

OHI-2300 Johdatus tietojenkäsittelyteoriaan Syksy 2008



Organisaatio & aikataulu

Luennot: prof. Tapio Elomaa

- P1: Ti 14-16 TC 103 ja to 14-16 TC 133
- P2: Ti 14-16 TB 219 ja to 12-14 TB 224
- 26.8.–20.11.
 - Jussi Kujalan väitös ke 24.9. klo 12 R G202
 - Torstaina 25.9. ei luentoa
 - perioditauko 6.–10.10.

Viikkoharjoitukset 8.9. →

- DI Jussi Kujala (DI Timo Aho), ma 12-14 TC 131

Koe: ke 26.11.2008 klo 9-12

Arvostelu: 30+8=38

{elomaa, kujala2}@cs.tut.fi

Viikkoharjoitukset

- Viikkoharjoituksiin osallistuminen on **erittäin** suositeltavaa
- Valmiudesta esittää taululla vastaus kysymykseen saa kustakin merkinnän
- Kussakin harjoituksissa on n. 6 tehtävää →
kaikkiaan merkintöjä voi kerätä n. $6 \times 10 = 60$ kappaletta

Merkintöjä	Lisäpisteet
20% (n. 12)	1
30% (n. 18)	2
40% (n. 24)	3
...	...
80% (n. 48)	7
90% (n. 54)	8

Arvostelu

- Kurssista voi selvittää parhaalla mahdollisella arvosanalla pelkällä tentillä, mutta
 - tentti (max 30 p.) ei ole aivan helppo
 - siksi kannattaa osallistua viikottaisiin laskuharjoituksiin, joista saa lisäpisteitä (max 8 p.) ja
 - joiden tehtäviä itsenäisesti ratkomalla oppii kurssin näennäisesti vaikeita asioita
- Arvosana määräytyy oletettavasti seuraavasti:

pisteet	15	18	21	24	27
arvosana	1	2	3	4	5

Materiaali

- Tällä toteutuskerralla oppikirja on

Michael Sipser: *Introduction to the Theory of Computation*,
Second Ed. (International Ed.), Thomson, 2006

- Melkein mikä tahansa aiheen oppikirja kattaa kurssilla käsitellyt asiat
- Valmista monistetta ei ole, luentokalvot pannaan verkkoon luentojen tahtiin
 - <http://www.cs.tut.fi/kurssit/OHJ-2300/>
 - <http://www.cs.tut.fi/~elomaa/opetus/OHJ-2300-08.html/>
- Tenti perustuu luentoihin (ei siis yksinomaan kalvoihin)

Luentoaikataulu

1. **Johdanto** (1)
2. **Kertausta: automaattit, kieliopit ja kielet** (2-3)
 - Säännölliset kielet
 - Kontekstiittomat kielet
3. **Laskettavuusteoria** (4-7)
 - Laskennan malleja
 - Ratkeavuus
 - Palautukset
 - Edistyneitä aiheita
4. **Laskennan vaativuusteoria** (8-12)
 - Aikavaativuus
 - Tilavaativuus
 - Käytännöllinen ratkeavuus
 - Edistyneitä aiheita

1. Johdanto

- Tällä kurssilla annetaan johdatus tietojenkäsittelyn matemaattis-teoreettiseen perustaan
- Tietojenkäsittelijän yleissivistykseen kuuluvaa tietoa
- Tavoitteena on saada perusymmärtämys siitä minkä tyyppisiä ongelmia tietokoneella voi periaatteessa ratkaista
- Vielä tärkeämpää on havaita mille ratkeavista ongelmista voi ohjelmoida tehokkaan ratkaisun
- Kurssilla tarkastellut tulokset ovat fundamentaalisia, tulevien vuosien tehonlisäykset tietokoneissa eivät tule poistamaan näiden tulosten merkitystä



1.1 Laskennalliset ongelmat

- Laskennallinen ongelma voidaan mallintaa tietokoneella ratkaistavaksi;
 - esim. aritmetiikka, aakkostaminen, palkanlaskenta, kurssikirjanpito, ...
- Ongelman ratkaisevaa ohjelmaa yleisempi esitys usein ymmärrettävämpi ja mahdollistaa esim. analysoinnin
- Ongelmalla on *tapauksia* (= syötteitä), ratkaisu on *algoritmi*, joka liittää kuhunkin tapaukseen *vastauksen* (= tulosteen)
- Tapaus ja sen vastaus ovat *äärellisesti* (esim. bittijonoin) *esitettäviä*. Tapauksia voi olla ääretön määrä.
- Laskennallinen ongelma on *kuvaus äärellisesti esitettävien tapauksien joukosta äärellisesti esitettävien vastausten joukkoon*.



Esitykset

- Äärellinen esitys = aakkoston merkkijono
 - Aakkosto on äärellinen, epätyhjä joukko *symboleita*
 - Esim. *binääriaakkosto* $\{0, 1\}$ ja latinalainen aakkosto $\{A, B, \dots, Z\}$
- *Merkkijono* on äärellinen järjestetty jono aakkoston merkkejä
 - Esim. 00010 ja TWILIGHT
- Merkkijonon x *pituus* $|x|$ on siihen sisältyvien merkkien lukumäärä
 - Esim. $|00010| = 5$ ja $|TWILIGHT| = 8$
- *Tyhjän merkkijonon* ϵ *pituus* $|\epsilon| = 0$



- Merkkijonojen yhteenliittäminen eli *katenaatio* on niiden perusoperaatio.
 - $TWILIGHT \circ ZONE = TWILIGHTZONE$
 - $x = 01, y = 10 \rightarrow xx = 0101, xy = 0110, yy = 1010$ ja $yx = 1001$
 - Kaikilla x on $x\epsilon = \epsilon x = x$
 - Kaikilla x, y on $|xy| = |x| + |y|$
- Jos $w = xy$, niin x on w :n *alkuosa* (prefix) ja y on sen *loppuosa* (suffix)
- Aakkoston Σ kaikkien merkkijonojen joukko on Σ^*
 - Esim. $\Sigma = \{0, 1\} \rightarrow \Sigma^* = \{\epsilon, 0, 1, 00, 01, 10, 11, \dots\}$
- Muita merkintöjä: $\Sigma^k, \Sigma^{\leq k}$ ja $\Sigma^* = \bigcup_{k \in \mathbb{N}} \Sigma^k$



Päätösongelmat

- Laskennallinen ongelma π on siis kuvaus

$$\pi: \Sigma^* \rightarrow \Gamma^*$$
 missä Σ ja Γ ovat aakkostoja
- Päätösongelmat** ovat tärkeä aliluokka; niissä ongelman tapauksen vastaus on kyllä tai ei
 - Ne ovat siis muotoa $\pi: \Sigma^* \rightarrow \{0, 1\}$
- Jokaista päätösongelmaa π vastaa niiden tapausten joukko, joilla vastaus on kyllä:

$$A_\pi = \{x \in \Sigma^* \mid \pi(x) = 1\}$$

- Kääntäen jokaista merkkijonojoukkoa A vastaa päätösongelma $\pi_A: \Sigma^* \rightarrow \{0, 1\}$,

$$\pi_A(x) = \begin{cases} 1, & \text{jos } x \in A \\ 0, & \text{jos } x \notin A \end{cases}$$

- Merkkijonojoukkoa $A \subseteq \Sigma^*$ sanotaan aakkoston Σ (formaaliksi) kieleksi
- Vastaavaa päätösongelmaa π_A nimitetään kielen A *tunnistusongelmaksi*
- Formaalit kielet ja päätösongelmat voidaan samaistaa

Laskennallisten ongelmien ratkeavuus

- Sanotaan, että ohjelma $P(x)$ ratkaisee laskennallisen ongelman π , jos kaikilla syötteillä x ohjelma P tulostaa arvon $\pi(x)$
- Voidaanko kaikki mahdolliset laskennalliset ongelmat ratkaista ohjelmin?
- Ääretön joukko X on *numeroituva*, jos on olemassa bijektio $f: \mathbb{N} \rightarrow X$.
- Myös äärelliset joukot ovat numeroituvia
 - Bijektio = injektio + surjektio
 - Injektio: $f(x_1) = f(x_2) \Rightarrow x_1 = x_2$
 - Surjektio: $f(A) = B$
- Ääretön joukko, joka ei ole numeroituva on *ylinnumeroituva*

Lause 1.1 Minkä tahansa aakkoston Σ merkkijonojen joukko Σ^* on numeroituva.

Todistus. Olkoon $\Sigma = \{a_1, a_2, \dots, a_n\}$. Kiinnitetään merkeille "aakkosjärjestys"; olkoon se $a_1 < a_2 < \dots < a_n$.

Joukon Σ^* merkkijonot voidaan nyt luetella *kanonisessa järjestyksessä*:

1. Ensin luetellaan 0:n mittaiset merkkijonot, sitten 1:n mittaiset, 2:n mittaiset jne.
2. Kunkin pituusryhmän sisällä merkkijonot luetellaan valitussa aakkosjärjestyksessä

Tällöin bijektio $f: \mathbb{N} \rightarrow \Sigma^*$ on

$$\begin{aligned} 0 &\mapsto \varepsilon \\ 1 &\mapsto a_1 \\ 2 &\mapsto a_2 \\ &\vdots \\ n &\mapsto a_n \\ n+1 &\mapsto a_1a_1 \\ n+2 &\mapsto a_1a_2 \\ &\vdots \end{aligned}$$

$$\begin{aligned} 2n &\mapsto a_1a_n \\ 2n+1 &\mapsto a_2a_1 \\ &\vdots \\ 3n &\mapsto a_2a_n \\ &\vdots \\ n^2+n &\mapsto a_na_n \\ n^2+n+1 &\mapsto a_1a_1a_1 \\ n^2+n+2 &\mapsto a_1a_1a_2 \\ &\vdots \end{aligned}$$

□

Lause 1.2 Minkä tahansa aakkoston Σ päätösongelmien joukko on ylinumeroituva.

Todistus. Merkitään Π :llä kaikkien Σ :n päätösongelmien kokoelmaa: $\Pi = \{ \pi \mid \pi \text{ on kuvaus } \Sigma^* \rightarrow \{0, 1\} \}$.

Oletetaan, että Π on numeroituva, so. on olemassa kaikki Π :n alkiot kattava numerointi $\Pi = \{ \pi_0, \pi_1, \pi_2, \dots \}$.

Olkoon Σ^* :n merkkijonot Lauseen 1.1 todistuksen kanonisessa järjestyksessä lueteltuina x_0, x_1, x_2, \dots

Muodostetaan uusi päätösongelma $\xi: \Sigma^* \rightarrow \{0, 1\}$:

$$\xi(x_i) = \begin{cases} 1, & \text{jos } \pi_i(x_i) = 0 \\ 0, & \text{jos } \pi_i(x_i) = 1 \end{cases}$$

Koska Π on kaikki Σ :n päätösongelmat kattava numerointi, niin $\xi \in \Pi$. Täten $\xi = \pi_k$ jollakin $k \in \mathbb{N}$. Tällöin

$$\xi(x_k) = \begin{cases} 1, & \text{jos } \pi_k(x_k) = \xi(x_k) = 0 \\ 0, & \text{jos } \pi_k(x_k) = \xi(x_k) = 1 \end{cases}$$

Tämä on ristiriita, joten tehty oletus (Π on numeroituva) ei voi pitää paikkaansa. Näin ollen Π on ylinumeroituva. \square



- Todistustekniikka on ns. **Cantorin diagonaaliargumentti**.
- Koska esim. Java-ohjelmat ovat loppujen lopuksi vain ASCII-merkistön merkkijonoja, niin niitä on Lauseen 1.1 mukaan vain numeroituva joukko
- Laskennallisten ongelmien joukko on kuitenkin Lauseen 1.2 perusteella ylinumeroituva
- Näin ollen kaikista laskennallisista ongelmista voidaan Java-ohjelmin ratkaista vain häviävän pieni osa
- Sama ongelma on millä tahansa ohjelmointikielellä
- Ratkeamattomat ongelmat käsittävät myös mielenkiintoisia / käytännöllisiä ongelmia



Georg Cantor (1845-1918)

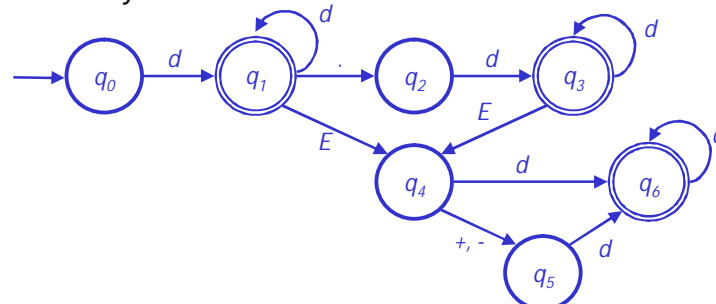
- Synt. Pietari
- Frankfurtiin 1856
- Tri (Berliini) 1867
- Halle 1869-
 - Privatdozent
 - Prof. 1872
- Rationaalilukujen numeroituvuus 1873
- Joukko-opin teoria 1874
- Kontinuumihypot. 1878



2. Kertausta

2.1 Äärelliset automaattit

- Laskentajärjestelmän, jolla on vain äärellisen monta mahdollista tilaa, toiminta voidaan kuvata **äärellisenä automaattina**
- Äärellinen automaatti havainnollistetaan usein **tilasiirtymäkaaviona**



- Äärellinen automaatti on viisikko $M = (Q, \Sigma, \delta, q_0, F)$, missä
 - Q on automaatin *tilojen* joukko,
 - Σ on automaatin *syöteaakkosto*,
 - $\delta: Q \times \Sigma \rightarrow Q$ on *siirtymäfunktio*,
 - $q_0 \in Q$ on *alkutila* ja
 - $F \subseteq Q$ (hyväksyvien) *lopputilojen* joukko.
- Automaatti M *hyväksyy* merkkijonon $w = w_1w_2\dots w_n \in \Sigma^n$ jos Q :ssa on tilojen jono r_0, r_1, \dots, r_n s.e.
 - $r_0 = q_0$
 - $\delta(r_i, w_{i+1}) = r_{i+1}, i = 0, \dots, n-1,$
 - $r_n \in F.$

- Automaatin M *tunnistama kieli* on

$$L(M) = \{ w \in \Sigma^* \mid M \text{ hyväksyy } w:n \}$$
- Kieli on *säännöllinen*, jos jokin äärellinen automaatti tunnistaa sen
- Kielten perusoperaatiot ovat
 - *yhdiste* $A \cup B = \{ x \mid x \in A \vee x \in B \},$
 - *katenaatio* $A \circ B = \{ xy \mid x \in A \wedge y \in B \}$ ja
 - *sulkeuma* $A^* = \{ x_1x_2\dots x_k \mid k \geq 0 \wedge x_i \in A \forall i \}$



Lause 1.25 *Säännöllisten kielten luokka on suljettu yhdisteen suhteen. Toisin sanoen, jos A_1 ja A_2 ovat säännöllisiä kieliä, niin silloin myös $A_1 \cup A_2$ on säännöllinen.*

Lause 1.26 *Säännöllisten kielten luokka on suljettu katenaation suhteen.*