

# Az informatika számítástudományi alapjai

## 1. előadás

Vaszil György

[vaszil.gyorgy@inf.unideb.hu](mailto:vaszil.gyorgy@inf.unideb.hu)

I. emelet 110-es szoba

# Miről szól ez az egész?

A számítógépek képességeiről és a képességek **elvi** korlátairól.

- Elméleti számítástudomány
  - **Automata**: matematikai “gép”
    - a **számítógép hardver** elvi, matematikai leírása
    - a **számítási folyamat** elvi, matematikai leírása
  - Formális **nyelv**: jelsorozatok/sztringek/”szavak” sokasága/halmaza
    - az automaták **számítási képességeinek** jellemzése
    - a számítási **feladatok** formális/elvi/matematikai leírása
    - **Bár!:** *(programozási nyelvek leírása, compilerek)*

# Miről szól ez az egész?

A számítógépek képességeiről és a képeségek *elvi* korlátairól.

A számítógépek modellezése, a **számítási folyamat “hardverfüggetlen”**) leírása (automaták) azért érdekes:

- A számítógéppel *elvileg is* megoldhatatlan feladatokról (kiszámíthatóság)
  - Mikor megoldható egy feladat?
- A számítógéppel elvileg megoldható feladatok **gyakorlati megoldhatósága** (számítási bonyolultság)
  - Mitől **egyszerű** és mitől **nehéz** egy feladat megoldása?

# Theory of Computation

## Computability

What can we compute?

- Most general notions of computability
- Uncomputable functions

## Complexity

What can we compute fast? CS473!

- Faster algorithms, polynomial time
- Problems that cannot be solved fast:
  - \* Cryptography

## Automata

What can we compute with very little space?

- Constant space (+stack)
  - \* String searching, language parsing, hardware verification, etc.

# Theory of Computation

I  
N  
C  
R  
E  
A  
S  
I  
N  
G  
  
C  
O  
M  
P  
L  
E  
X  
I  
T  
Y



Turing machines

Context-free  
languages

Automata

Automata:

- Foundations of computing
- Mathematical methods of argument
- Simple setting

# Theory of Computation

I  
N  
C  
R  
E  
A  
S  
I  
N  
G  
  
C  
O  
M  
P  
L  
E  
X  
I  
T  
Y



Turing machines

Context-free  
languages

Automata

Context-free languages

--- Grammars, parsing

--- Finite state machines with recursion (or stack)

--- Still a simple setting; but infinite state

# Theory of Computation

I  
N  
C  
R  
E  
A  
S  
I  
N  
G  
  
C  
O  
M  
P  
L  
E  
X  
I  
T  
Y

Turing machines

Context-free  
languages

Automata

Turing machines (1940s):

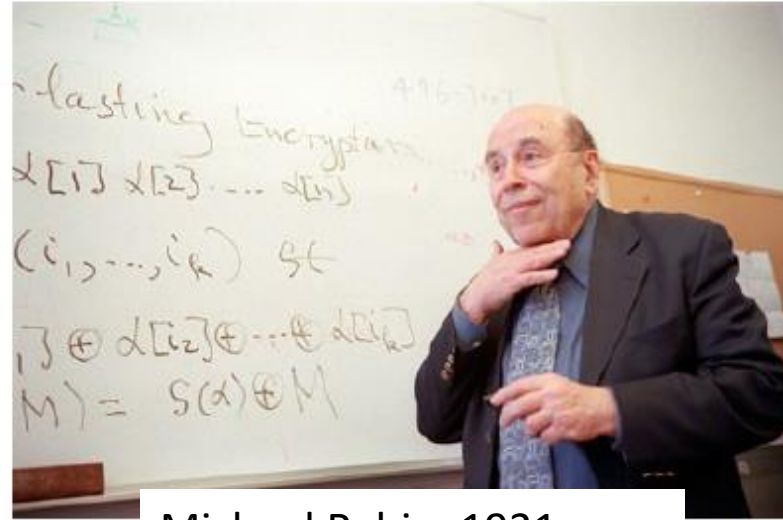
- The most general notion of computing
- The Church-Turing thesis
- Limits to computing:  
Uncomputable functions

Motivation from mathematics:

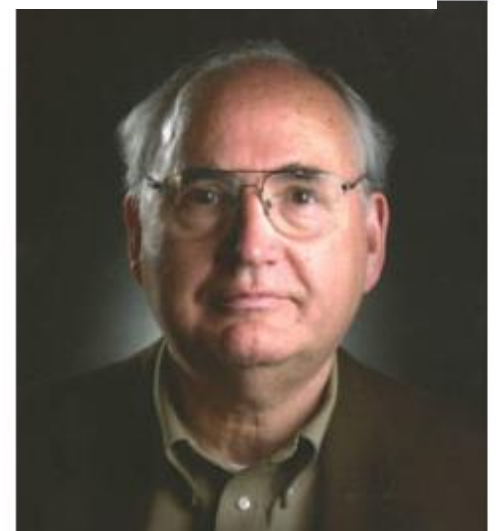
- Can we solve any mathematical question *methodically*?
- Godel's theorem: NO!
- "Even the most powerful machines cannot solve some problems."

# Finite automata

- *Automata: machines with finite memory*
- *“Finite Automata and Their Decision Problem”*  
- Rabin and Scott (1959)
- *Introduced nondeterministic automata and the formalism we still use today*
- *Initial motivation: modeling circuits*
- *Turing Award (1976)*



Michael Rabin, 1931-

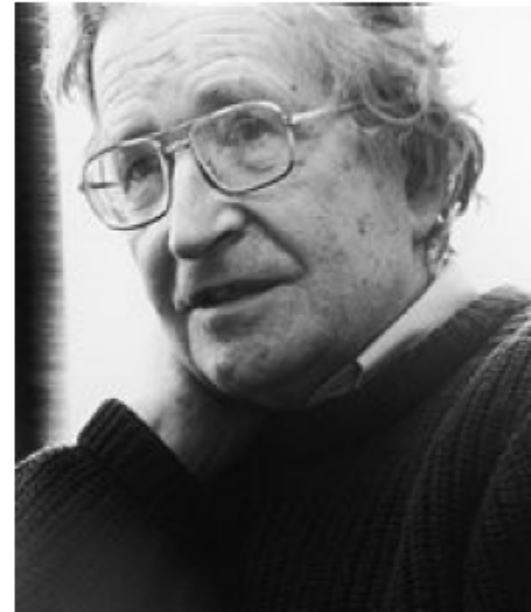


Dana Scott, 1932-



# Noam Chomsky

- Linguist ; introduced the notion of formal languages arguing generative grammars are at the base of natural languages
- Hierarchy of formal languages that coincides with computation
- Eg. Context-free grammars capture most skeletons of prog. languages



**Noam Chomsky: 1928-**

**“Logical Structure of Linguistic Theory” (1957)**

# Alan Turing

- “father of computer science”
- Defined the first formal notion of a computer (Turing machine) in 1936:  
“*On Computable Numbers, with an Application to the Entscheidungsproblem*”
- Proved uncomputable functions exist (“halting problem”)
- Church-Turing thesis: all real world computable functions are Turing m/c computable
- Cryptanalysis work breaking Enigma in WW-II



**Alan Turing: 1912 - 1954**

# Alonzo Church

First notions of computable functions

First language for programs

- lambda calculus
- formal algebraic language  
for computable functions



**Alonzo Church:**  
**1903 - 1995**

# Kurt Gödel

- Logician extraordinaire
- Hilbert, Russell, etc. tried to formalize mathematics
- “Incompleteness theorem” (1931)
  - Cannot prove consistency of arithmetic formally
  - Consequence: unprovable theorems



**Kurt Gödel: 1906 - 1978**

# Miért lehet hasznos egy ilyen “elméleti” tárgy? Például...

- Speciális célú (domain specific) programnyelvek készítése → formális grammatikák
- Szövegminta keresés → reguláris kifejezések, véges automaták
- Nehéznek tűnő feladatok nehézségének belátása  
-> számítási bonyolultság, NP teljesség
- Kriptográfia -> számítási bonyolultság

# Talán a legfontosabb azonban...

... hogy a technológia gyorsan változik, a “technológia-függő” tudás gyorsan elavul.

Hogy lépést tudjunk tartani, olyan készségekre van szükségünk, amit az “elméleti” tárgyak nyújtanak:

- precíz gondolkodás
- problémamegoldó képesség
- jártasság a konkrét megoldások mögötti általános elvek területén

# Kitérő a hét szabad művészetről

**Mi a számítástudomány?**

Copyright by David Evans 2004

# What is Computer Science?



# Geometry vs. Computer Science

- Geometry (mathematics) is about *declarative* knowledge: “what is”

If now  $CD$  measures  $AB$ , since it also measures itself, then  $CD$  *is* a common measure of  $CD$  and  $AB$

- ~~Computer Science~~ is about *imperative* knowledge: “how to”

Computer Science has **nothing** to do with beige (or translucent blue) boxes called “computers” and is not a science.

# Science, Engineering, Other?

# Science?

- Understanding Nature through Observation
  - About *real* things like bowling balls, black holes, antimatter, electrons, comets, etc.
- Math and Computer Science are about *fake* things like numbers, graphs, functions, lists, etc.
  - Computer Science is a useful tool for *doing* real science, but not a real science

# Engineering?

“Engineering is **design under constraint**... Engineering is synthetic - it strives to create what can be, but it is constrained by nature, by cost, by concerns of safety, reliability, environmental impact, manufacturability, maintainability and many other such 'ilities.' ...”

William Wulf

# Constraints Computer Scientists Face

- Not like those for engineers:
  - Cost, weight, physics, etc.
  - If ~8 Million times what people had in 1969 isn't enough for you, wait until 2006 and you will have 20 Million times...
- More like those for Musicians and Poets:
  - Imagination and Creativity
  - Complexity of what we can understand

# So, what is computer science?

- ~~Science~~

- No: its about fake things like numbers, not about observing and understanding nature

- ~~Engineering~~

- No: we don't have to deal with engineering-type constraints

- Liberal Art

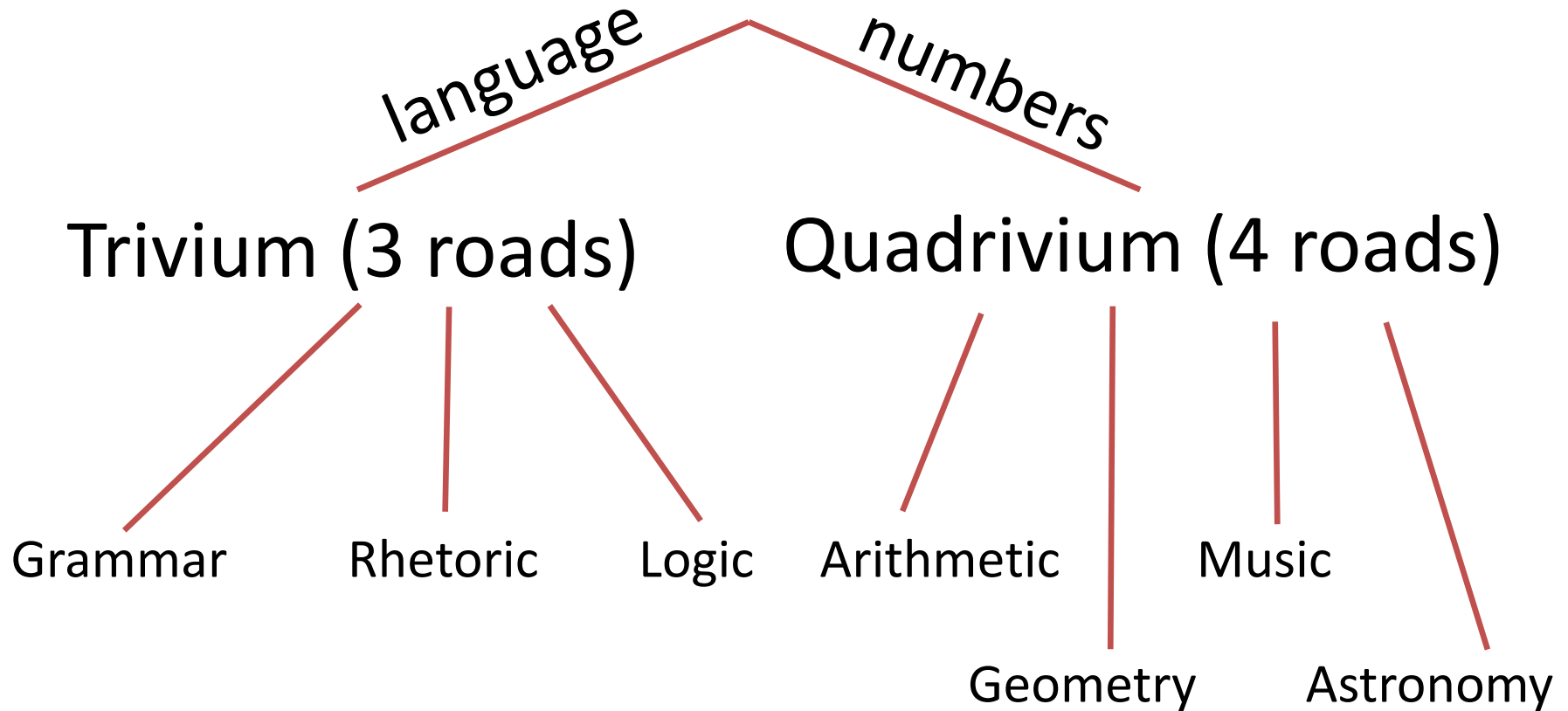
# Liberal Arts: ~1100

- Illiberal Arts
  - arts for the non-free: pursued for economic reasons
- Liberal Arts
  - arts for the *free* (*arts that lead you to „freedom“*): pursued for intrinsic reasons

I make bold to say that I never have despised anything belonging to erudition, but have learned much which to others seemed to be trifling and foolish. ... Some things are worth knowing on their own account; but others, although apparently offering no return for our trouble, should not be neglected, because without them the former cannot be thoroughly mastered. ...

*Didascalium*, Hugo of St. Victor, around 1120

# The Liberal Arts





# Liberal Arts

Trivium

- Grammar: study of meaning in written expression
- Rhetoric: comprehension of verbal and written discourse
- Logic: argumentative discourse for discovering truth

Yes, we need to understand meaning to describe computations

Interfaces between components, discourse between programs and users

Logic for controlling and reasoning about computations

Quadrivium

- Arithmetic: understanding numbers
- Geometry: quantification of space
- Music: number in time
- Astronomy: laws of the planets and stars

Yes (PS 6)

Yes (PS 1, 2, 3)

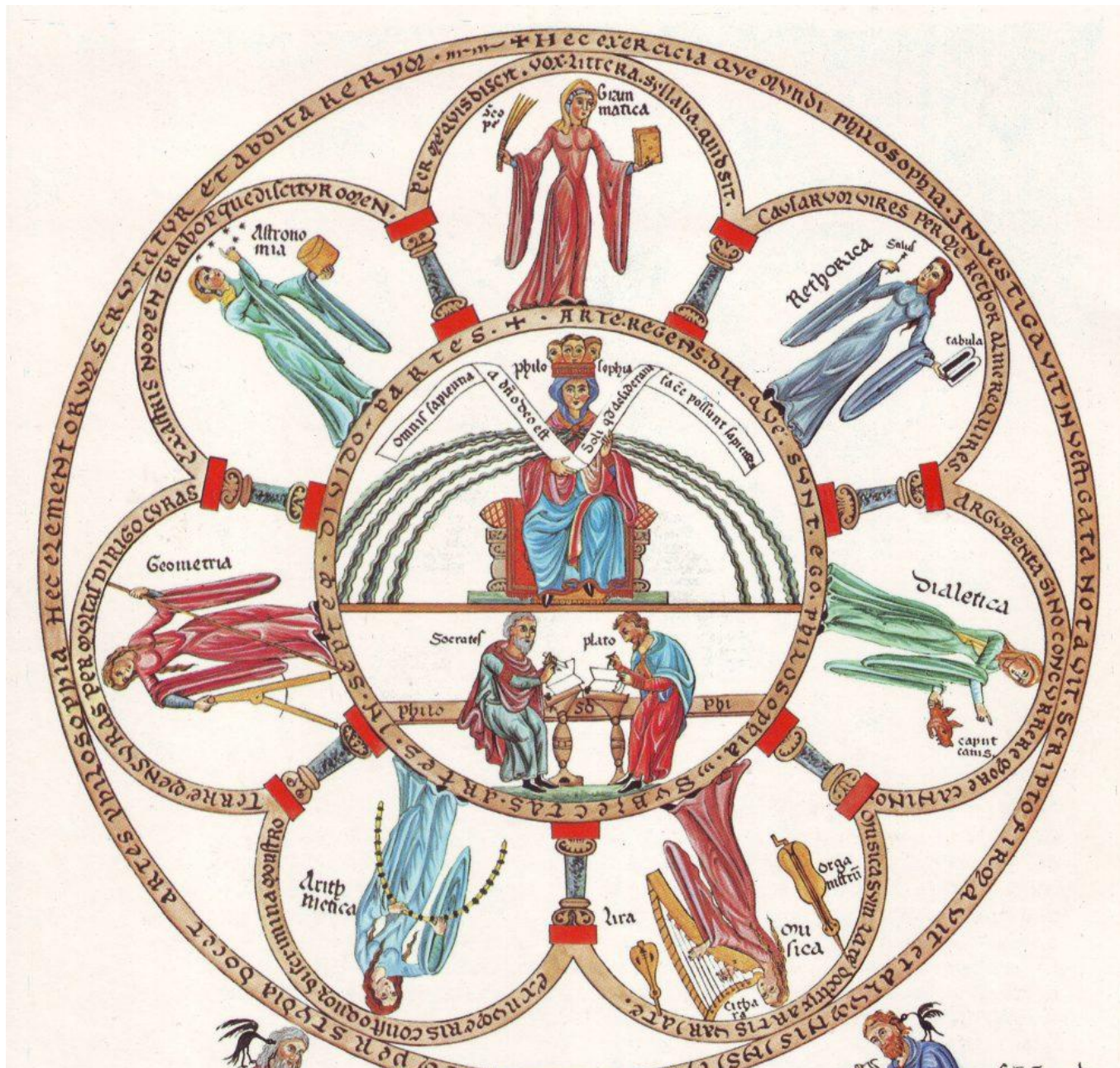
Yes, its called **GEB** for a reason!

Yes (we'll read Neil DeGrasse Tyson's essay)

## Bold (Possibly Untrue) Claim

CS200 is the ***most*** consistent with the original intent of a Liberal Arts education of any course offered at UVa this semester!

# Hét szabad művészet



# Trivium és quadrivium

## Content: The Trivium Arts

That which is sayable or that can be expressed with language.



Structure of  
Language

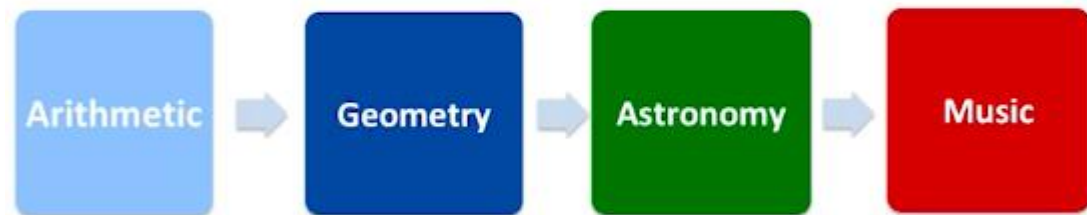
Art of Correct  
Reasoning

Art of Persuasive  
Speech & Writing

Az értelmet a quadrivium világítja meg, kifejezésére pedig a trivium szolgál.

## Content: The Quadrivium Arts

Learnable Objects: That which can be observed, quantified, measured, enabling a mathematical description of nature.



The unit with no  
dimensions

One-dimensional  
lines, two-dimensional  
surfaces, three-  
dimensional solids

Geometrical solids  
in motion

Audible progressions  
of numerical  
ratios

# Miért érdekes a számítástudomány?

- Egyrészt: Hasonlít a tradicionálisan „szabad művészetekként” ismert tevékenységekre (azokat az ember önmagában és önmagáért, „ember mivoltának beteljesítése” érdekében folytatja)
- Másrészt: Praktikus is („szervilis művészetek”)
  - Jártasság a konkrét megoldások mögötti általános elvek területén

1. hét Ábécék, szavak, nyelvek; műveletek szavakkal és nyelvekkel: halmazműveletek (unió, metszet, különbség, komplementer), konkatenáció, konkatenáció lezárása, determinisztikus véges automaták, elfogadott nyelv.
2. hét (Determinisztikus) véges automatákkal; elfogadott nyelvek uniójának, metszetének elfogadása; pumpálási lemma. Myhill-Nerode tétele; minimalizálás.
3. hét Nemdeterminisztikus véges automaták; determinisztikussá alakítás.
4. hét Reguláris kifejezések és véges automaták (Kleene tétele).
5. hét Generatív grammatikák, generált nyelvek; környezetfüggetlen nyelvek, egyértelműség.
6. hét Első ZH.
7. hét Normálformák (törlő szabályok kiküszöbölése, Chomsky normálforma); reguláris grammatikák, szintaktikai elemzés,  $LL(k)$  nyelvek.
8. hét Veremautomaták, determinisztikus környezetfüggetlen nyelvek,  $LR(k)$  nyelvek.
9. hét Környezetfüggetlen nyelvek, pumpálási lemma.
10. hét Turing gépek, nyelvek elfogadása, függvények kiszámítása; többszalagos Turing gépek, nemdeterminisztikus Turing gépek; Church tézis.
11. hét Univerzális Turing gép; rekurzív és rekurzívan felsorolható nyelvek, rekurzívan felsorolható nyelvek felsorolása.
12. hét Általános grammatikák, környezetfüggő grammatikák, a Chomsky hierarchia; nem rekurzívan felsorolható nyelvek létezése; megállási probléma, más eldönthetetlen problémák.
13. hét Második ZH.



# Tehát: Irodalom

- Dömösi P., Falucskai J., Horváth G., Mecsei Z. és Nagy B.: *Formális nyelvek és automaták*, jegyzet 2011.

[http://www.tankonyvtar.hu/en/tartalom/tamop425/  
0046\\_formalis\\_nyelvek\\_es\\_automata/0046\\_formalis\\_nyelvek\\_es\\_automata.pdf](http://www.tankonyvtar.hu/en/tartalom/tamop425/0046_formalis_nyelvek_es_automata/0046_formalis_nyelvek_es_automata.pdf)

- Herendi T.: *Számításelmélet*, jegyzet 2014.

[http://www.tankonyvtar.hu/hu/tartalom/tamop412A/  
2011-0103\\_25\\_szamitaselmelet/2011-0103\\_25\\_szamitaselmelet.pdf](http://www.tankonyvtar.hu/hu/tartalom/tamop412A/2011-0103_25_szamitaselmelet/2011-0103_25_szamitaselmelet.pdf)

- J. Martin: *Introduction to Languages, and the Theory of Computation*, 4th edition, McGraw-Hill, New York, NY, 2011.
- M. Sipser *Introduction to the Theory of Computation*, Thomson, 2006.

FájlSzerkesztésNézetElőzményekKönyvtárEszközökSúgó

<Neo-A Neo-Librarlibrary.lolThe Demolibrar(eboNeoNeo-AristA (99)G KerékKari T NumbLaTeXDebreKurXWSL - editorBing>+v

←→https://elearning.unideb.hu/course/view.php?id=1970590%☆↻🏠🔔🗉📄📊📌📁☰

☰elearning.unideb.hu🔗KurzusaimE-learning szolgáltatásokmagyar (hu)🔍🔔💬Dr. Vaszil GyörgyHallgató👤

🎓Az informatika számítástudományi alapjai (INBPM0314E/21o)

👤Résztevők

🛡️Kitűzők

📅Osztályzatok

📁Általános

📁1. előadás - Bevezetés: „motiváció”, alapfogalmak

📁Irányítópult

🏠Portál kezdőoldala

📅Naptár

📄Saját állományaim

🎓Kurzusaim

🎓Adatszerkezetek és algoritmusok (INBMM0207V/21o)

🎓Az informatika logikai alapjai (INBPM0101E/21o)

🎓Az informatika logikai alapjai (INBPM0101G-01/21o)

Az informatika számítástudományi alapjai (INBPM0314E) 2021. ősz

Irányítópult / Kurzusaim / Informatikai Kar / Számítógéptudományi Tanszék / Az informatika számítástudományi alapjai (INBPM0314E/21o)

1. előadás - Bevezetés: „motiváció”, alapfogalmak

Alapfogalmak: halmazok, függvények, relációk, ábécék, szavak, nyelvek. Nyelvek megadása: műveletek, jelölések.

📄Feladatsor

Bemelegítő feladatsor az első előadás előtti hétfői nap gyakorlataira

📄Kurzusblokkok

logo-footer.png

Debreceni Egyetem  
https://elearning.unideb.hu

Tölts le a Moodle-t a mobiljára

Kapcsolat

🌐https://metk.unideb.hu

✉elearning@metk.unideb.hu



## Követel mély

- Gyakorlati aláírás :
  - részvétel (3 alkalommal lehet kiágyani)
  - két  $2H$ , a félév töcepein és végén
    - mindkét  $2H$ -n teljesíteni kell a minimum követelményeket
    - a félév végén a  $2H$  javítható
- Azonra írásbeli len.

Kérdések???

# Hogyan fogjunk hozzá?

## Az első fólia/dia még egyszer:

- Elméleti számítástudomány
  - **Automata**: matematikai “gép”
    - a **számítógép hardver** elvi, matematikai leírása
    - a **számítási folyamat** elvi, matematikai leírása
  - Formális **nyelv**: jelsorozatok/sztringek/”szavak” sokasága/halmaza
    - az automaták **számítási képességeinek** jellemzése
    - a számítási **feladatok** formális/elvi/matematikai leírása
    - *(programozási nyelvek leírása, compilerek)*

# A mai órán

- Alapfogalmak: halmazok, függvények, relációk, ábécék, szavak, nyelvek
- Nyelvek megadása: műveletek, jelölések

## Alapfogalmak

- Halmaz

pl:  $\{7, 21, 5\}$

- eleme, nem eleme, részhalmaz, ~~halmaz~~  $\emptyset$

- Egy halmaz lehet végtelen is:

pl.  $\mathbb{N}$  - a természetes számok halmaza  
 $\{0, 1, 2, 3, \dots\}$

- unió, metszet, komplement

**1.8.** In each case below, find an expression for the indicated set, involving  $A$ ,  $B$ ,  $C$ , and any of the operations  $\cup$ ,  $\cap$ ,  $-$ , and  $'$ .

- a.  $\{x|x \in A \text{ or } x \in B \text{ but not both}\}$
- b.  $\{x|x \text{ is an element of exactly one of the three sets } A, B, \text{ and } C\}$
- c.  $\{x|x \text{ is an element of at most one of the three sets } A, B, \text{ and } C\}$
- d.  $\{x|x \text{ is an element of exactly two of the three sets } A, B, \text{ and } C\}$

A vesszőzés a komplementerképzést jelenti, azaz  $A'$  azoknak a dolgoknak a halmazát jelöli, amik nincsenek benne  $A$ -ban.

**1.12.** a. How many elements are there in the set

$$\{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}, \{\emptyset, \{\{\emptyset, \{\emptyset\}\}, \{\emptyset, \{\emptyset\}\}\}\}?\}$$

Relat'ic', ~~plu~~: n - sekret az {iga, havis}  
halmarha repero' pizgmaing.

Rel: k'o' - papir - allo

$L \in \text{Gv}^2 i : A \times A \mapsto \{iga, havis\}$   
(...)

$A = \{ \begin{matrix} \text{v'o'} \\ \text{papier} \\ \text{allo} \end{matrix} \}$

$L \in \text{Gv}^2 i (x, y) = iga, ha \quad x \text{ lepp}^2 i \quad y - t.$

pl.  $L \in \text{Gv}^2 i (v'o', \text{papier}) = havis$

$L \in \text{Gv}^2 i (\text{papier}, v'o') = iga$

Binäre relationen (charakter)  
"infix" modus.

Bsp: " $<$ " - ~~is~~ ist

$3 < 4$  und  $<(3, 4) = \text{ig}$

$< : \mathbb{N} \times \mathbb{N} \rightarrow \{\text{ig}, \text{keis}\}$

$<(x, y) = \text{ig}$ , da  $x$  ~~is~~ ist  
mit  $y$ .

A'he'ce', nauat, uzelner

A'he'ce' : siimhe'lnet (hehi'?) u'gr haluara

Adott a'he'ce' pletti ne' : an adotti a'he'ce' hehi'iböl  
alotett u'gr rororat

üner ne' : an eggelen hehi't zem farthaluare'  
rororat, jile 2 (lambda, "leer"  
- üner)

Se' hogga : a nit alotot' rororat elemeiner  
na'ua



## Példák

•  $V_1 = \{0, 1\}$  azaz kettő szám az, 011101,  $\lambda$

• hossz :  $|00| = 2$ ,  $|011101| = 6$ ,  $|\lambda| = 0$

azaz

•  $V_2 = \{\text{if, then, else, } a, b, c\}$  kettő szó az

$w = \text{if } a \text{ then } b \text{ else } c$  , azaz

$|w| = 6$

azaz

•  $V_3 = \{1, 2, +\}$ , azaz  $1+1 \neq 2$

## Nyelvr

A dett áhe'ce' kletti gramár u'gét uen  
éppér u'gétellé halmasa - u'gét.

- $V^*$ : az ésser  $V$  áhe'ce' kletti né halmasa

$$\text{pl.: } \{0,1\}^* = \{\lambda, 0, 1, 00, 01, 10, 11, 000, \dots\}$$

- $V^+$ : az ésser uén ürr né halmasa

- $L_{\text{u'gét}}$ :  $L \subseteq V^*$  ( $V$  áhe'ce' kletti u'gét)  
Vbetüchöl képrett namar halmasa

# Beispiel grammatik

$$V_1 = \{0, 1\} \text{ klett}$$

- $\emptyset$
- $\{\lambda\}$
- $V^*$
- $L_1 = \{00, 101, 010\}$
- $L_2 = \{ \underbrace{0 \dots 0}_n \underbrace{1 \dots 1}_n \mid n \geq 1 \}$
- $L_3$  :
  - a)  $\lambda \in L_3$
  - b)  $\text{wenn } w \in L_3, \text{ dann } 0w1, 1w0 \in L_3$
  - c)  $u \in L_3 \text{ \& } v \in L_3, \text{ dann } uv \in L_3$
  - d)  $L_3$  - hier nicht mehr, weil mit a-c nachfolgend nicht mehr zusammenhängend

## Konkatenaicid

•  $u = a_1 a_2 \dots a_n$ ,  $v = b_1 \dots b_m$ ,  $u, v \in V^*$  Gët nö

~~$u \cdot v = a_1 a_2 \dots a_n b_1 b_2 \dots b_m$~~

A Gët nö,  $u \cdot v$  ~~notat~~ Konkatenaicidjari a  
 $u \cdot v = a_1 a_2 \dots a_n b_1 b_2 \dots b_m$  nöf sitjii?

Pl.  $V = \{a, b\}$

$$abba \cdot baba = abbababa$$

illene

$$baba = \{ba\} \cdot ba = (ba)^2$$

illote

$$\lambda \cdot u = u \cdot \lambda = u \text{ herwidze } u \text{ nöra}$$

## Nyelvisműveletek

- "Nyelvisműveletek": ha  $L_1$  és  $L_2$   $V$  ábéli nyelv

$$L_1 \cup L_2 = \{w \mid w \in L_1 \text{ vagy } w \in L_2\} - \text{unió}$$

$$L_1 \cap L_2 = \{w \mid w \in L_1 \text{ és } w \in L_2\} - \text{metszet}$$

$$L_1 - L_2 = \{w \mid w \in L_1 \text{ és } w \notin L_2\} - \text{különbözet}$$

$$\bar{L}_1 = \overline{L_1} = V^* - L_1 - \text{komplement}$$

- Konkaténáció:

$$L_1 \cdot L_2 = \{u \cdot v \mid u \in L_1, v \in L_2\}$$

$$L^0 = \{\lambda\}, L^1 = L, L^2 = L \cdot L, L^n = L^{n-1} \cdot L$$

$$L^* = \bigcup_{n \geq 0} L^n = L^0 \cup L^1 \cup L^2 \cup \dots$$

## Bildung

- $L_1 = \{0, 01\}$  ,  $L_2 = \{1, 11\}$

$$L_1 \cdot L_2 = \{01, 011, 0111\}$$

- $L_1^* = ?$

$$L_2^* = ?$$

- $L_1^+ = ?$

$$L_2^+ = ?$$

## Beispiel

- $L_1 = \{0, 01\}$ ,  $L_2 = \{1, 11\}$

$$L_1 \cdot L_2 = \{01, 011, 0111\}$$

—————\*

- $V = \{a, b\}$

$$V^2 = \{aa, ab, ba, bb\}, \quad V^3 = \{aaa, aab, \dots, bbb\}$$

$$\bigcup_{i \geq 0} V^i = \{\lambda\} \cup \{a, b\} \cup \{aa, ab, ba, bb\} \cup \{\dots\}$$

validieren:  $\bigcup_{i \geq 0} V^i = V^*$ , an i'sser  $V$  plekken  
ni leeg maken

$$\Rightarrow V^+ = \bigcup_{i \geq 1} V^i$$



## Peildriid

- Mier kegiid, kee  $L^* = L^+$  ?
- Iga - e unid  $L_1, L_2$  - re ?

$$(L_1 \cup L_2)^* = L_1^* \cup L_2^*$$



# A mai anyag a jegyzetben, könyvekben...

- J. Martin: 1.2 és 1.4 fejezet
- Dömösi et al.: 2.1 és 2.2 fejezet