Theoretical Computer Science

Overview of Automata

Lecture 2 - Manuel Mazzara

Different kind of Automata

- Finite State Automata (**FSA**)
  - **no temporary memory, just states are used to memorize**
- Pushdown Automata (**PDA**)
  - **stack (destructive memory), need to destroy while reading**
- Turing Machines (**TMs**)
  - **equivalent to random (non-sequential) access memory**
  - In fact, it is sequential, but does not change the computational power

Power of Automata

Simple problems		More complex problems		Hardest problems
Finite Automata	$<$	Pushdown Automata	$<$	Turing Machine

Less power → More power

Solve more computational problems

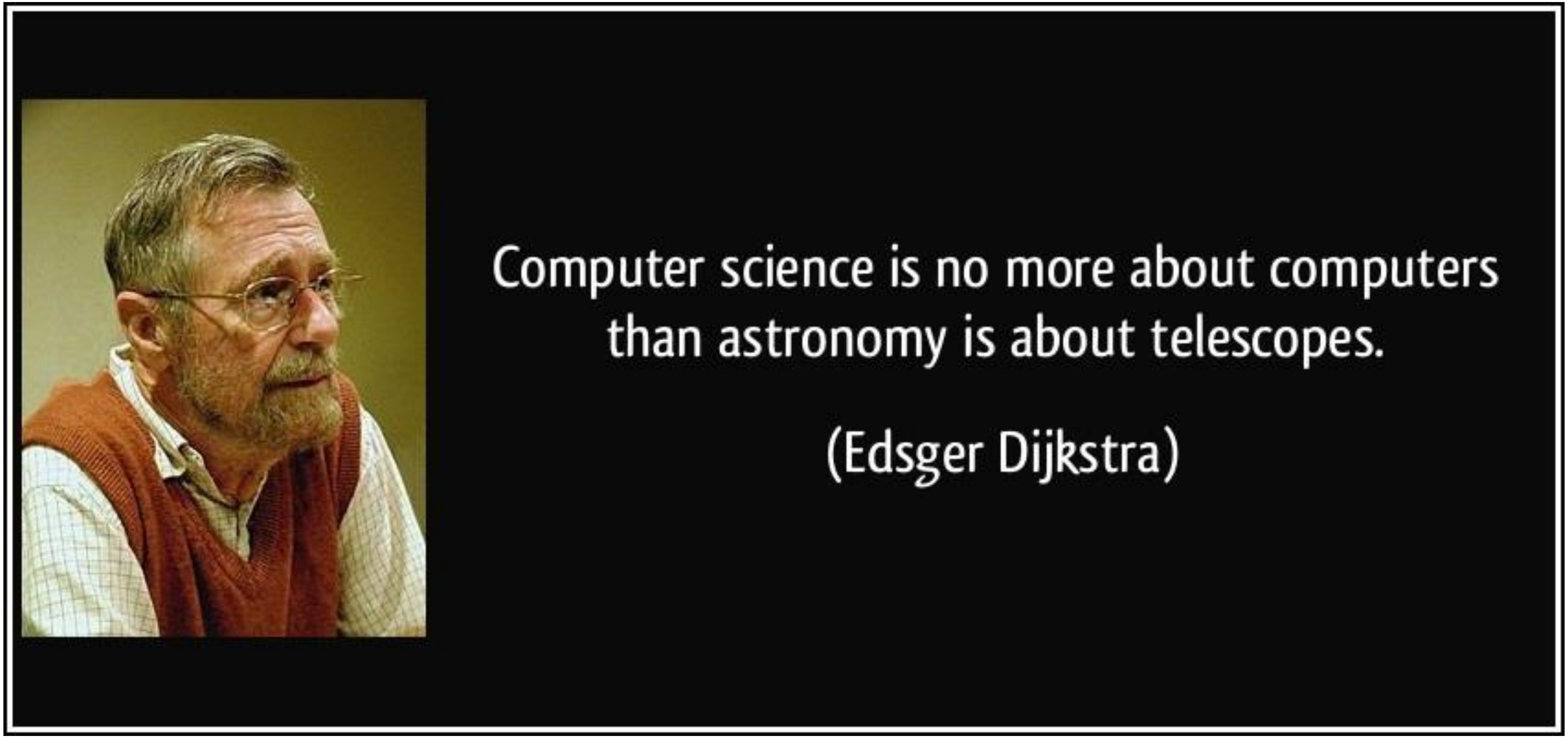
A course-long question

- **Turing Machine** is the most powerful computational model known
- Are there computational problems that a Turing Machine cannot solve?
- The Answer is “yes” (**unsolvable problems**)
- **There are indeed unsolvable problems**, and we will see in detail what this means

Theoretical Computer Science

About Theoretical Computer Science

Lecture 2 - Manuel Mazzara


Computer science is no more about computers
than astronomy is about telescopes.
(Edsger Dijkstra)

Seminal Turing's article on AI

VOL. LIX. No. 236.] [October, 1950

MIND

A QUARTERLY REVIEW

OF

PSYCHOLOGY AND PHILOSOPHY

I.—COMPUTING MACHINERY AND INTELLIGENCE

BY A. M. TURING

1. *The Imitation Game.*

I PROPOSE to consider the question, 'Can machines think ?' This should begin with definitions of the meaning of the terms 'machine' and 'think'. The definitions might be framed so as to reflect so far as possible the normal use of the words, but this attitude is dangerous. If the meaning of the words 'machine' and 'think' are to be found by examining how they are commonly

"The question of whether a computer can think is no more interesting than the question of whether a submarine can swim."

— *Edsger W. Dijkstra*

Edsger Wybe Dijkstra
11 May 1930 – 6 August 2002

- Structured Programming
- Software Engineering
- Concurrent and Distributed Computing
  - Semaphores
  - Mutual exclusion
  - Deadlock

- *Solution of a Problem in Concurrent Programming Control* - E.W. Dijkstra, Communications of the ACM, Vol. 8 , No. 9, p. 569, **1965**

*"The **revolution** in views of programming started by Dijkstra's iconoclasm led to a movement known as **structured programming**, which advocated a **systematic, rational approach to program construction**. Structured programming is **the basis for all that has been done since in programming methodology**, including object-oriented programming.”*

Bertrand Meyer - Touch of Class (page 188)

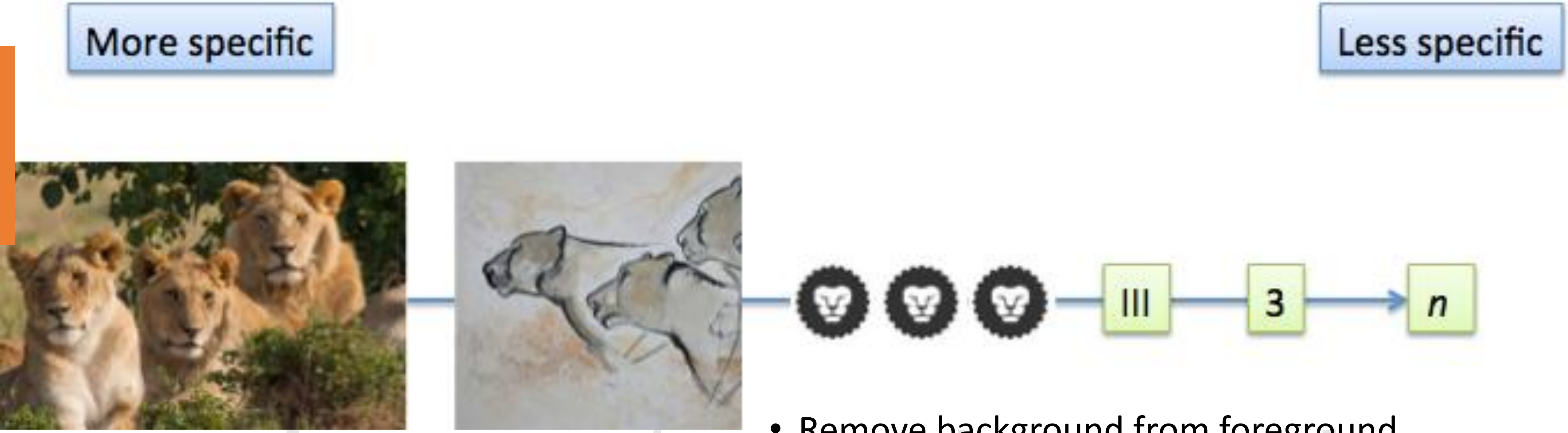
Rickmansworth
Moor Park
Northwood
Northwood Hills
Pinner
North Harrow
Harrow & Wealdstone
Kenton
Northwick Park
West Ruislip
Hillingdon
Ruislip Manor
Rayners Lane
Uxbridge
Ickenham
Ruislip
Eastcote
West Harrow
Harrow on-the-Hill
Preston Road
Ruislip Gardens
South Ruislip
South Harrow
Sudbury Hill
Sudbury Town
Alperton
Northolt
Greenford
Perivale
Hanger Lane
Park Royal
North Ealing
Ealing Broadway
West Acton
North Acton
East Acton
Ealing Common
South Ealing
Northfields
Boston Manor
Osterley
Hounslow East
Hounslow Central
Hounslow West
Hatton Cross
Heathrow Terminals 1, 2, 3
Heathrow Terminal 4
Acton Town
Chiswick Park
Turnham Green
Stamford Brook
Ravenscourt Park
Gunnersbury
Kew Gardens
Richmond
South Kenton
North Wembley
Wembley Central
Stonebridge Park
Harlesden
Willesden Junction
Kensal Green
Queen's Park
Kilburn Park
Maida Vale
Warwick Avenue
Royal Oak
Westbourne Park
Ladbroke Grove
Latimer Road
White City
Holland Park
Shepherd's Bush
Goldhawk Road
Hammersmith
Barons Court
West Kensington
Kensington (Olympia)
Earl's Court
West Brompton
Fulham Broadway
Parsons Green
Putney Bridge
East Putney
Southfields
Wimbledon Park
Canons Park
Queensbury
Kingsbury
Wembley Park
Neasden
Dollis Hill
Willesden Green
Kilburn
West Hampstead
Finchley Road
Swiss Cottage
St. John's Wood
Paddington
Bayswater
Queensway
Notting Hill Gate
High Street Kensington
Gloucester Road
South Kensington
Edgware Road
Marylebone
Baker Street
Marble Arch
Lancaster Gate
Bond Street
Hyde Park Corner
Knightsbridge
Sloane Square
Victoria
Pimlico
Vauxhall
Burnt Oak
Colindale
Hendon Central
Brent Cross
Golders Green
Hampstead
Belsize Park
Chalk Farm
Camden Town
Mornington Crescent
Great Portland Street
Euston
Euston Square
Regent's Park
Oxford Circus
Tottenham Court Road
Green Park
Piccadilly Circus
St. James's Park
Westminster
Charing Cross
Embankment
Waterloo
Lambeth North
Kennington
Oval
Stockwell
Clapham North
Clapham Common
Clapham South
Balham
Tooting Bec
Tooting Broadway
Brixton
West Finchley
Finchley Central
East Finchley
Highgate
Archway
Tufnell Park
Kentish Town
King's Cross St. Pancras
Warren Street
Goodge Street
Leicester Square
Covent Garden
Russell Square
Holborn
Aldwych
Temple
Blackfriars
Mansion House
Cannon Street
Monument for Bank
Bank
London Bridge
Borough
Elephant & Castle
Arnos Grove
Bounds Green
Wood Green
Turnpike Lane
Manor House
Finsbury Park
Arsenal
Holloway Road
Caledonian Road
Angel
Farringdon
Barbican
Moorgate
St. Paul's
Chancery Lane
Old Street
Drayton Park
Highbury & Islington
Essex Road
Tottenham Hale
Walthamstow Central
Blackhorse Road
Seven Sisters
Liverpool Street
Aldgate
Tower Hill
Fenchurch Street
Aldgate East
Shoreditch
Whitechapel
Shadwell
Wapping
Rotherhithe
Surrey Docks
New Cross Gate
New Cross
Buckhurst Hill
Roding Valley
Chigwell
Grange Hill
Woodford
Hainault
Fairlop
Barkingside
Newbury Park
South Woodford
Snaresbrook
Redbridge
Wanstead
Gants Hill
Leytonstone
Leyton
Stratford
Bethnal Green
Mile End
Stepney Green
Bow Road
Bromley-by-Bow
West Ham
Plaistow
Upton Park
East Ham
Barking
Upney
Becontree
Dagenham Heathway
Dagenham East
Elm Park
Hornchurch
Upminster Bridge
Upminster
RIVER THAMES
UNDERGROUND
Daily Services
Peak Hour and Special Services
Bakerloo Line
peak hours
Central Line
peak hours
Circle Line
Jubilee Line
Metropolitan Line
peak hours
East London Line
Piccadilly Line
peak hours
Victoria Line
British Rail
Interchange with other lines
Interchange with British Rail
Interchange with British Rail within walking distance
Closed Sundays
Closed Saturdays and Sundays
Served by Piccadilly Line early mornings and late evenings Mondays to Saturdays and all

Mathematical abstractions

- Representing real systems
  - **Abstraction allows you to focus on the important aspects of a problem**

- **Formal reasoning** can improve our ability to design and build systems
  - **Uncover design flaws**
  - Precisely **define requirements**
  - Mathematics allows you to **reason about solutions** to the problem

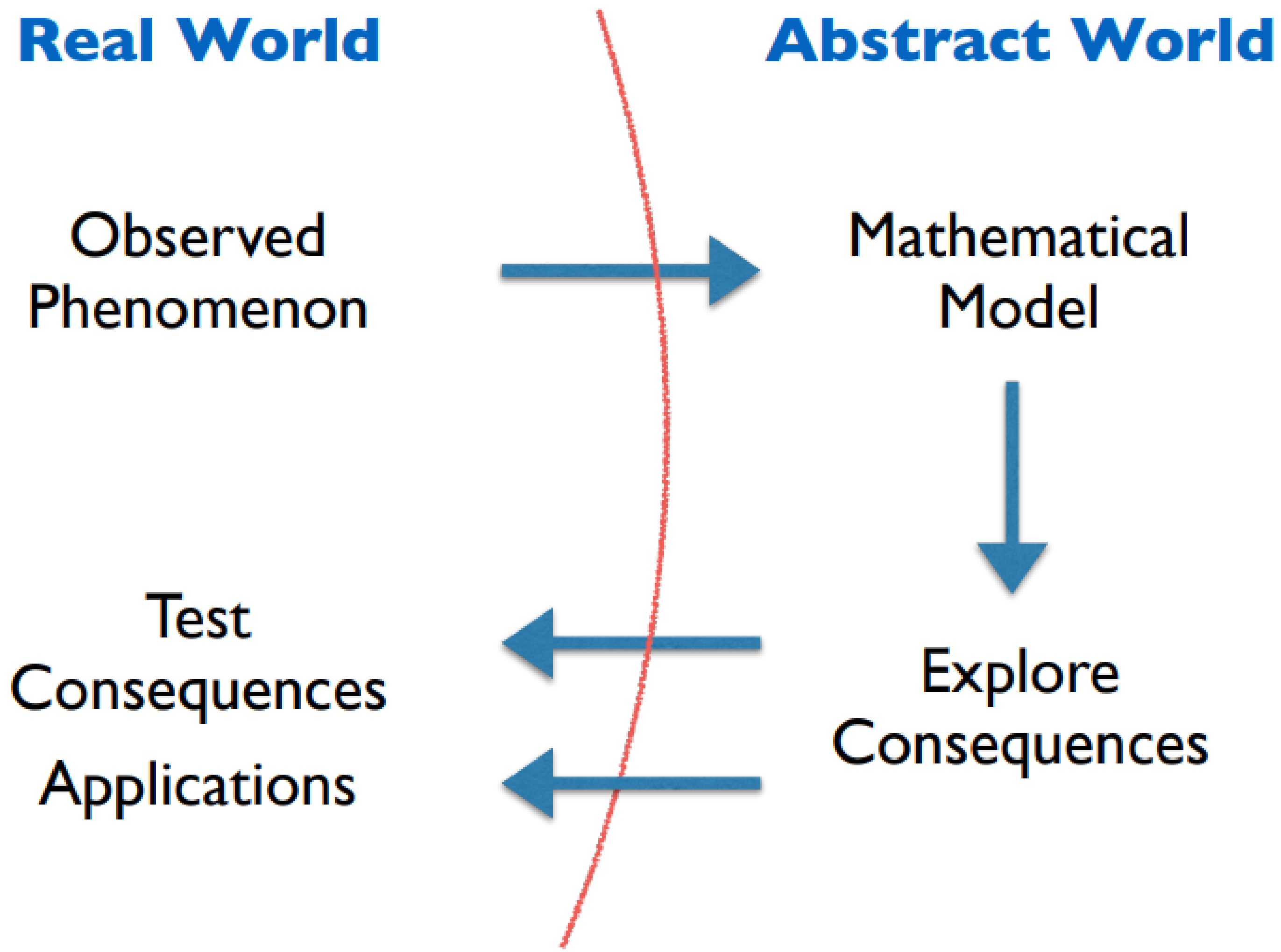
- Different models have different strengths and weaknesses



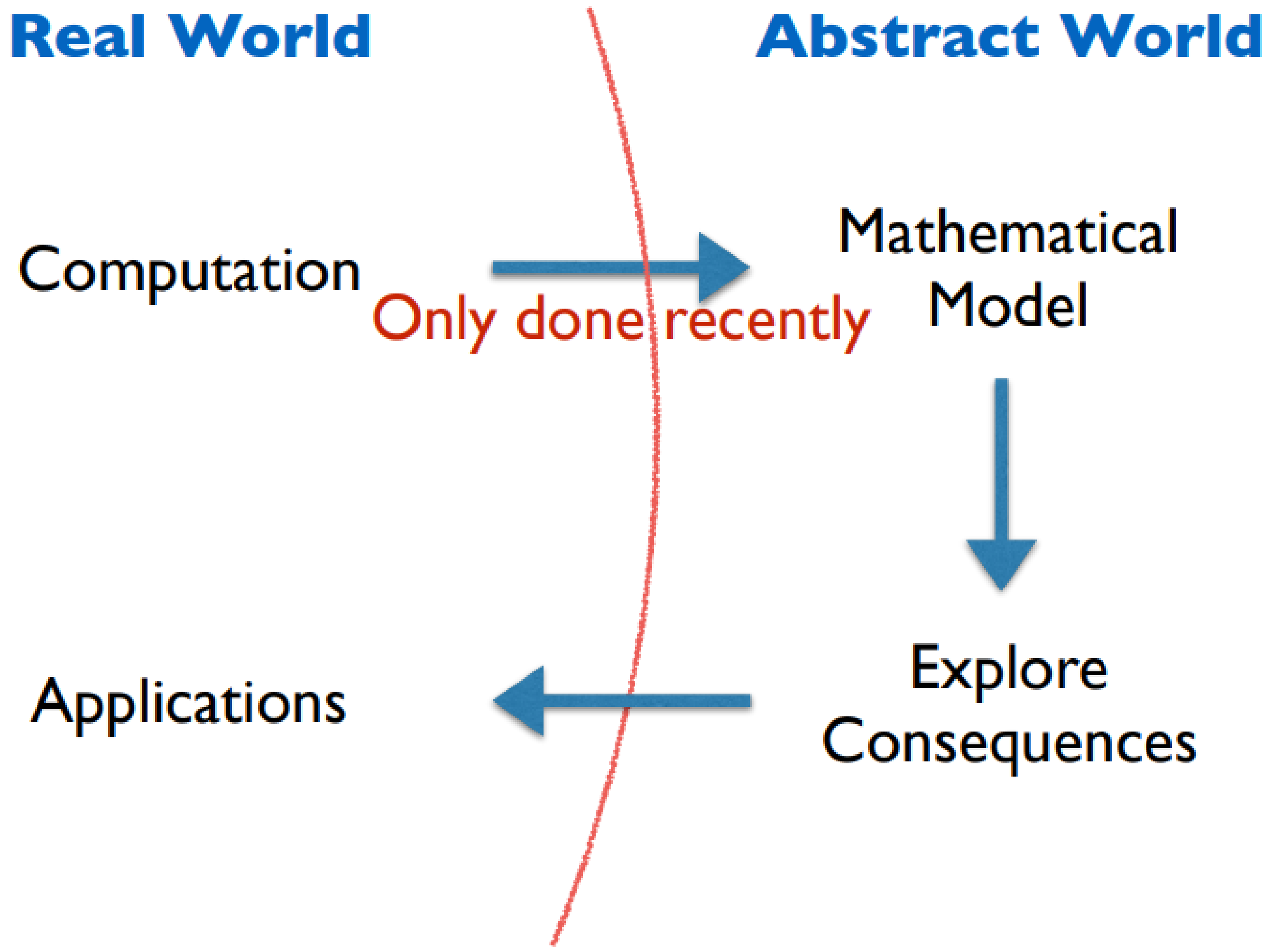
Abstraction

- Remove background from foreground
- Remove differences between each animal
- Remove "animal-ness" (treat lions as generic "lines")
- Remove need to count objects with literal lines
- Remove need to specify a fixed number

The role of theoretical physics

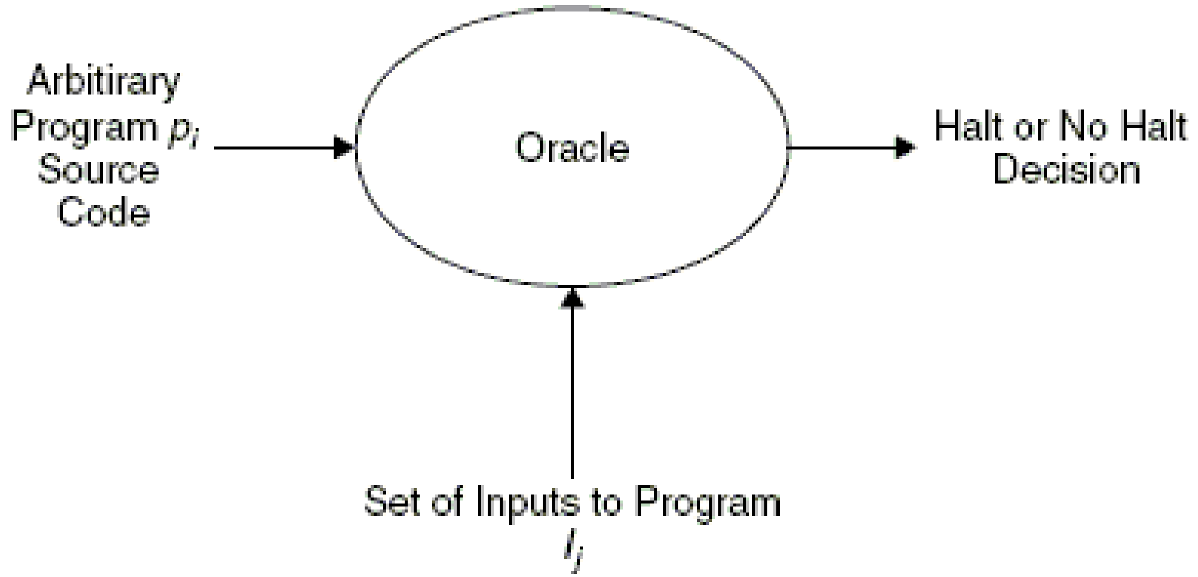


The role of theoretical computer science



Our big question

- Is every function computable?
  - Can I write an algorithm for any function N→N?
  - **Halting Problem**



Theoretical Computer Science

Introduction to Formal Languages

Lecture 2 - Manuel Mazzara

Big “existential” question (1)

- Do you know anything in nature with an infinite set of “bricks” non determined a-priori?
  - Think about chemistry, physics and and biology!

- Everything, including life seems to be expressed by **building bricks**, finite in nature and **pre-determined**, that do not change over time, and express complexity trough *“combinatorial explosion”*

- ***Can you falsify this statement?***

Periodic Table of the Elements

Normal boiling points are in °C.
SP = Triple Point
Pressure is listed if not 1 atm.
Allotrope is listed if more than one allotrope.

Atomic Number | Boiling Point | Symbol | Name | Atomic Mass

Atomic Number	Symbol	Name	Atomic Mass	Boiling Point
1	H	Hydrogen	1.008	-252.762
2	He	Helium	4.003	-268.93
3	Li	Lithium	6.941	1342
4	Be	Beryllium	9.012	2471
5	B	Boron	10.811	4000
6	C	Carbon	12.011	graphite 3825 SP
7	N	Nitrogen	14.007	-195.798
8	O	Oxygen	15.999	-182.953
9	F	Fluorine	18.998	-188.12
10	Ne	Neon	20.180	-246.053
11	Na	Sodium	22.990	882.940
12	Mg	Magnesium	24.305	1090
13	Al	Aluminum	26.982	2519
14	Si	Silicon	28.086	3265
15	P	Phosphorus	30.974	white 280.5
16	S	Sulfur	32.066	444.61
17	Cl	Chlorine	35.453	-101.5
18	Ar	Argon	39.948	-185.847
19	K	Potassium	39.098	759
20	Ca	Calcium	40.078	1484
21	Sc	Scandium	44.956	2836
22	Ti	Titanium	47.88	3287
23	V	Vanadium	50.942	3407
24	Cr	Chromium	51.996	2671
25	Mn	Manganese	54.938	2061
26	Fe	Iron	55.933	2861
27	Co	Cobalt	58.933	2927
28	Ni	Nickel	58.693	2913
29	Cu	Copper	63.546	2562
30	Zn	Zinc	65.39	907
31	Ga	Gallium	69.732	2204
32	Ge	Germanium	72.61	2833
33	As	Arsenic	74.922	616 SP
34	Se	Selenium	78.972	685
35	Br	Bromine	79.904	58.8
36	Kr	Krypton	84.80	-153.34
37	Rb	Rubidium	84.468	688
38	Sr	Strontium	87.62	1382
39	Y	Yttrium	88.906	3345
40	Zr	Zirconium	91.224	4409
41	Nb	Niobium	92.906	4744
42	Mo	Molybdenum	95.95	4639
43	Tc	Technetium	98.907	4265
44	Ru	Ruthenium	101.07	4150
45	Rh	Rhodium	102.906	3695
46	Pd	Palladium	106.42	2963
47	Ag	Silver	107.868	2162
48	Cd	Cadmium	112.411	767
49	In	Indium	114.818	2072
50	Sn	Tin	118.71	2602
51	Sb	Antimony	121.760	1587
52	Te	Tellurium	127.6	988
53	I	Iodine	126.904	184.4
54	Xe	Xenon	131.29	-108.09
55	Cs	Cesium	132.905	671
56	Ba	Barium	137.327	1897
57-71				
72	Hf	Hafnium	178.49	4603
73	Ta	Tantalum	180.948	5458
74	W	Tungsten	183.85	5555
75	Re	Rhenium	186.207	5596
76	Os	Osmium	190.23	5012
77	Ir	Iridium	192.22	4428
78	Pt	Platinum	195.08	3825
79	Au	Gold	196.967	2856
80	Hg	Mercury	200.59	356.62
81	Tl	Thallium	204.383	1473
82	Pb	Lead	207.2	1749
83	Bi	Bismuth	208.980	1564
84	Po	Polonium	[208.982]	962
85	At	Astatine	209.987	337
86	Rn	Radon	222.018	-61.7
87	Fr	Francium	223.020	677
88	Ra	Radium	226.025	1737
89-103				
104	Rf	Rutherfordium	[261]	unknown
105	Db	Dubnium	[262]	unknown
106	Sg	Seaborgium	[266]	unknown
107	Bh	Bohrium	[264]	unknown
108	Hs	Hassium	[269]	unknown
109	Mt	Meitnerium	[268]	unknown
110	Ds	Darmstadtium	[269]	unknown
111	Rg	Roentgenium	[272]	unknown
112	Cn	Copernicium	[277]	unknown
113	Uut	Ununtrium	unknown	unknown
114	Fl	Flerovium	[289]	unknown
115	Uup	Ununpentium	unknown	unknown
116	Lv	Livermorium	[298]	unknown
117	Uus	Ununseptium	unknown	unknown
118	Uuo	Ununoctium	unknown	unknown

Lanthanide Series

Atomic Number	Symbol	Name	Atomic Mass	Boiling Point
57	La	Lanthanum	138.906	3464
58	Ce	Cerium	140.115	3443
59	Pr	Praseodymium	140.908	3520
60	Nd	Neodymium	144.24	3074
61	Pm	Promethium	144.913	3000
62	Sm	Samarium	150.36	1794
63	Eu	Europium	151.966	1529
64	Gd	Gadolinium	157.25	3273
65	Tb	Terbium	158.925	3230
66	Dy	Dysprosium	162.50	2567
67	Ho	Holmium	164.930	2700
68	Er	Erbium	167.26	2868
69	Tm	Thulium	168.934	1950
70	Yb	Ytterbium	173.04	1196
71	Lu	Lutetium	174.967	3402

Actinide Series

Atomic Number	Symbol	Name	Atomic Mass	Boiling Point
89	Ac	Actinium	227.028	3198
90	Th	Thorium	232.038	4788
91	Pa	Protactinium	231.036	4027
92	U	Uranium	238.029	4131
93	Np	Neptunium	237.048	4174
94	Pu	Plutonium	244.064	3228
95	Am	Americium	243.061	2011
96	Cm	Curium	247.070	3100
97	Bk	Berkelium	247.070	2627
98	Cf	Californium	251.080	unknown
99	Es	Einsteinium	[254]	unknown
100	Fm	Fermium	257.095	unknown
101	Md	Mendelevium	258.1	unknown
102	No	Nobelium	259.101	unknown
103	Lr	Lawrencium	[262]	unknown

Alkali Metal | Alkaline Earth | Transition Metal | Basic Metal | Semimetal | Nonmetal | Halogen | Noble Gas | Lanthanide | Actinide

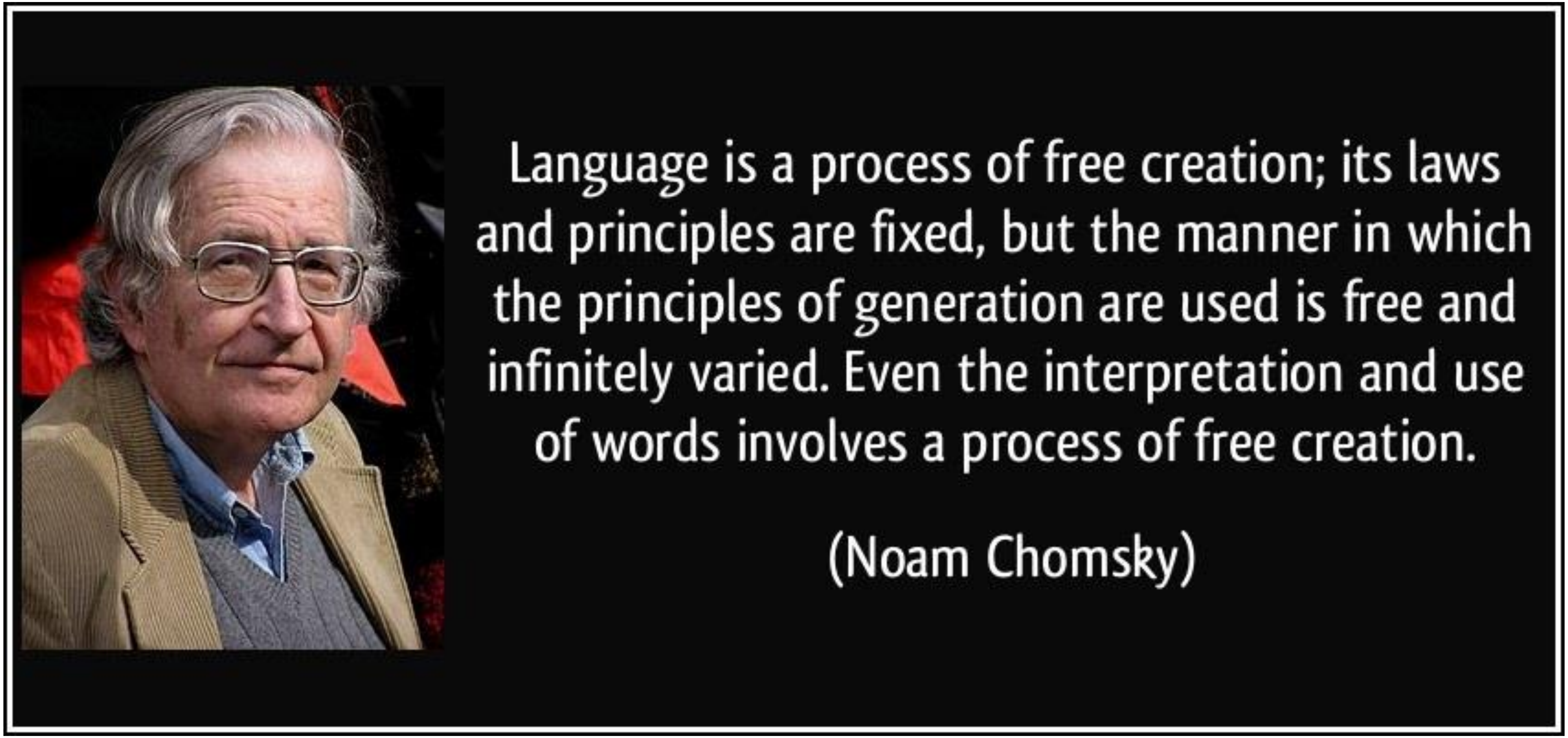
Big "existential" question (2)

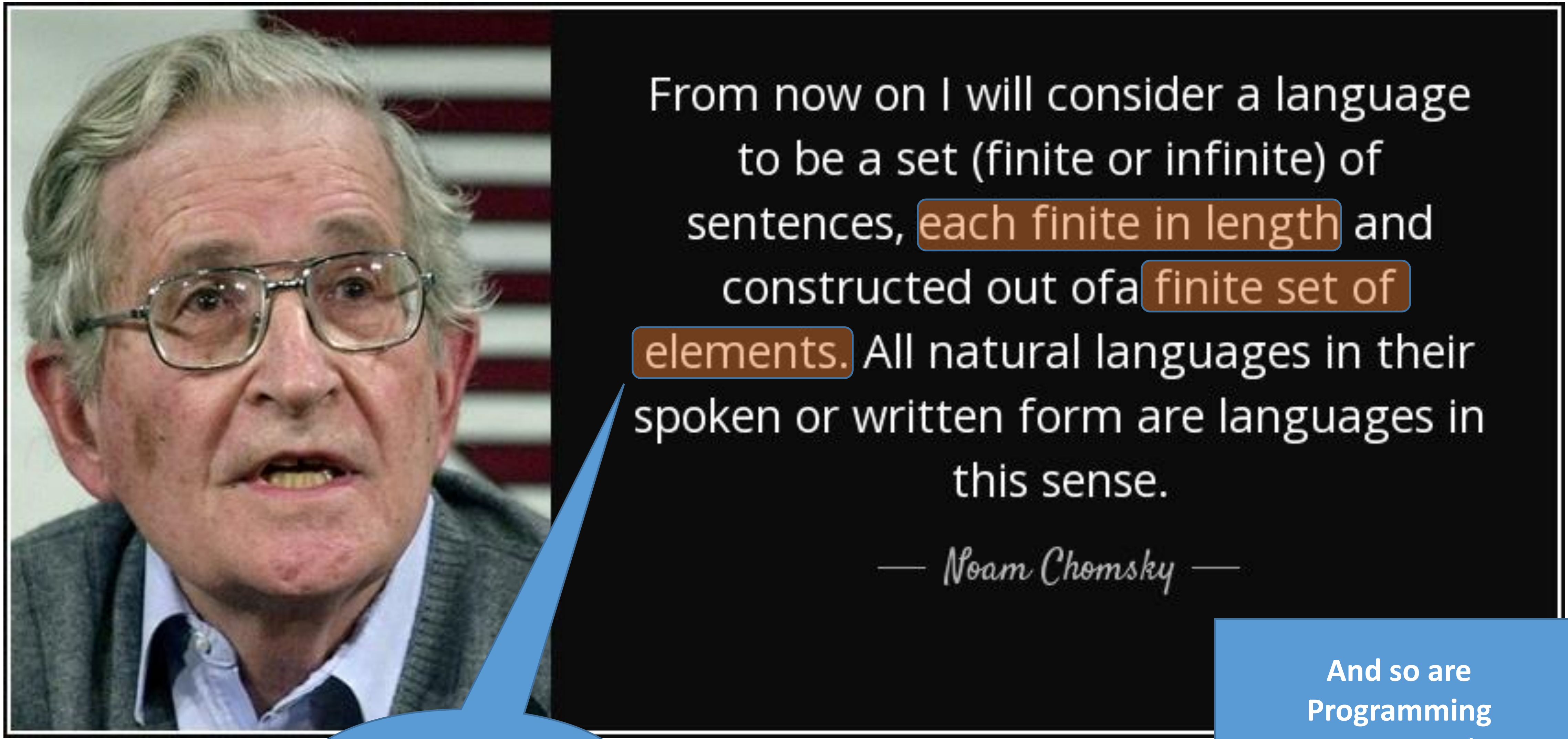
- Everything, including life seems to be expressed by **building bricks, finite in nature and pre-determined**, that do not change over time, and **express complexity trough combinatorial explosion**

- Is brain functioning any different?

- **Language** is not different!

- **Computation** is not different!


Language is a process of free creation; its laws and principles are fixed, but the manner in which the principles of generation are used is free and infinitely varied. Even the interpretation and use of words involves a process of free creation.
(Noam Chomsky)

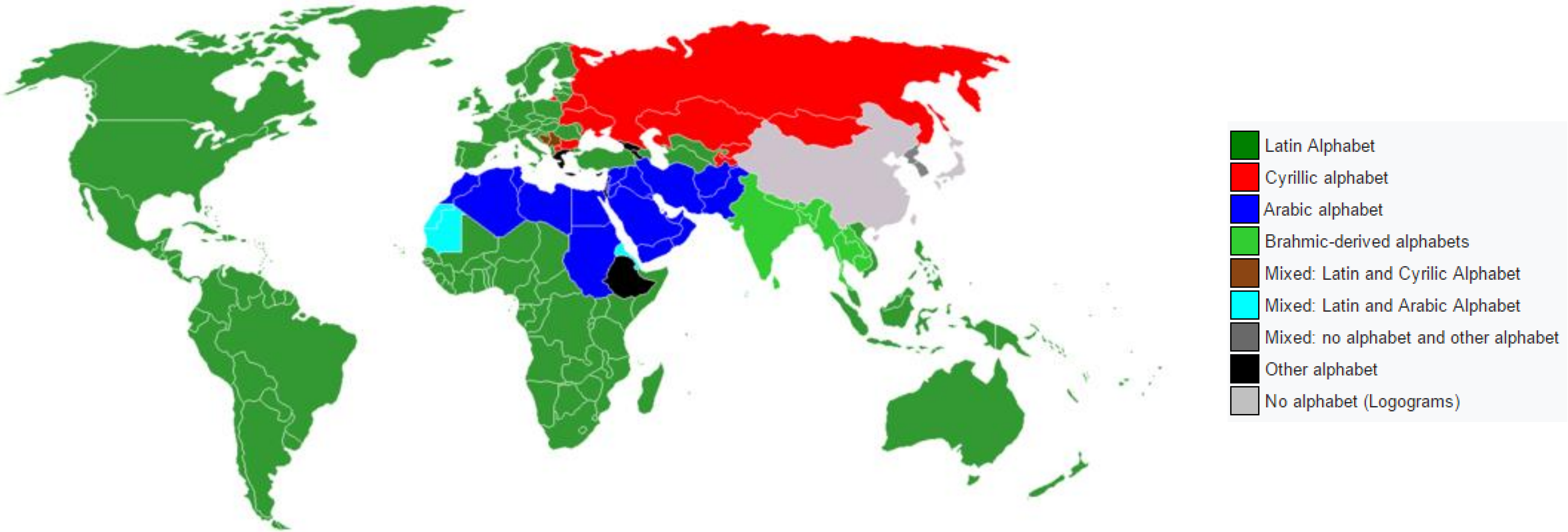


An Alphabet!

And so are Programming Languages!

What is an alphabet?

Natural languages alphabets



Elements of languages

- **Alphabet** or vocabulary
  - Finite set of **basic symbols**
  - Examples:
    - Roman alphabet $\{$a, b, …, z$\}$
    - Digits $\{$0, 1, …, 9$\}$
    - Binary alphabet $\{$0, 1$\}$
- **String** over an alphabet **A**
  - **Finite sequence of symbols** of the alphabet **A**
  - Repetitions are allowed

Examples

- Roman alphabet **A**=$\{$a, b, …, z$\}$
  - **a** is a string on **A**
  - **aa** is a string on **A**
  - **aba**, **add**, **aza**, … are strings on **A**
- Alphabet of digits **D**=$\{$0, 1, …, 9$\}$
  - **0**, **1**, **2**, …,**9** are strings over **D**
  - **012**, **999**, **923456**, … are strings over **D**

Length of a string

- The **<u>length</u>** of a string is the **<u>number of symbols</u>** contained in the string
  - We denote the length of a string **x** as **|x|**
- Examples:
  - $|a|= 1$
  - $|991346|=6$

- The **<u>empty string</u>** is a string that has zero symbols
  - We denote it as $\boldsymbol{\varepsilon}$
  - $|\varepsilon|=0$

Comparing strings

- Two strings
  - $\mathbf{x}=x_1x_2\ldots x_n$
  - $\mathbf{y}=y_1y_2\ldots y_m$

  are **equal** if and only if
  - $|x|=|y|$ $(n=m)$ Same length
  - $x_i=y_i$, $\forall i$ $(1\leq i\leq n)$ Corresponding elements are the same
- Examples
  - aabb and aabba are not equal
  - ababs and baasb are not equal

Concatenation

- Given two strings **x** and **y**, the **concatenation** (or product) of **x** and **y** is a string **xy** (or **x·y**), where **x** is followed by **y**

  – Example: strings on **A**={a, b, c, d}
    - x=abadd
    - y=dcc
    - xy=abadddcc
    - yx=dcc abadd

- Remarks
  – A string **x** concatenated with ε is still **x**
  – We abbreviate **xx** as $\mathbf{x}^2$, **xxx** as $\mathbf{x}^3$, …
  – Concatenation is associative and **non-commutative**

Substrings

- A string **x** is a **<u>substring</u>** (or a factor) of a string **s** if there exist two strings **y** and **z** such that **s=yxz**
  - y or z can be ε
    - If y= ε, x is called prefix
    - If z= ε, x is called suffix
  - If both y and z are ε, x is equal to s
- Example: s=aadabbc
  - aad is a prefix of s
  - abbc is a suffix of s
  - ada is a substring of s

Kleene Star

- The **Kleene star** is a **unary operator** that applies to a set of symbols or a set of strings
  - It is denoted as *
  - In algebra it is called the **free monoid** on a set
- If **A** is an alphabet then **A*** is the **set of all strings over symbols in A**, including the **empty string**.
- Examples:
  - If A={a, b, c} then A*={ε, a, b, c, aa, ab, ac, ba, bb, bc, ca, …}
  - If B={0, 1} then B*={ε, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, …}

Do you remember what is a monoid in abstract algebra (the study of algebraic structures)?

Free monoid

- A <u>**monoid**</u> is a set equipped with an **associative binary operation** and an **identity element**

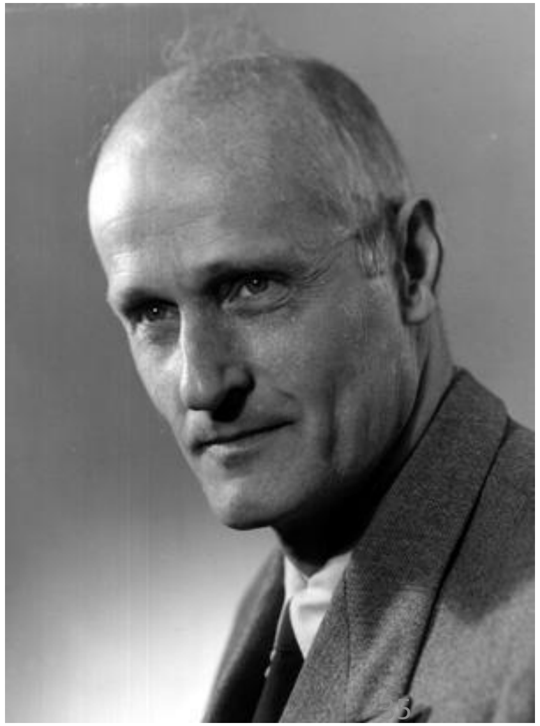
- The **free monoid** on a set is the monoid whose elements are **all the finite sequences (or strings) of zero or more elements from that set**
  - **String concatenation** is the monoid operation
  - The unique sequence of zero elements, the empty string (denoted by ε or λ) is the **identity element** (it leaves any element of the set unchanged when combined with it)

- **The <u>free monoid</u> on a set A is usually denoted A$*$**

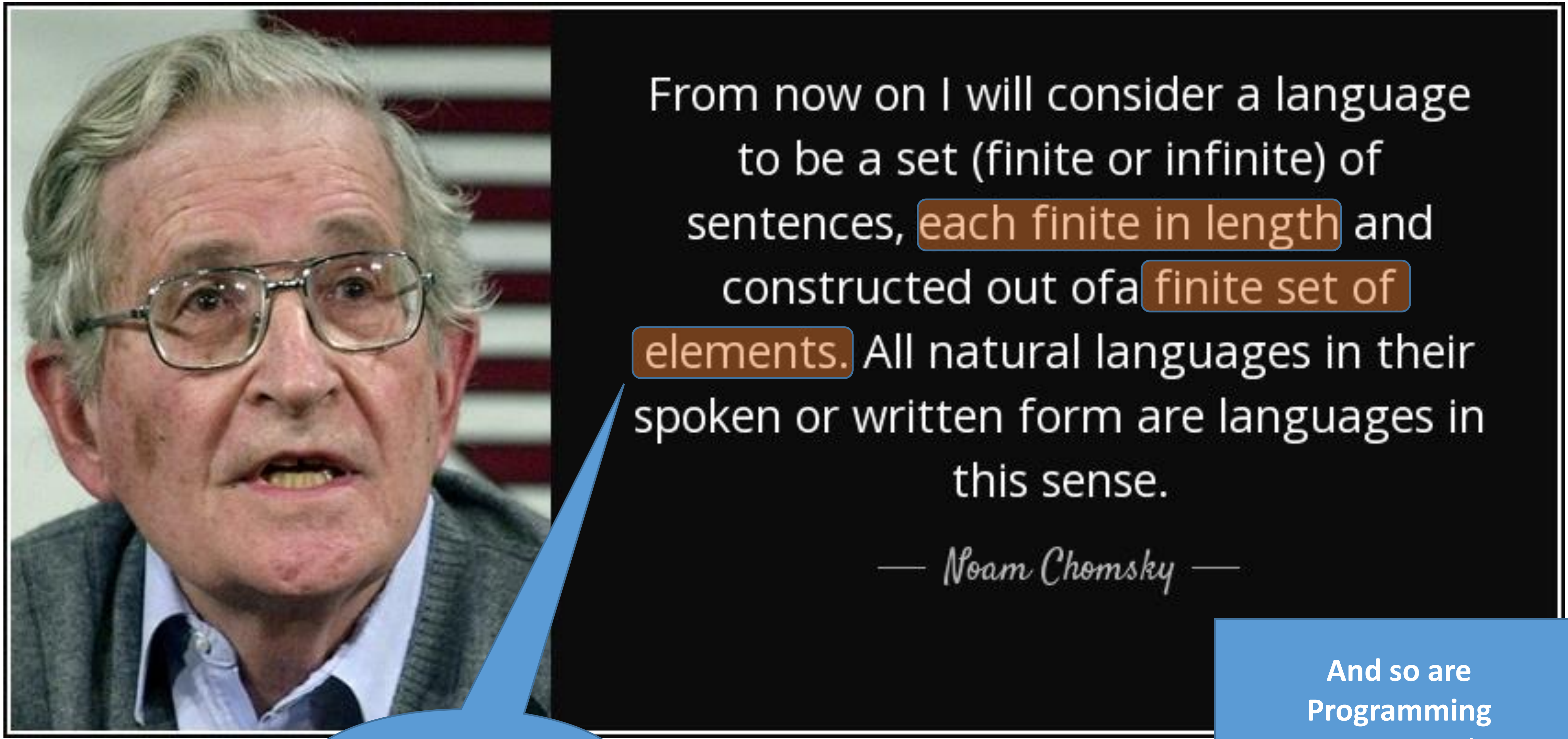
Stephen Kleene

- Kleene star is widely used for **regular expressions**
- It was introduced by **Stephen Kleene** in this context
- **Stephen Kleene** (1909-1994)
  - American mathematician
  - Student of <u>**Alonzo Church**</u>

Lambda calculus
Church-Turing thesis



What is a language?



An Alphabet!

And so are Programming Languages!

What map is this?

Languages

- A **<u>language is a set of strings over an alphabet</u>**
- Languages:
  - Russian, Italian, English, French
  - C, Java, Pascal, Eiffel

  but also

  - Graphical languages
  - Music
  - Multimedia

Formally

- A language **L** over an alphabet **A** is a <u>**subset**</u> of **A***
- Examples
  - $A=\{a, b, c\}$

    $A^*=\{\varepsilon, a, b, c, aa, ab, ac, ba, bb, bc, ca, \dots\}$

    $L_1= \{\varepsilon, a, b, c, bc, ca\}$

    $L_2= \{aa, ab, ac, ba, bb, bc, ca, cb, cc\}$

Operations

- Operations on **<u>sets</u>** apply also to **<u>languages</u>**
  - **A language is a <u>set of strings</u>**
- Operations on languages are
  - Union
  - Intersection
  - Difference
  - Complement
  - Concatenation
  - Power of n
  - Kleene star/closure

Set operations (1)

- $L_1 \cup L_2$
  - Example:

    $L_1 = \{\varepsilon$, a, b, c, **bc**, **ca**$\}$

    $L_2 = \{$ba, bb, **bc**, **ca**, cb, cc$\}$

    $L_1 \cup L_2 = \{\varepsilon$, a, b, c, ba, bb, bc, ca, cb, cc$\}$
- $L_1 \cap L_2$
  - Example: $L_1 \cap L_2 = \{$bc, ca$\}$

Set operations (2)

- $L_1 \setminus L_2$ (or $L_1 - L_2$)
  - Generally used when $L_2 \subseteq L_1$
  - Example:

    $L_1 = \{$ ba, bb, bc, ca, cb, cc $\}$

    $L_2 = \{$ bc, ca $\}$

    $L_1 \setminus L_2 = \{$ba, bb, cb, cc$\}$
- $L^c = A^* \setminus L$
  - A is the alphabet over which L is defined
  - Example: L_1^c = set of all strings on $\{a,b,c\}^*$ except the strings of length 2 that start with a 'b' or a 'c'

Concatenation

- $L_1 \cdot L_2$ (or L_1L_2)=$\{x \cdot y \mid x \in L_1, y \in L_2\}$
  - Remark: '$\cdot$' is <u>not commutative</u>
  - $L_1 \cdot L_2 \neq L_2 \cdot L_1$
- Example

  $L_1 = \{\varepsilon$, a, b, c, bc, ca$\}$

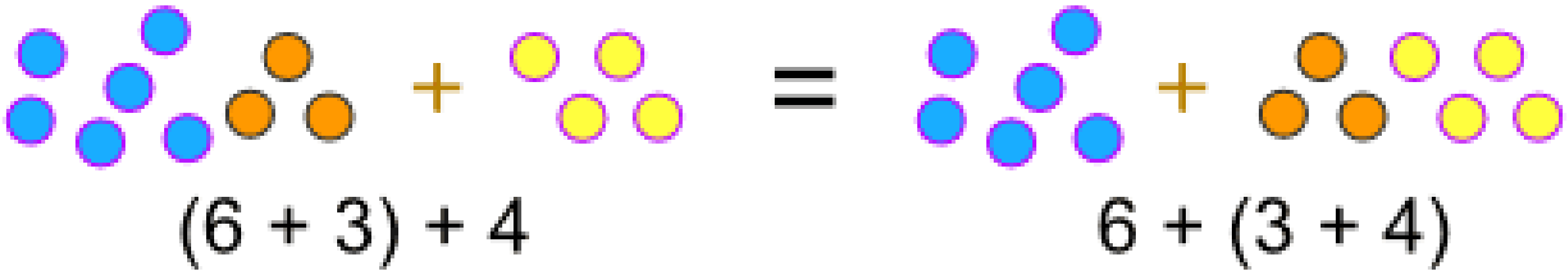
  $L_2 = \{$ba, bb, bc, ca, cb, cc$\}$

  $L_1L_2 = \{$ba, bb, bc, ca, cb, cc, aba, abb, abc, aca, acb, acc, bba, bbb, bbc, bca, bcb, bcc, cba, cbb, cbc, cca, ccb, ccc, bcba, bcbb, bcbc, bcca, bccb, bccc , caba, cabb, cabc, caca, cacb, cacc$\}$

Power

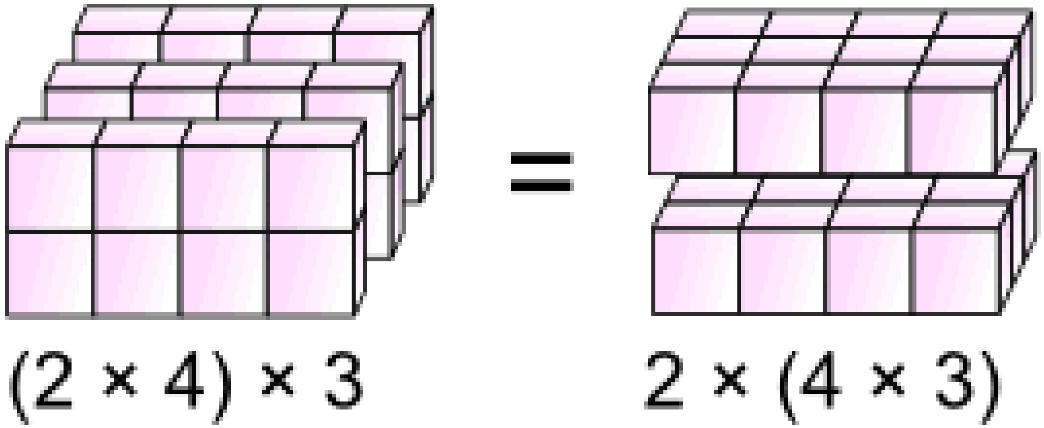
- L^n is obtained by concatenating L with itself n times
  - $L^0 = \{\varepsilon\}$
  - $L^i = L^{i-1} \cdot L$
- Examples:
  - $L^2 = L \cdot L$
  - $L^3 = L \cdot L \cdot L$
  - $L^4 = L \cdot L \cdot L \cdot L$
  - …
- Remark: '$\cdot$' is associative

Associative Law (1)



Do you remember it?

Associative Law (2)



Do you remember it?

Kleene Star (1)

- Kleene star is a **unary operation**, either on sets of strings or on sets of symbols

- The application of the Kleene star to a set A is written as A*

- Defined by Stephen Kleene in the context of regular expressions (will see this later in the course)

Kleene Star (2)

Given a set V *we define:*

$V_0 = \{\varepsilon\}$ (the language consisting only of the empty string),

$V_1 = V$

$V_{i+1} = \{\, wv : w \in V_i$ and $v \in V \,\}$ for each $i>0$.

Inductive definition

$$V^* = \bigcup_{i \in \mathbb{N}} V_i = \{\varepsilon\} \cup V \cup V_2 \cup V_3 \cup V_4 \cup \ldots$$

What do formal languages represent?

- A language is a **set of strings**
  - $L_1 = \{bc, ca\}$
  - $L_2 = \{ba, bb, bc, ca, cb, cc\}$
  - $L_3 = \{x \in \{a,b\}^* \mid (\exists\, y \in \{a,b\}^*)\ x = ay\}$

- How can sets of strings be applied in computer science?
  - Formal languages are not only mere mathematical representations

Languages in CS

- A language is a way of representing or communicating information
  - Not just meaningless strings
- There are many kinds of languages
  - Natural languages
  - Programming languages
  - Logic languages
  - …

Example (1)

- Consider the following languages:
  - L_1: set of "Word@Mac" documents
  - L_2: set of "Word@PC" documents
- Operations:
  - L_1^c is set of documents that are not compatible with "Word@Mac"
  - $L_1 \cup L_2$ is the set of documents that are compatible with either Mac or PC
  - $L_1 \cap L_2$ is the set of documents that are compatible with both Mac and PC

Example (2)

- Consider the following languages:
  - L_1: set of e-mail messages
  - L_2: set of spam messages
- Operations:
  - L_1-L_2 implements a filter

Languages in practice

- A language can represent
  - **Computations**
  - Documents
  - Programs
  - Multimedia

- **Operations on languages create new classes of languages**