on numeroituvasti ääretön, joten myös millä tahansa ohjelmointikielellä mahdollisten ohjelmien joukko on numeroituva.

Seuraavaksi todistetaan, että kaikkien formaalien kielten joukko on ylinumeroituva. Formaaleja kieliä on siis "enemmän" kuin mahdollisia tietokoneohjelmia, ja siksi *millään ohjelmointikielellä ei voida laatia tunnistusautomaatteja kaikille formaaleille kielille*. (Tai toisin sanoen: on olemassa "periaatteessa mahdollisia" I/O-kuvauksia, joita ei voida toteuttaa tietokoneella.)

Lause 1.12 Minkä tahansa aakkoston Σ kaikkien formaalien kielten perhe on ylinumeroituva.

Todistus (ns. Cantorin diagonaaliargumentti). Merkitään aakkoston Σ kaikkien formaalien kielten perhettä $\mathcal{P}(\Sigma^*) = \mathcal{A}$. Tehdään vastaoletus: oletetaan, että olisi olemassa kaikki Σ:n formaalit kielet kattava numerointi:

$$\mathcal{A} = \{A_0, A_1, A_2, \ldots\}.$$

Olkoot Σ^*:n merkkijonot kanonisessa järjestyksessä $x_0, x_1, x_2, \ldots$. Määritellään em. numerointeja käyttäen formaali kieli $\tilde{A}$:

$$\tilde{A} = \{x_i \in \Sigma^* \mid x_i \notin A_i\}.$$

Koska $\tilde{A} \in \mathcal{A}$ ja $\mathcal{A}$:n numerointi oletettiin kattavaksi, pitäisi olla $\tilde{A} = A_k$ jollakin $k \in \mathbb{N}$. Mutta tällöin $\tilde{A}$:n määritelmän mukaan

$$x_k \in \tilde{A} \Leftrightarrow x_k \notin A_k = \tilde{A}.$$

Saatu ristiriita osoittaa, että vastaoletus on väärä. □

$$\tilde{A}$$

$$\begin{array}{c|ccccc}
\searrow & A_0 & A_1 & A_2 & A_3 & \cdots \\
\hline
x_0 & \overset{1}{\cancel{0}} & 0 & 0 & 1 & \cdots \\
x_1 & 0 & \overset{0}{\cancel{1}} & 0 & 0 & \cdots \\
x_2 & 1 & 1 & \overset{0}{\cancel{1}} & 1 & \cdots \\
x_3 & 0 & 0 & 0 & \overset{1}{\cancel{0}} & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots
\end{array}$$

Kuvallisesti todistuksen idea voidaan esittää seuraavasti. Muodostetaan kielten $A_0, A_1, A_2, \ldots$ ja merkkijonojen $x_0, x_1, x_2, \ldots$ "insidenssimatriisi", jonka rivin i sarakkeessa j on arvo 1 jos $x_i \in A_j$ ja muuten 0. Tällöin kieli $\tilde{A}$ poikkeaa kustakin kielestä A_k matriisin "diagonaalilla":

1.8 *Ekskursio: Turingin pysähtymisongelma

Lauseiden 1.11 ja 1.12 mukaan on siis olemassa formaaleja kieliä (I/O-kuvauksia), joita ei voida toteuttaa esim. C-ohjelmilla. Entä jokin *konkreettinen esimerkki* tällaisesta?

Tunnetuin esimerkki on ns. *Turingin pysähtymisongelma*. (Alan Turing, 1936). C-ohjelmia käyttäen tulos voidaan muotoilla seuraavasti:

Väite. Ei ole olemassa C-funktiota `halt(p,x)`, joka saa syötteenään mielivaltaisen C-funktion tekstin `p` ja tälle tarkoitetun syötteen `x` ja tuottaa tuloksen 1, jos `p`:n suoritus pysähtyy syötteellä `x`, ja 0 jos `p`:n suoritus `x`:llä jää ikuiseen silmukkaan.

Todistus. Oletetaan väitteen vastaisesti, että tällainen funktio `halt` voitaisiin laatia. Muodostetaan tätä käyttäen toinen funktio `confuse` (ks. alla).

Merkitään funktion `confuse` ohjelmatekstiä `c`:llä ja tarkastellaan funktion `confuse` laskentaa tällä omalla kuvauksellaan.

```
void confuse(char *p){
   int halt(char *p, char *x){
   ... /* Funktion halt runko. */
   }
   if (halt(p,p) == 1) while (1);
}
```

Saadaan ristiriita:

`confuse(c)` pysähtyy

$\Leftrightarrow$

`halt(c,c) == 1`

$\Leftrightarrow$

`confuse(c)` ei pysähdy.

Ristiriidasta seuraa, että oletettua pysähtymistestausfunktiota `halt` ei voi olla olemassa. □

Samansukuisia ns. *ratkeamattomia ongelmia* on itse asiassa *paljon*. Asiaan palataan kurssin loppupuolella.

2.8 Säännöllisten kielten rajoituksista

Kardinaliteettisyistä on oltava olemassa (paljon) ei-säännöllisiä kieliä: kieliä on ylinumeroituva määrä, säännöllisiä lausekkeita vain numeroituvasti.

Voidaanko löytää konkreettinen, *mielenkiintoinen* esimerkki kielestä, joka ei olisi säännöllinen? Helposti.

Säännöllisten kielten perusrajoitus: äärellisillä automaateilla on vain rajallinen “muisti”. Siten ne eivät pysty ratkaisemaan ongelmia, joissa vaaditaan mielivaltaisen suurten lukujen tarkkaa muistamista.

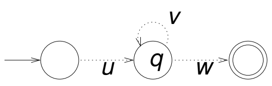
Esimerkki: sulkulausekekieli

$$L_{\text{match}} = \{ (^k)^k \mid k \geq 0 \}.$$

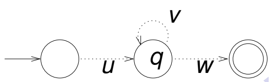
Formalisointi: “pumppauslemma”.

Lemma 2.6 (Pumppauslemma) Olkoon A säännöllinen kieli. Tällöin on olemassa sellainen $n \geq 1$, että mikä tahansa $x \in A$, $|x| \geq n$, voidaan jakaa osiin $x = uvw$ siten, että $|uv| \leq n$, $|v| \geq 1$, ja $uv^i w \in A$ kaikilla $i = 0, 1, 2, \ldots$

Todistus. Olkoon M jokin A:n tunnistava deterministinen äärellinen automaatti, ja olkoon n M:n tilojen määrä. Tarkastellaan M:n läpikäymiä tiloja syötteellä $x \in A$, $|x| \geq n$. Koska M jokaisella x:n merkillä siirtyy tilasta toiseen, sen täytyy kulkea jonkin tilan kautta (ainakin) kaksi kertaa — itse asiassa jo x:n n:n ensimmäisen merkin aikana. Olkoon q ensimmäinen toistettu tila.



Olkoon u M:n käsittelemä x:n alkuosa sen tullessa ensimmäisen kerran tilaan q, v se osa x:stä jonka M käsittelee ennen ensimmäistä paluutaan q:hun, ja w loput x:stä. Tällöin on $|uv| \leq n$, $|v| \geq 1$, ja $uv^i w \in A$ kaikilla $i = 0, 1, 2, \ldots$ □



Esimerkki. Tarkastellaan em. sulkulausekekieltä (merk. ‘(’ = a, ‘)’ = b):

$$L = L_{\text{match}} = \{ a^k b^k \mid k \geq 0 \}.$$

Oletetaan, että L olisi säännöllinen. Tällöin pitäisi pumppauslemman mukaan olla jokin $n \geq 1$, jota pitempiä L:n merkkijonoja voidaan pumpata. Valitaan $x = a^n b^n$, jolloin $|x| = 2n > n$. Lemman mukaan x voidaan jakaa pumpattavaksi osiin $x = uvw$, $|uv| \leq n$, $|v| \geq 1$; siis on oltava

$$u = a^i,\ v = a^j,\ w = a^{n-(i+j)} b^n, \quad i \leq n-1,\ j \geq 1.$$

Mutta esimerkiksi “0-kertaisesti” pumpattaessa:

$$uv^0 w = a^i a^{n-(i+j)} b^n = a^{n-j} b^n \notin L.$$

Siten L ei voi olla säännöllinen.

3.8 Yhteydettömien kielten rajoituksista

Yhteydettömille kielille on voimassa säännöllisten kielten pumppauslemman vastine. Nyt kuitenkin merkkijonoa on pumpattava samanaikaisesti kahdesta paikasta.

Lemma 3.9 ("uvwxy-lemma") Olkoon L yhteydetön kieli. Tällöin on olemassa sellainen $n \geq 1$, että mikä tahansa $z \in L$, $|z| \geq n$, voidaan jakaa osiin $z = uvwxy$ siten, että

(i) $|vx| \geq 1$,

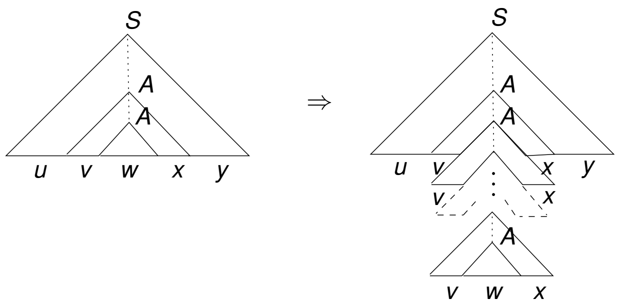
(ii) $|vwx| \leq n$,

(iii) $uv^iwx^iy \in L$ kaikilla $i = 0, 1, 2, \ldots$.

Todistus. Olkoon $G = (V, \Sigma, P, S)$ Chomskyn normaalimuotoinen kielioppi L:lle. Tällöin missä tahansa G:n jäsennyspuussa, jonka korkeus on h, on enintään 2^h lehteä. Toisin sanoen, minkä tahansa $z \in L$ jokaisessa jäsennyspuussa on polku, jonka pituus on vähintään $\log_2 |z|$.

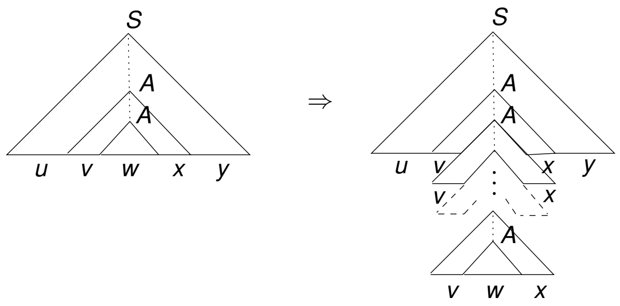
Olkoon $k = |V - \Sigma|$ kieliopin G välikkeiden määrä. Asetetaan $n = 2^{k+1}$. Tarkastellaan jotakin $z \in L$, $|z| \geq n$, ja sen jotakin jäsennyspuuta.

Edellisen nojalla puussa on polku, jonka pituus on $\geq k+1$; tällä polulla on siis jonkin välikkeen toistuttava — itse asiassa jo polun $k+2$ alimman solmun joukossa. Olkoon A jokin tällainen välike.



Merkkijono z voidaan nyt osittaa $z = uvwxy$, missä w on A:n alimmasta ilmentymästä tuotettu osajono ja vwx seuraavaksi ylemmästä A:n ilmentymästä tuotettu osajono; osajonot saadaan johdosta

$$S \Rightarrow^* uAy \Rightarrow^* uvAxy \Rightarrow^* uvwxy.$$



Koska siis $S \Rightarrow^* uAy$, $A \Rightarrow^* vAx$ ja $A \Rightarrow^* w$, osajonoja v ja x voidaan "pumpata" w:n ympärillä:

$$S \Rightarrow^* uAy \Rightarrow^* uvAxy \Rightarrow^* uv^2Ax^2y \Rightarrow^* \ldots \Rightarrow^* uv^iAx^iy \Rightarrow^* uv^iwx^iy.$$

Siten $uv^iwx^iy \in L$ kaikilla $i = 0, 1, 2, \ldots$.

Koska kielioppi G on Chomskyn normaalimuodossa ja $A \Rightarrow^* vAx$, on oltava $|vx| \geq 1$.

Koska edelleen välikkeen A valinnan perusteella sen toiseksi ylin ilmentymä on enintään korkeudella $k+1$ jäsennyspuun lehdistä, on tähän ilmentymään juurtuvan alipuun tuotokselle voimassa pituusraja $|vwx| \leq 2^{k+1} = n$. $\square$

T–79.1001/1002 Tietojenkäsittelyteorian perusteet

Esimerkki. Tarkastellaan kieltä

$$L = \{a^k b^k c^k \mid k \geq 0\}.$$

Oletetaan, että L olisi yhteydetön; valitaan parametri n lemman mukaisesti ja tarkastellaan merkkijonoa $z = a^n b^n c^n \in L$.

Lemman 3.9 mukaan z voidaan jakaa pumpattavaksi osiin

$$z = uvwxy, \qquad |vx| \geq 1, \qquad |vwx| \leq n.$$

Viimeisen ehdon takia merkkijono vx ei voi sisältää sekä a:ta, b:tä että c:tä. Merkkijonossa $uv^0wx^0y = uwy$ on siten ylijäämä jotakin merkkiä muihin merkkeihin nähden, eikä se voi olla kielen L määritelmässä vaadittua muotoa, vaikka lemman mukaan pitäisi olla $uwy \in L$.

Pekka Orponen syksy 2005