Can Computers Discover New Mathematics?

A glimpse into a new type of mathematics research

Roderic Guigo Corominas, Summer 2025

About myself

My Educational Journey

- Bachelors in mathematics & physics at the Universitat de Barcelona
- PhD in mathematics at Boston University
- Preceptor in mathematics at Harvard University:
  - Teach undergraduate courses
  - Teach in the Pre-College program
  - Research mathematics education
  - Research pure and computational mathematics

About myself

My Mathematics Journey

- In high school my favorite subject was physics and chemistry
- In college, I enjoyed algebra, geometry and topology the most
- After, I went to graduate school for a short program in data science & big data
- During my PhD, I enjoyed using computational tools to show new examples, help with calculations, make conjectures in algebraic geometry
- Currently, I like to constantly learn new applications of computing to mathematics. This summer, I am mentoring two Harvard undergraduates in mathematics and machine learning projects

Answer: Yes

But why?

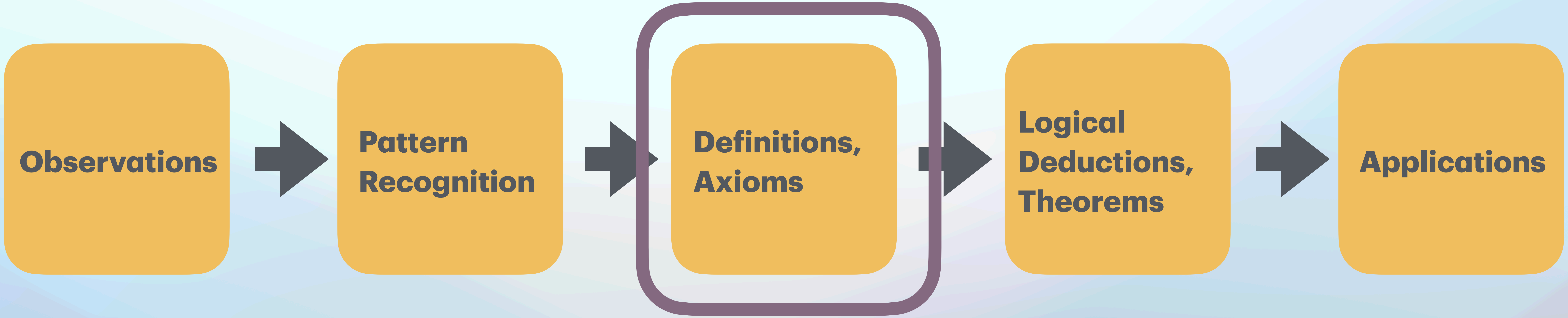
Mathematics

Part 1: How Humans do Math. Understanding the formalism

What is Mathematics?

Axiomatic Systems

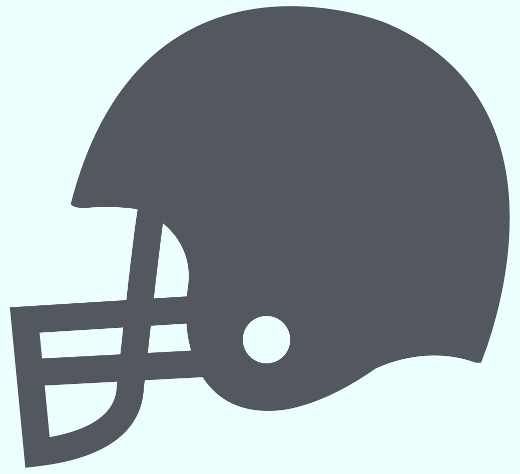
- Mathematics is both a language and a discipline—a framework for understanding patterns, structures, quantities, and logical relationships.
- It is at once abstract and practical, foundational and exploratory.

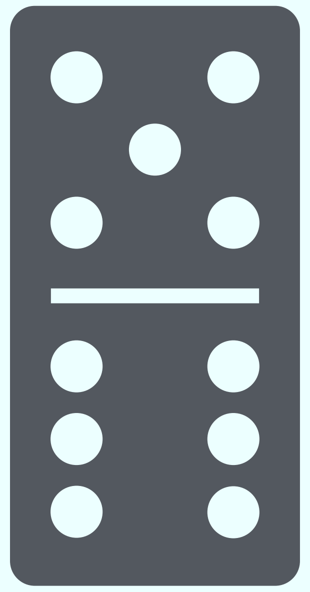


What is Mathematics?

Axiomatic Systems

- An **axiom** is like a rule. For example, in chess, if a given position is repeated 3 times, the game is considered a draw.
- An **axiomatic system** is a set of rules that (ideally) satisfy:
  - Consistency: axioms don't contradict each other
  - Completeness: every true statement can be derived from the axioms
  - Independence: no axiom can be proven from another axiom
- Any statement that can be derived logically from a set of axioms is a **Theorem**



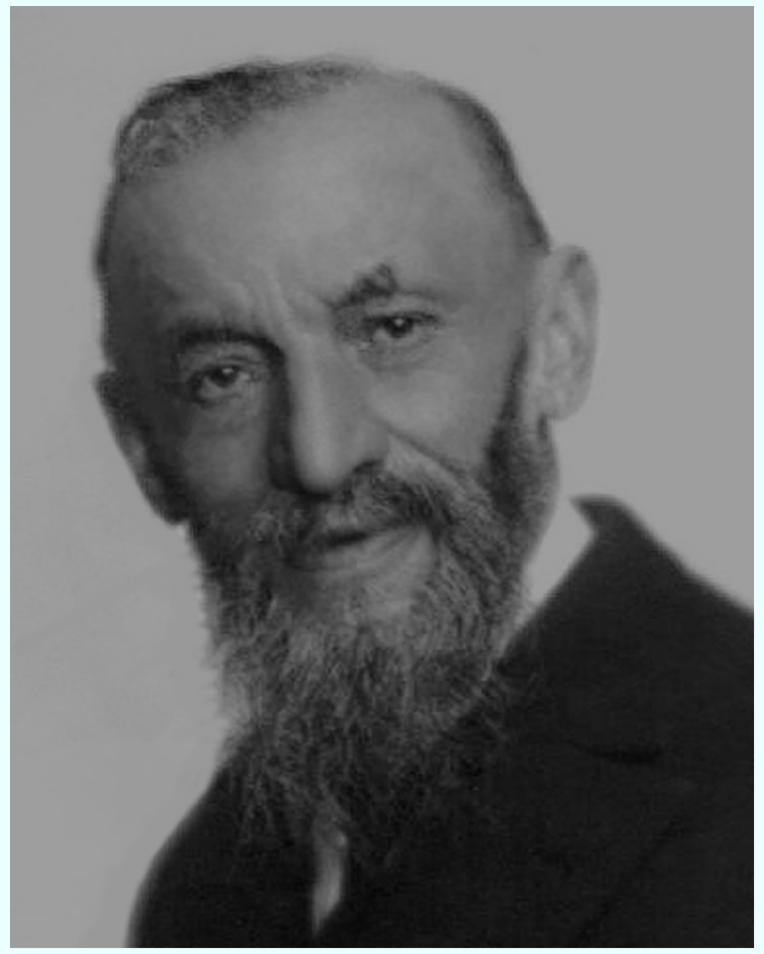


Just a set of rules. Sounds easy!

Axiomatic Systems

Natural Numbers

- Zero if a natural number: $0 \in \mathbb{N}$
- Successor is natural number: $\forall n \in \mathbb{N}, S(n) \in \mathbb{N}$
- Zero is not the successor of any number: $\forall n \in \mathbb{N}, S(n) \neq 0$
- Successor function is injective: $\forall m, n \in \mathbb{N}, S(m) = S(n) \Rightarrow m = n$
- Induction axiom: for any property $P(n)$, if
  - $P(0)$ holds, and
  - $\forall n \in \mathbb{N}, P(n) \Rightarrow P(S(n))$, then
  - $\forall n \in \mathbb{N}, P(n)$



Giuseppe Peano

Peano Axioms for the natural numbers

$\mathbb{N} = \{1,2,3,4,5,\ldots\}$

how does one add numbers?

Axiomatic Systems

Not so simple after all ...

- Zero if a natural number: $0 \in \mathbb{N}$
- Successor is natural number: $\forall n \in \mathbb{N}, S(n) \in \mathbb{N}$
- Zero is not the successor of any number: $\forall n \in \mathbb{N}, S(n) \neq 0$
- Successor function is injective: $\forall m, n \in \mathbb{N}, S(m) = S(n) \Rightarrow m = n$
- Induction axiom: for any property $P(n)$, if
  - $P(0)$ holds, and
  - $\forall n \in \mathbb{N}, P(n) \Rightarrow P(S(n))$, then
  - $\forall n \in \mathbb{N}, P(n)$

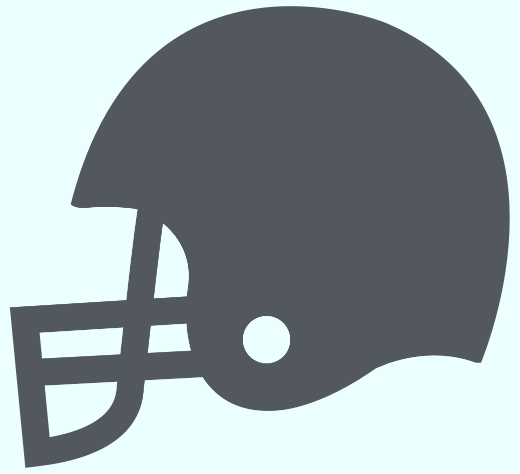
- Addition of natural numbers. Defined recursively:
  - $a + 0 = a$
  - $a + S(b) = S(a + b)$

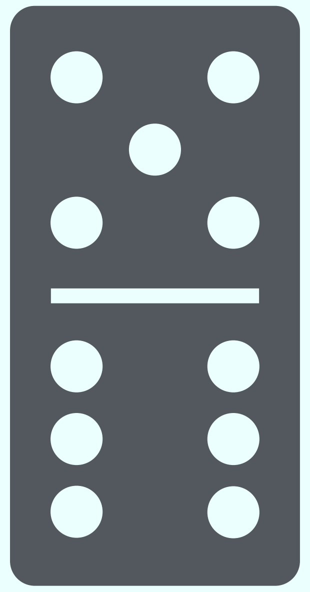
- Now we can finally prove $1 + 1 = 2$
  - $1 = S(0)$ and $2 = S(1) = S(S(0))$
  - $1 + 1 = S(0) + S(0) = S(0 + S(0)) = S(S(0)) = 2$

Axiomatic Systems

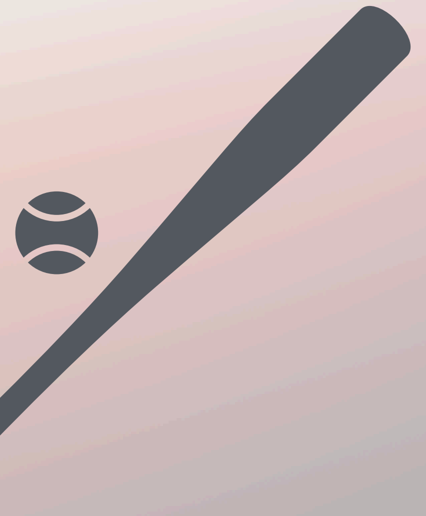
Summary: how to create/discover new math

- Make a set of axioms (from observations in physics, biology, economics): **Natural Numbers**
- Define some interesting objects of study: **Prime Numbers**
- Combine axioms and logical reasoning to make new theorems describing properties of these objects: **Euclid's Theorem (there are infinitely many prime numbers)**
- You have created/discovered new math!





Ok, Rules are complicated, but can they at least explain it all?



Axiomatic Systems

Caveats, Part I

- In any sufficiently powerful axiomatic system for arithmetic (numbers and basic operations)
  - 1) is incomplete. That is it has true statements that **cannot** be proven.
  - 2) **cannot** prove its own consistency.

Kurt Godel

Axiomatic Systems

Caveats, Part II

- Mathematicians don’t even agree about which axiomatic we should choose
    - 1) ZF: Zermelo-Fraenkel
    - 2) ZFC: with axiom of choice?
- And there are other axioms that cannot be determined to be independent
    - 1) Continuum hypothesis

Summary

- Math is hard
- We can’t even agree about axiomatic systems
- Our axiomatic systems are not great, or at least not perfect
- Computers, fundamentally, treat just with 0s and 1s
- How can they be helpful?

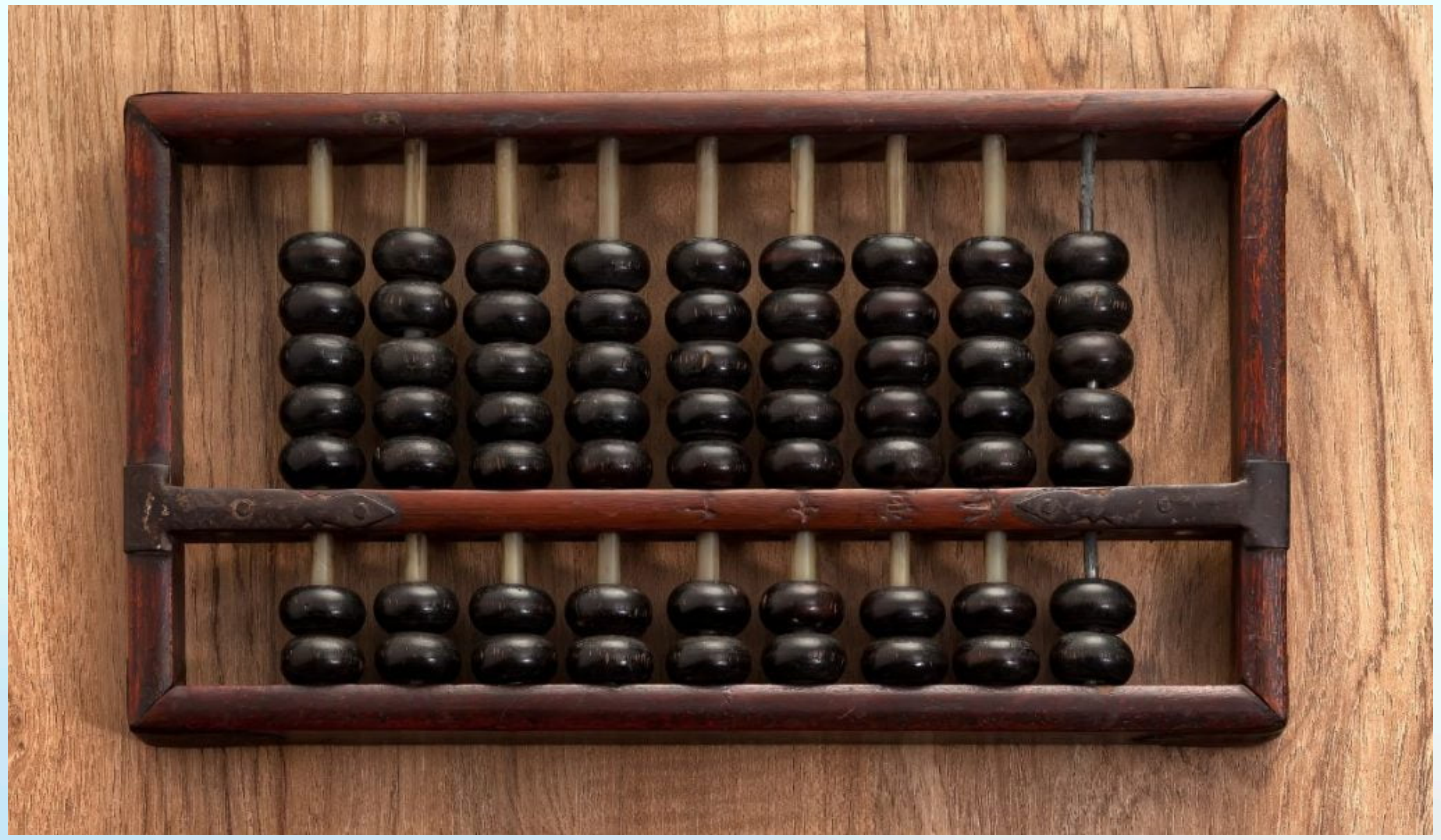
Computation meets Mathematics

Part II: Computers are Good at "Computing"

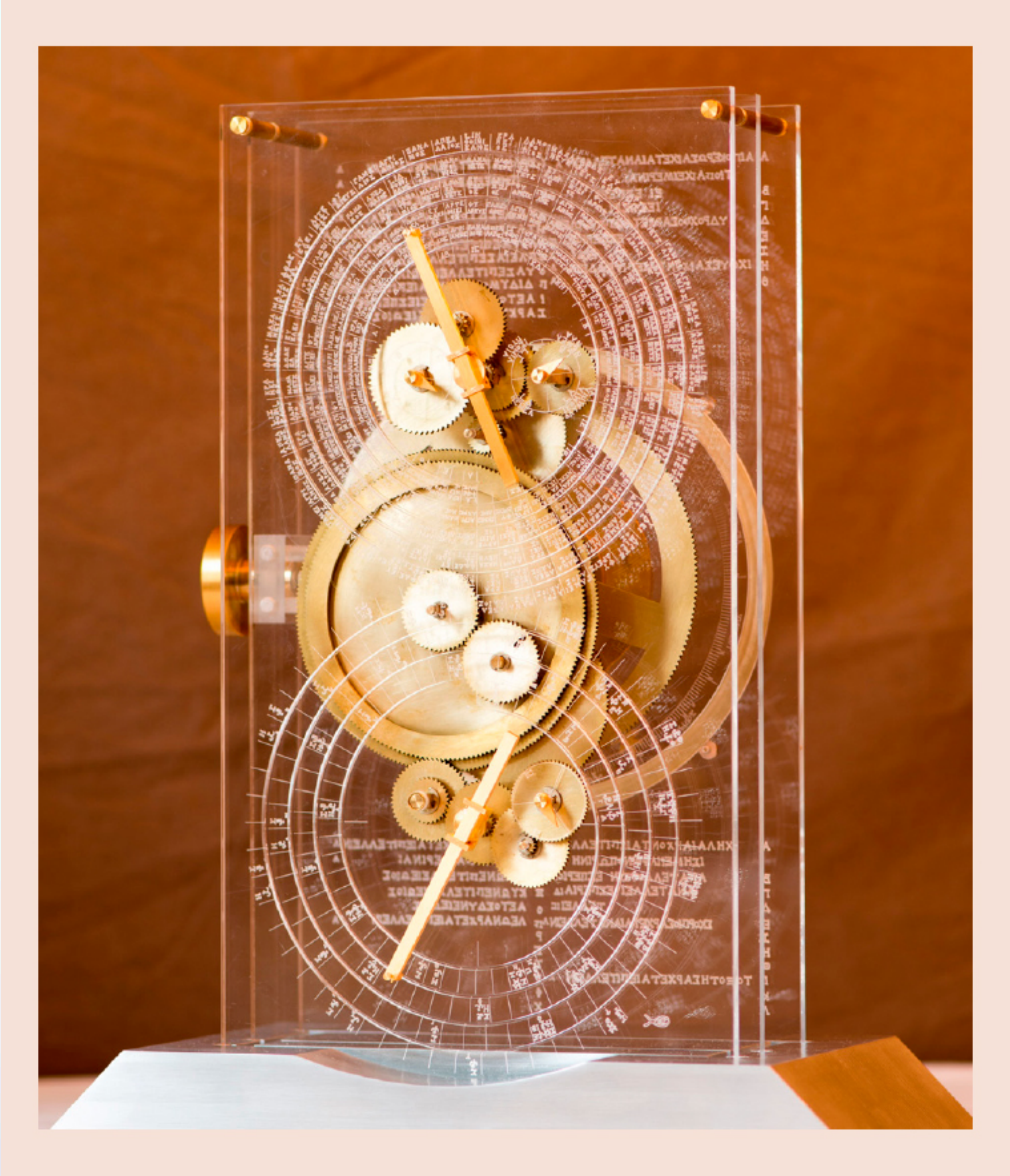
History

Early Devices

- Computing tools exist since ancient times



Abacus (ca. 2700 BCE)



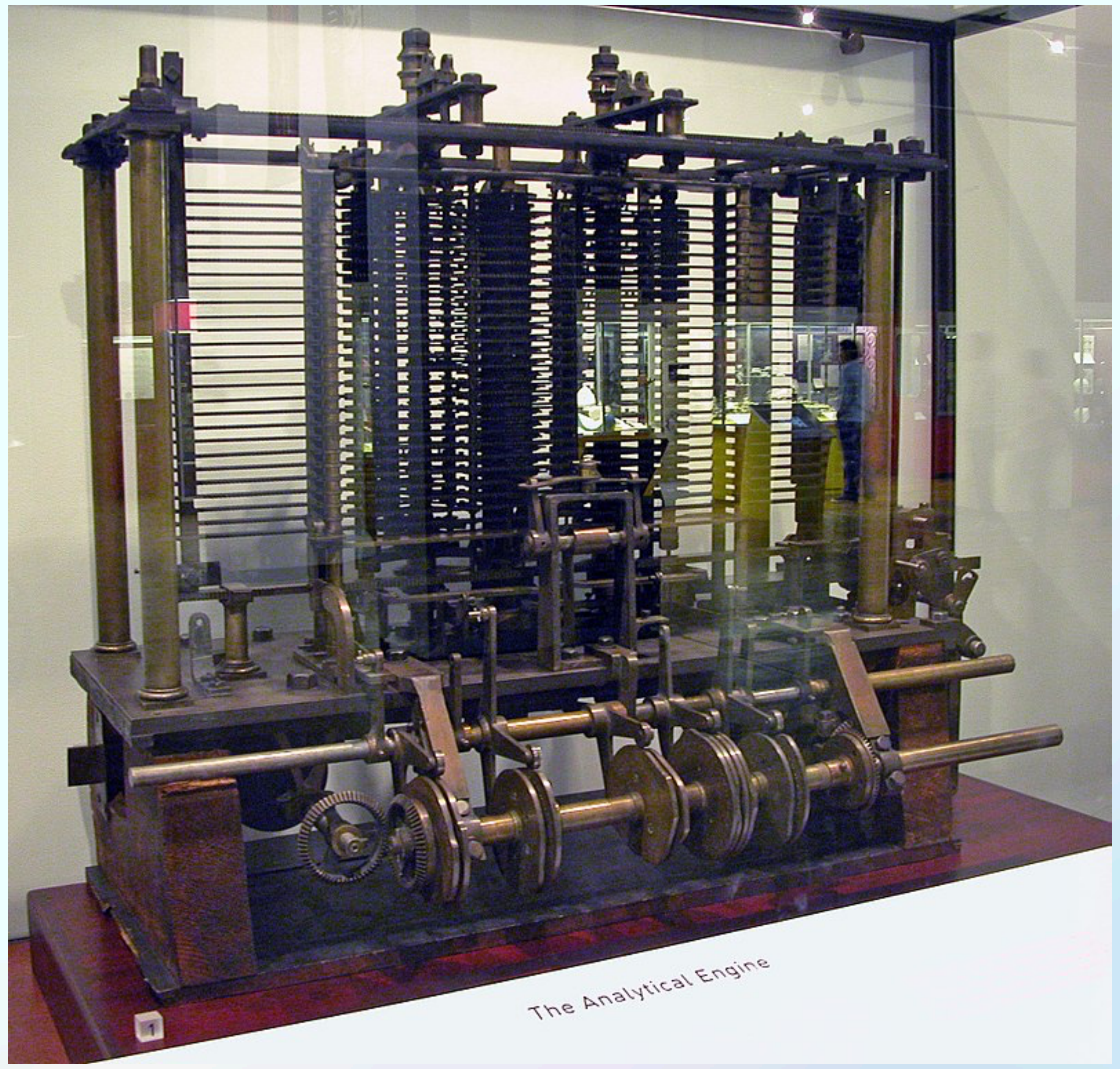
Antikythera Mechanism (ca. 100 BCE)

History

Mechanical Devices (1600s - 1900s)

Pascaline (1642)

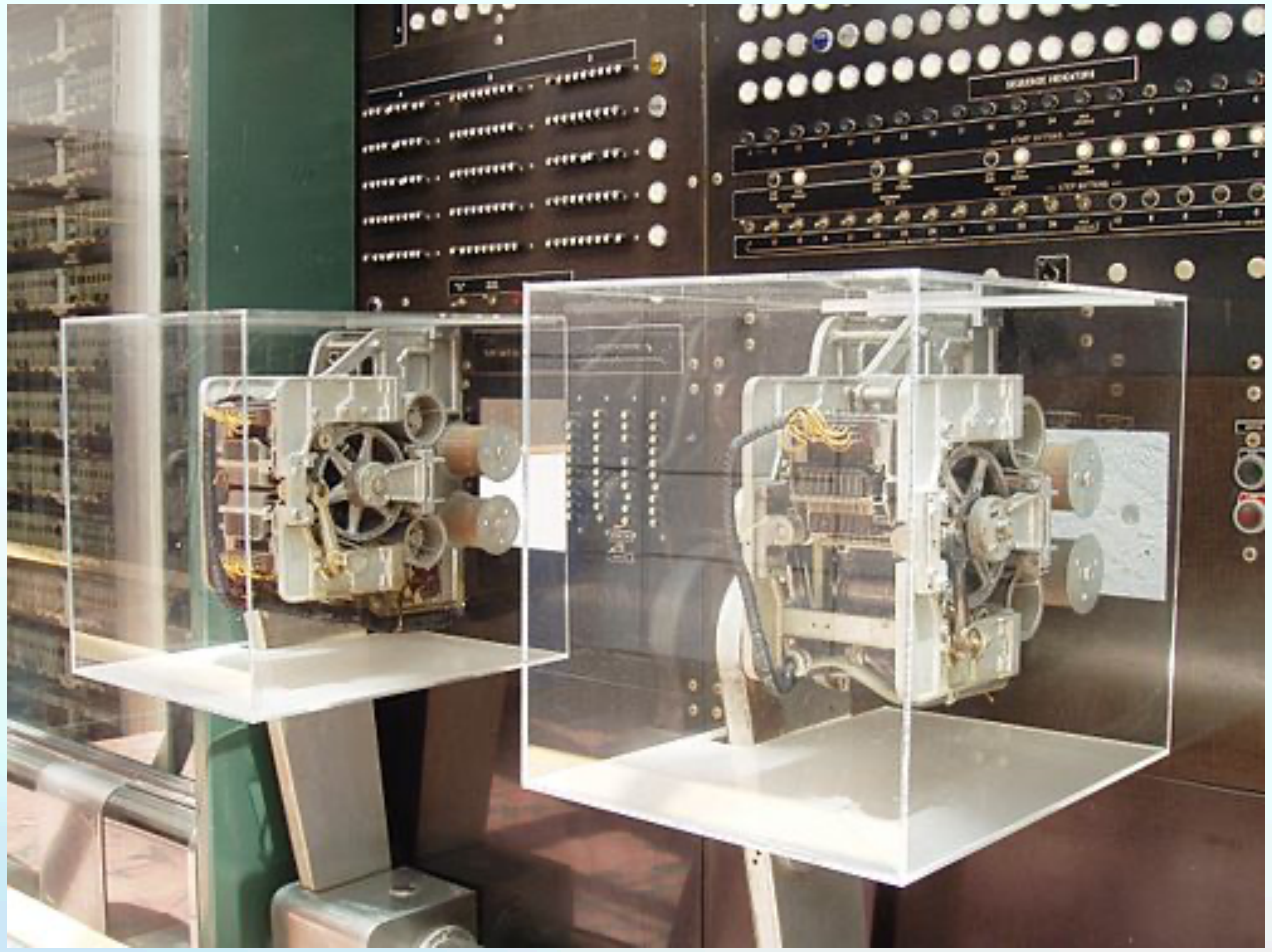
Stepped Reckoner (1642)



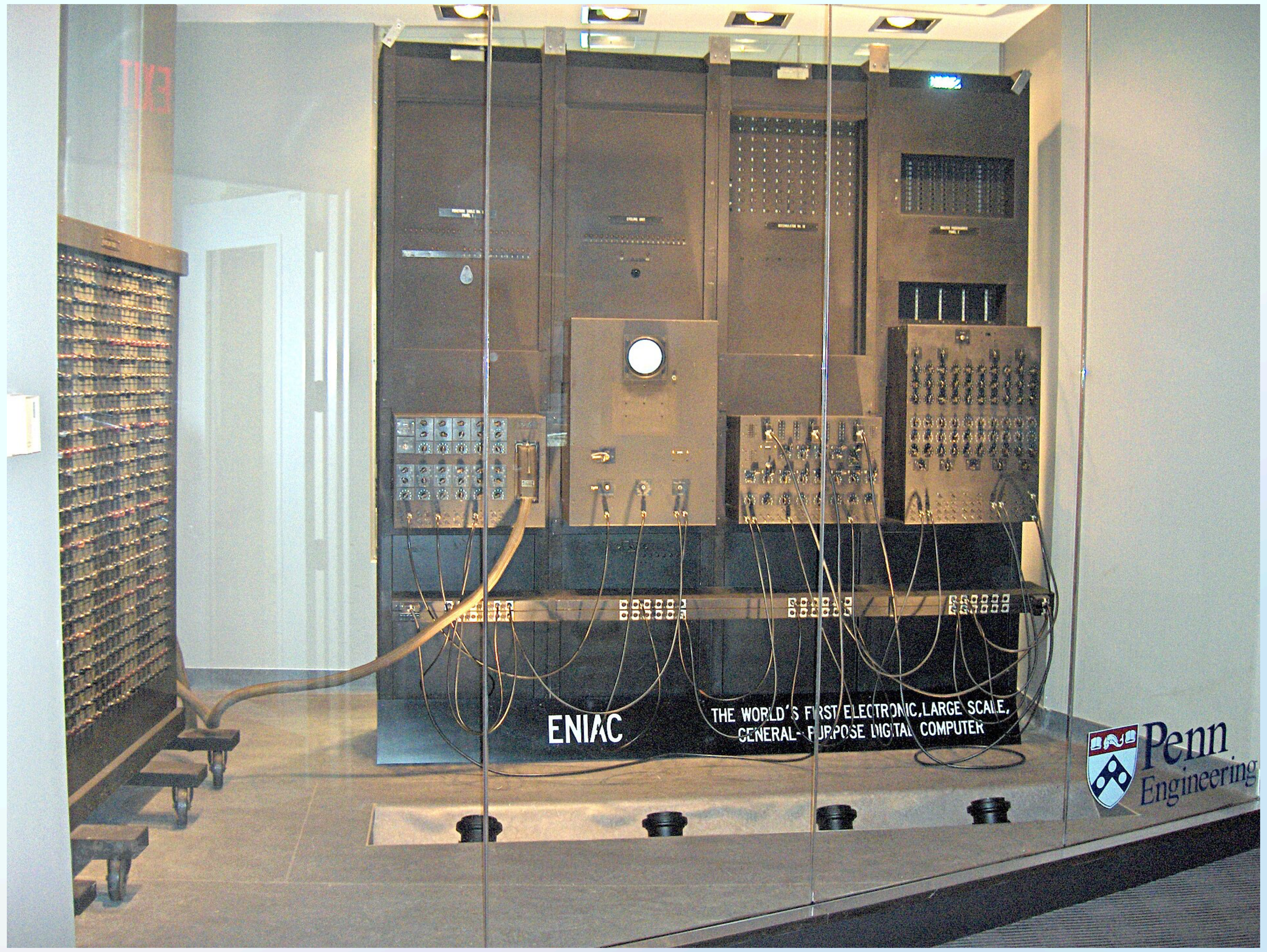
Analytical Engine (1642)

History

Early Computers



Harvard Mark I (1944)

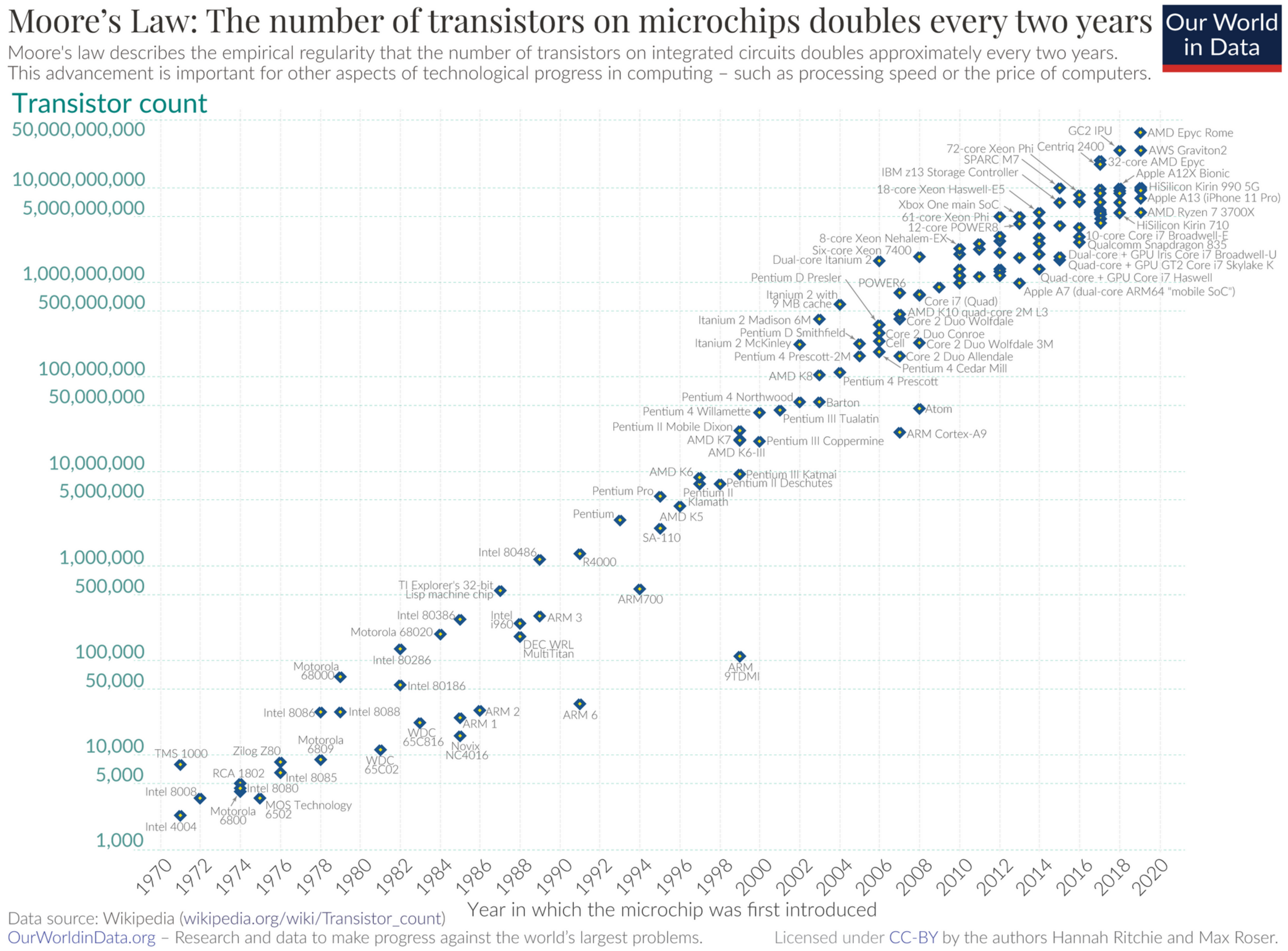


ENIAC (1945)

History

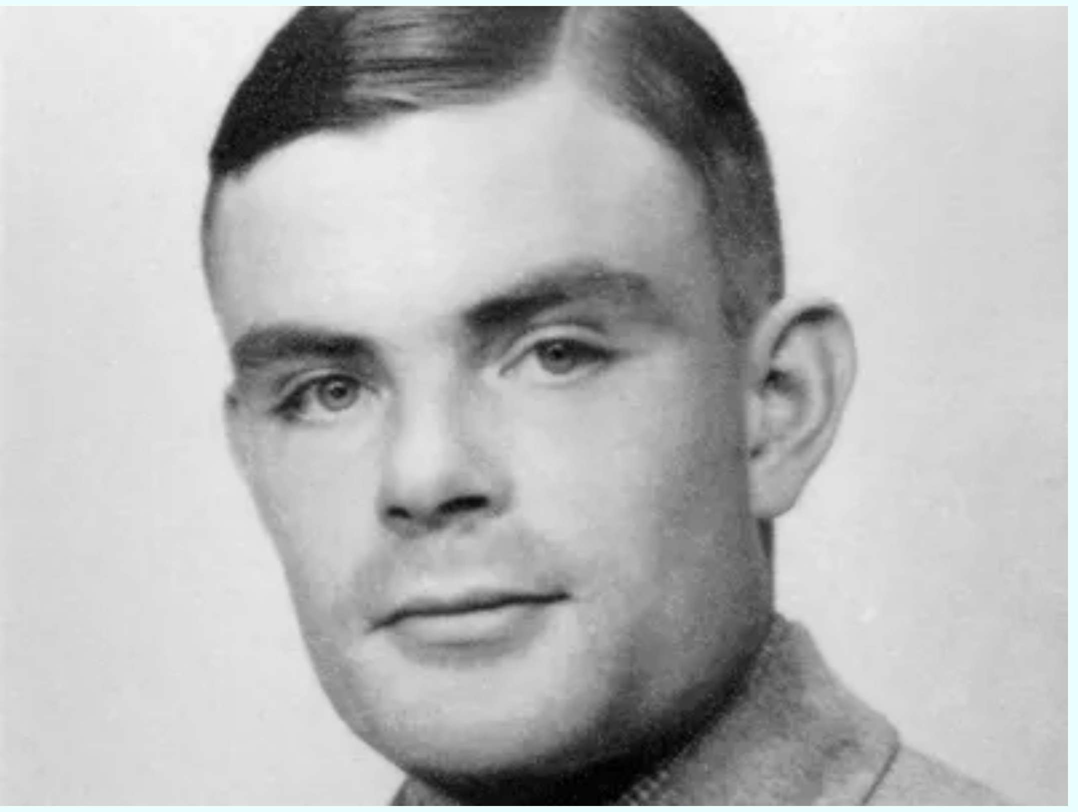
Modern Computers

- The growth in computing "power" is exponential (Moore's Law)
- Has allowed rapid advance in algorithms
- Some examples are:
  - Monte Carlo simulations (1940s)
  - QR Factorization (1950s)
  - Fast-Fourier Transform (1960s)
  - Fast Multipole Method (1980s)



History

Modern Computers



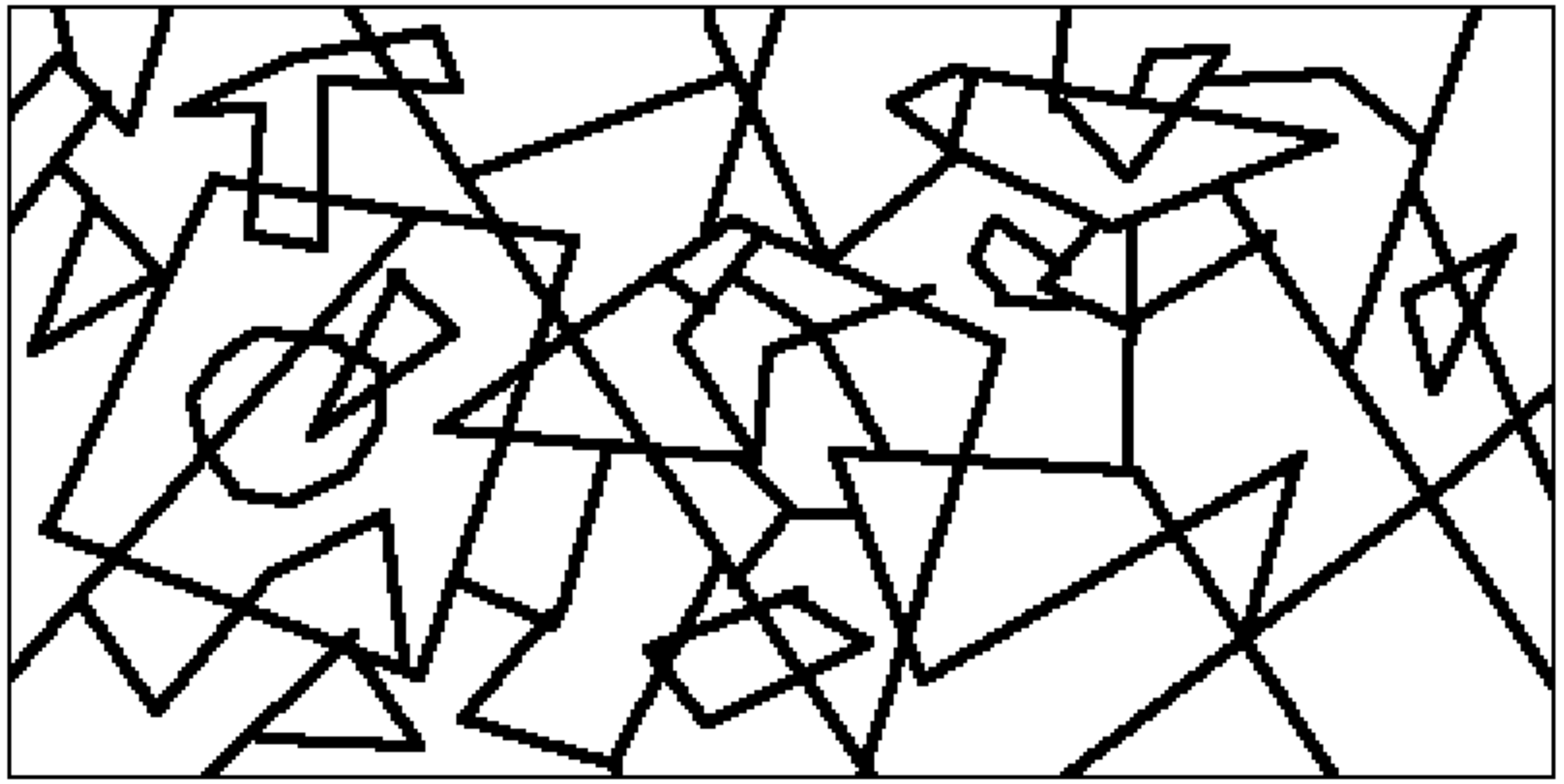
Alan Turing

- Are Turing Complete:
  - They can be used to simulate a Turing Machine
  - It is a measure of how powerful the system is
  - In other words: Modern computers can compute anything that is **theoretically computable** by an algorithm
  - This still has limitations of memory, computation time, etc.

Coloring Maps

First Major Computer Proof

- What is the minimum number of colors needed to color a map, such that no two adjacent regions have the same color?



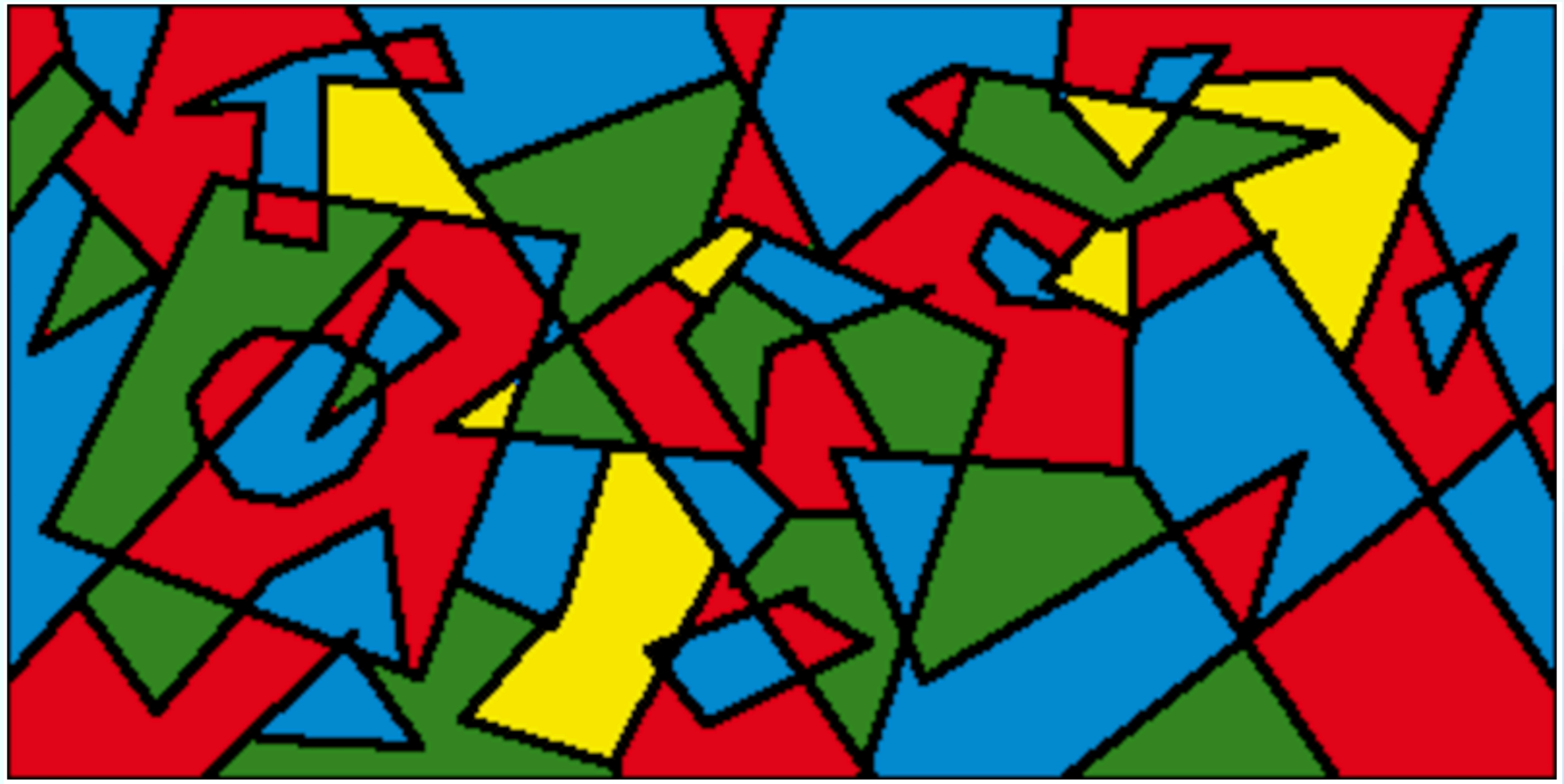
What is the minimum number of colors needed to color a map, such that no two adjacent regions have the same color?

- 3 — 0%
- 4 — 0%
- 5 — 0%
- Infinite — 0%

The Four Color Theorem

by

- Heinrich Heesch: developed methods in the 1970s but lacked computer power
- Kenneth Appel & Wolfgang Haken: No more than **four** colors are needed so that no two adjacent regions have the same color
- First "major" theorem to be proved using a computer
- Required some trial, error and controversy



The Sum of Three Cubes

Another surprisingly recent result

- Which integers n can be written as a sum of three cubes $x^3 + y^3 + z^3 = n$?
  - zeros are allowed
  - negatives numbers are allowed

- Your turn: can you find solutions for $n = 1,2,3,4,5$?

Which of the following numbers can be written as the sum of three cubes?

1
0%

2
0%

3
0%

4
0%

5
0%

The Sum of Three Cubes

Another surprisingly recent result

- $1 = 1^3 + 1^3 + (-1)^3$
- $2 = 7^3 + (-5)^3 + (-6)^3$ (or $2 = 1214928^3 + 3480205^3 + (-3528875)^3$)
- $3 = 1^3 + 1^3 + 1^3$
- $4 = ?$
- $5 = ?$

Not Possible!!!

- Every cube has a remainder of $1, 0, -1$ when divided by 9
- Combining three of these numbers we can get $-3, -2, -1, 0, 1, 2$ or 3
- Not possible to obtain 4 or 5

The Sum of Three Cubes

Another surprisingly recent result

- In 2009, solutions were known for $n \leq 1000$ except 33, 42, 74, 114, 165, 390, 579, 627, 633, 732, 795, 906, 921, and 975.

- Case $n = 74$ solved in 2016 (Huisman)

- Cases $n = 33, 42, 165, 579$ and 795 solved in 2019 (Booker & Sutherland) after **1.3 million hours** of computing.

What do you think?

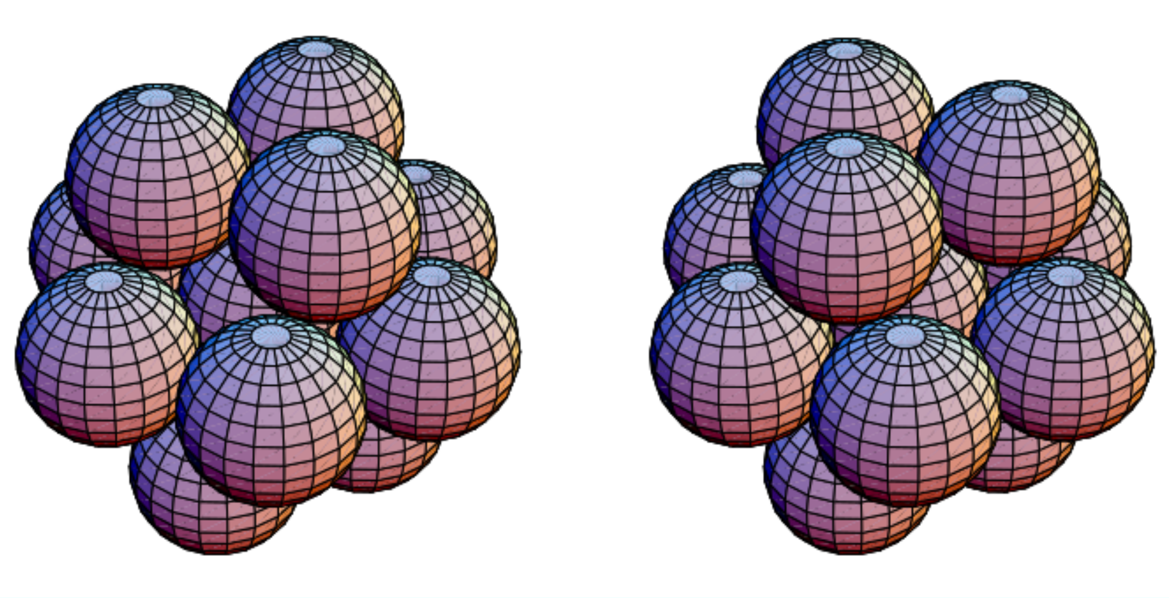
What is new math?

- Finding 12148143151523194182481934901841 x 12951353152192018949143019431804
- Finding the next largest prime number (currently $2^{136279841} - 1$)
- Finding integers x, y and z such that $x^3 + y^3 + z^3 = 114$
- Finding a proof or a counterexample to the Riemann Hypothesis

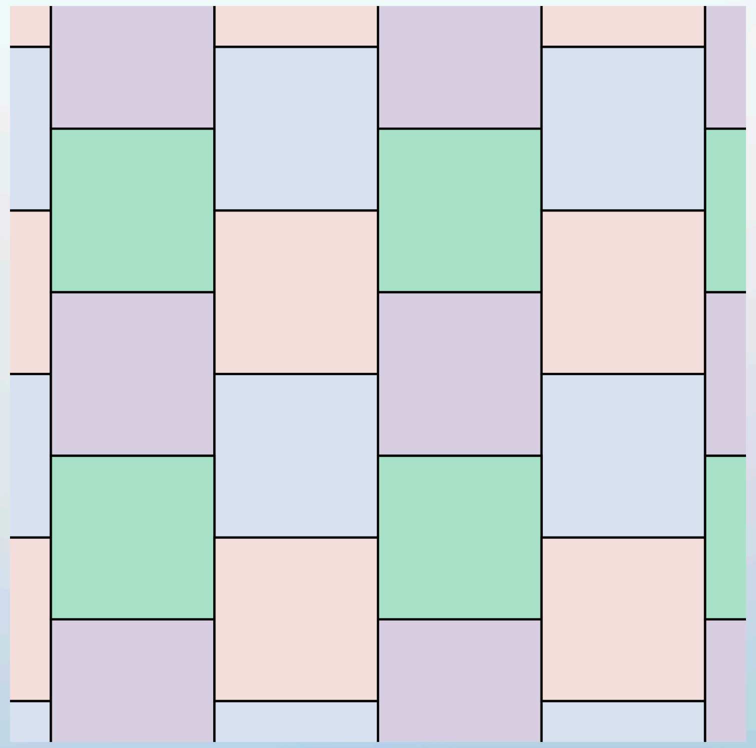
It is up to humans to implement the algorithms, but also to decide **what is interesting/important**

Computer Assisted Proofs

Other Major Examples



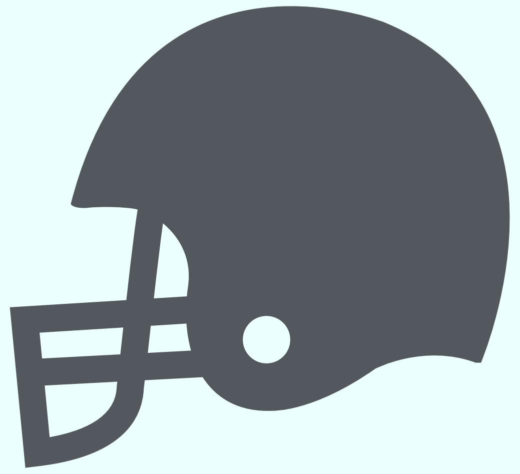
- **Kepler's Conjecture (1998):** what is the best way to pack spheres in a cube?
- **Rubik's Cube (2010):** The maximal number of face turns needed to solve any instance of the Rubik's Cube is **20**.
- **Sudoku (2012):** the minimum number of clues needed for there to be a unique solution is **17**.
- **Keller's Conjecture (2020):** in dimensions less than 7 in any tiling of equal "cubes", some must share a face.

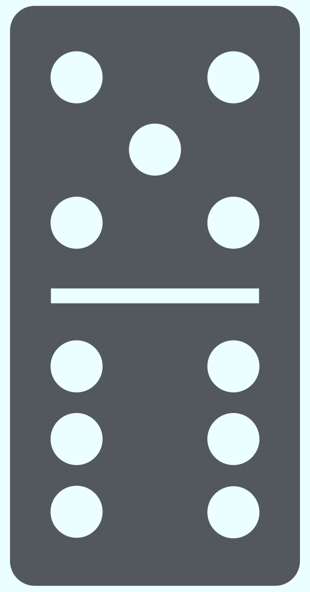


Computer-Assisted Proof

Strengths and Limitations

- Humans must develop the mathematical theory.
- Humans discover and implement algorithms,
- and “prove” that the output of the algorithm solves the problem.
- Uses computers as an aid, but humans still must lay the groundwork.
- These examples involve computations, not necessarily mathematical reasoning.





Ok, so computers are good at "computing" but this is not discovery...

Computation meets Mathematics

Part II: Computationally Formalizing Axiomatic Systems

Back at Peano axioms

for the natural numbers $\mathbb{N} = \{0,1,2,3,4,5\ldots\}$

- Zero if a natural number: $0 \in \mathbb{N}$
- Successor ia natural number: $\forall n \in \mathbb{N}, S(n) \in \mathbb{N}$
- Zero is not the successor of any number: $\forall n \in \mathbb{N}, S(n) \neq 0$
- Successor function is injective: $\forall m, n \in \mathbb{N}, S(m) = S(n) \Rightarrow m = n$
- Induction axiom: for any property $P(n)$, if
  - $P(0)$ holds, and
  - $\forall n \in \mathbb{N}, P(n) \Rightarrow P(S(n)),$ then
  - $\forall n \in \mathbb{N}, P(n)$

Can we encode this in a computer program?

Peano axioms (in Lean 4)

for the natural numbers $\mathbb{N} = \{0,1,2,3,4,5\ldots\}$

- Zero if a natural number: $0 \in \mathbb{N}$
- Successor ia natural number: $\forall n \in \mathbb{N}, S(n) \in \mathbb{N}$
- Zero is not the successor of any number: $\forall n \in \mathbb{N}, S(n) \neq 0$
- Successor function is injective: $\forall m, n \in \mathbb{N}, S(m) = S(n) \Rightarrow m = n$
- Induction axiom: for any property $P(n)$, if
  - $P(0)$ holds, and
  - $\forall n \in \mathbb{N}, P(n) \Rightarrow P(S(n))$, then
  - $\forall n \in \mathbb{N}, P(n)$

```
inductive Nat : Type
| zero : Nat
| succ : Nat → Nat
```

```
def Nat.add : Nat → Nat → Nat
| n, Nat.zero   => n                          -- n + 0 = n
| n, Nat.succ m => Nat.succ (Nat.add n m) -- n + succ(m) = succ(n + m)
```

What is the sum of all natural numbers smaller than or equal to 100?

5050	0%
5000	0%
10000	0%
4950	0%
None of the above	0%

Peano axioms (in Lean 4)

for the natural numbers $\mathbb{N} = \{0,1,2,3,4,5\ldots\}$

Proving that $1 + 2 + 3 + \cdots + n = \dfrac{n(n+1)}{2}$ in Lean requires:

```
import Mathlib.Data.Nat.Basic
import Mathlib.Tactic

open Nat

-- Theorem: The sum of the first n natural numbers is n(n+1)/2
theorem sum_nat_numbers (n : ℕ) :
  2 * (∑ k in Finset.range (n + 1), k) = n * (n + 1) := by
  induction n with
  | zero =>
    simp [Finset.range, Finset.sum_empty]
  | succ n ih =>
    simp only [Finset.range_succ, Finset.sum_insert Finset.not_mem_range_self]
    rw [mul_add, mul_add, mul_one, ih]
    ring
```

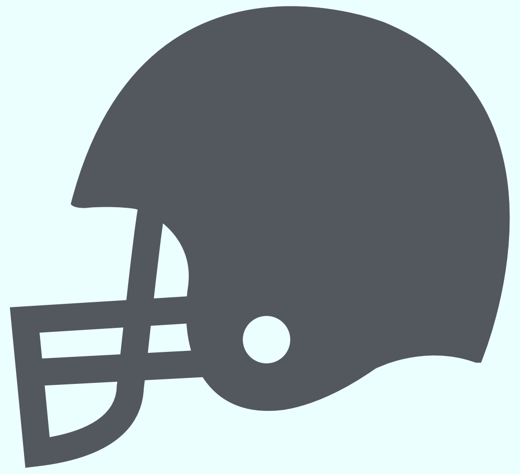
Proof Assistant Programming

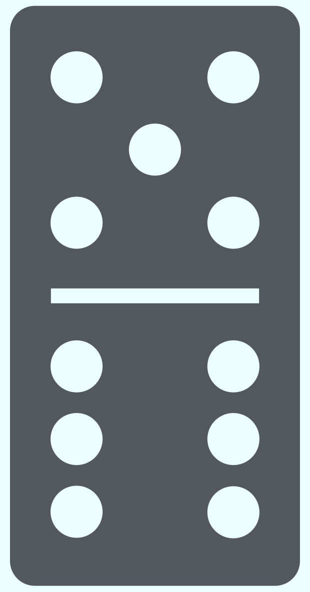
- Some commonly used languages include:
  - Lean
  - Coq
  - Isabelle/HOL
- They help with
  - Formalizing Mathematical Statements and Proofs
  - Checking Proofs for Correctness
  - Assisting with Proof Development
  - Generating Code from Proofs

Formal Programming Languages

What do they have in common?

- Computer can now use axioms, and verify proofs of theorems.
- This still requires humans to decide which statements to proof and
- write proofs ourselves.





Computers can replicate math, but humans need to hard code it still.

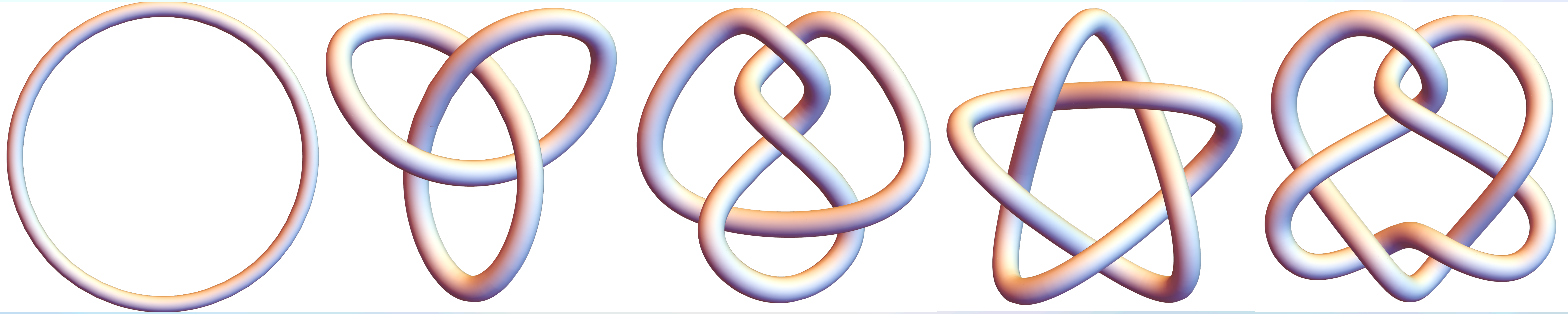
Computation meets Mathematics

Part III: The Statistics and Machine Learning Approach

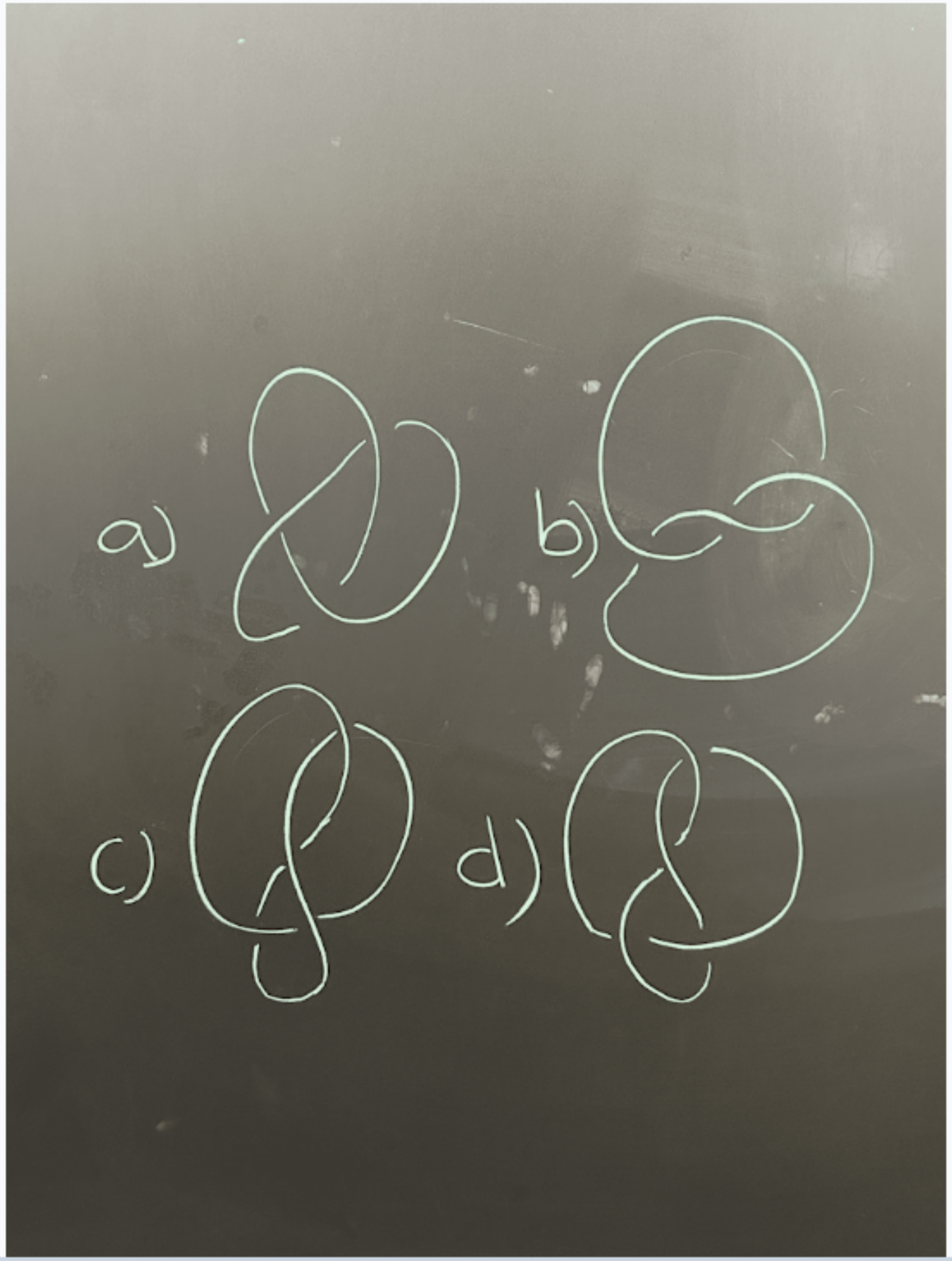
Data Treatment

Finding Empirical Relations in Knot Theory

- Knot theory is a branch of mathematics focused on the study of knots, which are closed loops embedded in three-dimensional space.



Which of these knots is different?



A

0%

B

0%

C

0%

D

0%

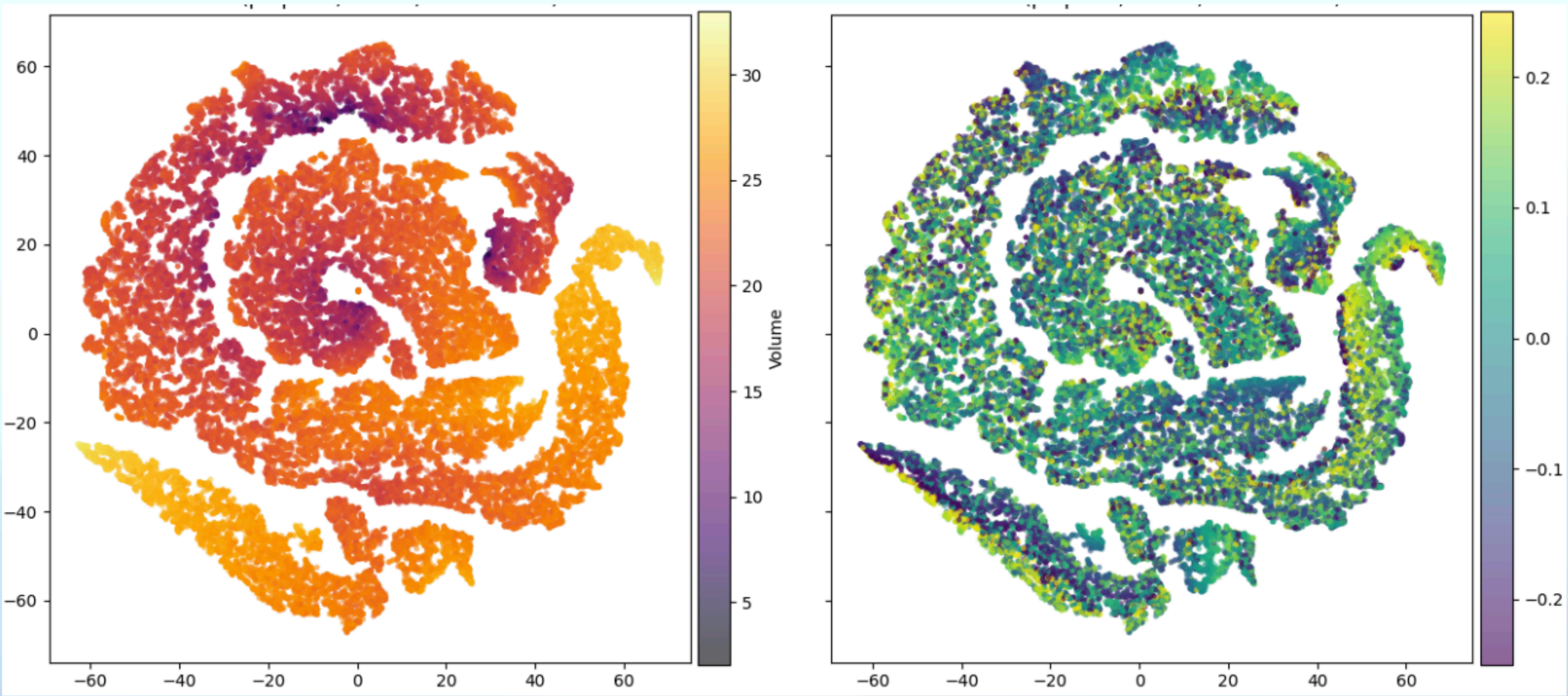
Data Treatment

Finding Empirical Relations

- Knots have many properties: crossing number, bridge number, stick number, tunner number, Alexander polynomial, A-polynomial, Jones polynomial, HOMFLY polynomial, genus, tricolorability, signature, Khovanov homology, …
- How can one discover new relationships between these properties?
- Some options:
  - Data visualization
  - Regression/Classification algorithms

Discovering New Relationships

Data Treatment and Visualization

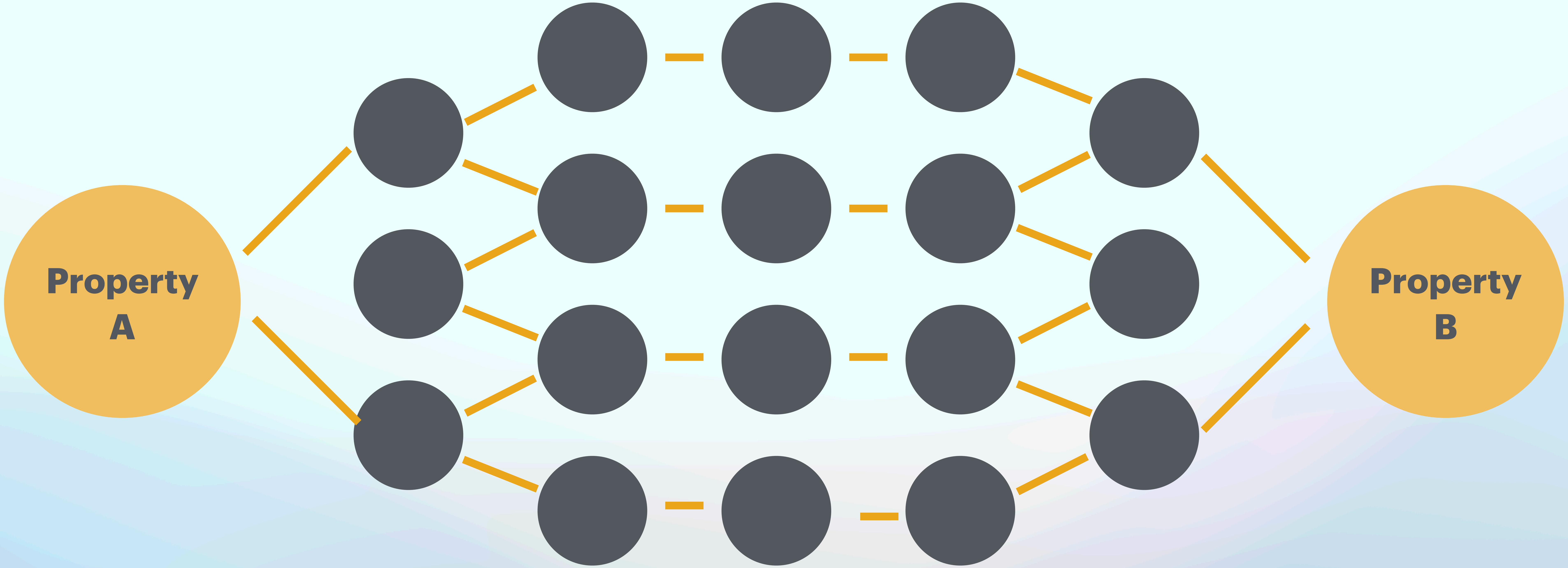


Property A

Property B

Discovering New Relationships

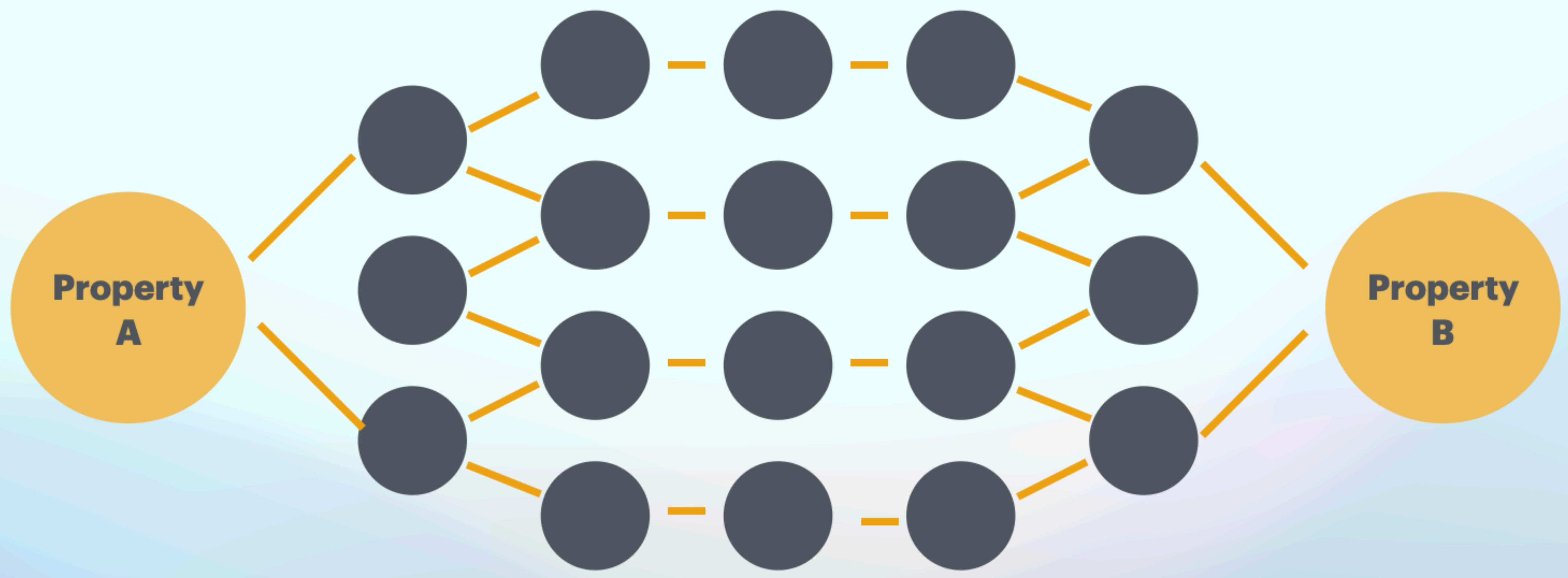
Neural Nets, Random Forests, etc…



Discovering New Relationships

Neural Nets, Random Forests, etc...

- If the neural network can succesfully predict **Property B** from **Property A**, there may be some relationship between the two!

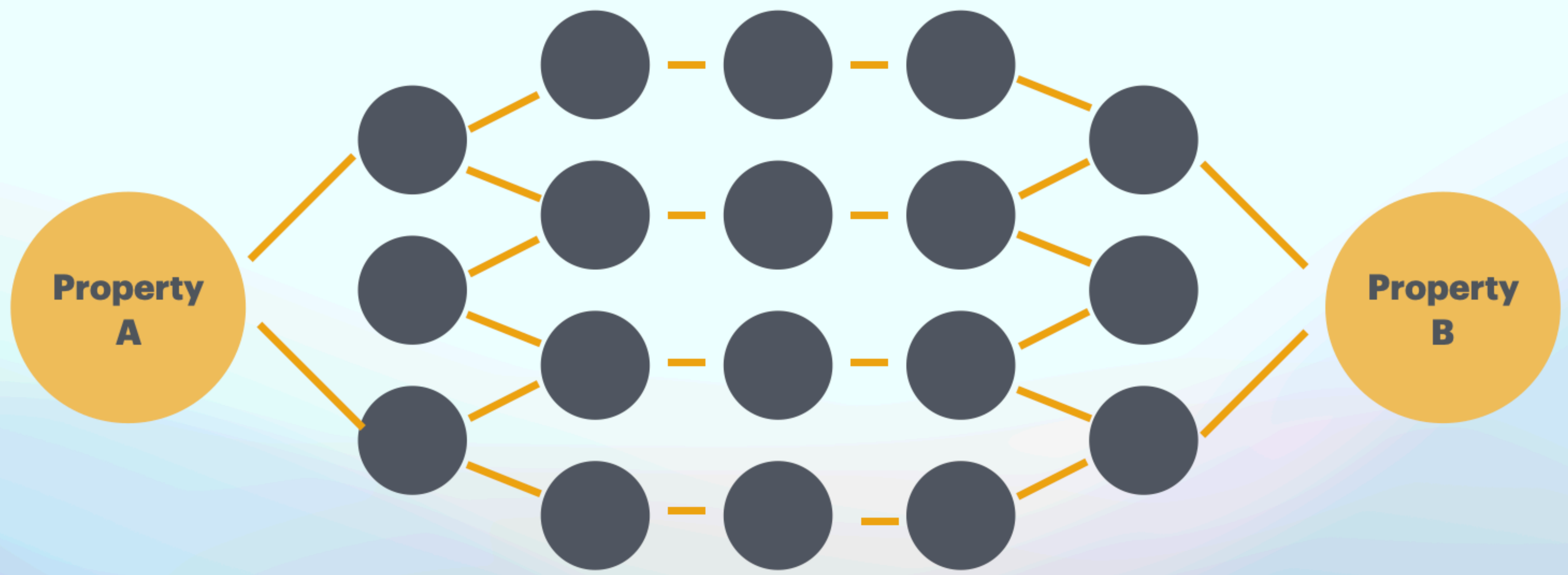


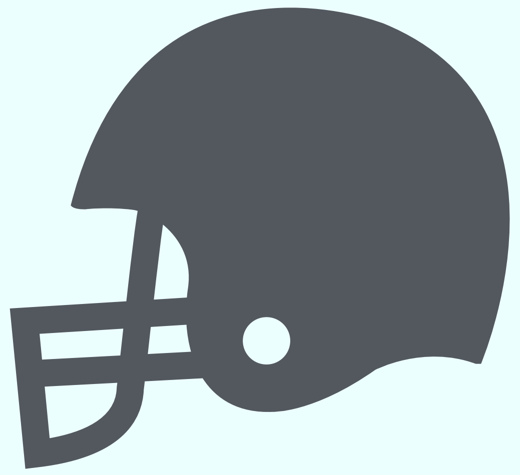
Discovering New Relationships

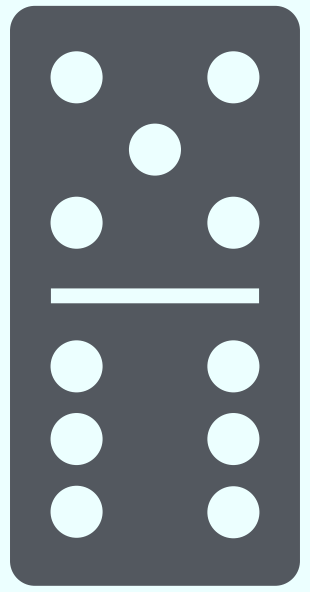
Neural Nets, Random Forests, etc...

- According to the **Universal approximation theorem,** a feedforward neural network with a single hidden layer containing a finite number of neurons can approximate any continuous function to any desired degree of accuracy.

- How does one find the actual function???







Machine Learning Gets Good Results, But May Lack Mathematical Meaning

Computation meets Mathematics

Part IV: Combining it All

Generative AI

- Generative AI refers to AI systems that can **create new content**, like text, images, audio, and videos, based on learned patterns from existing data. It's a type of AI that goes beyond analyzing and interpreting data to generate something entirely new, unlike traditional AI which focuses on tasks like prediction and classification within defined boundaries.

Matrix Multiplication

Not All New Developments are “New”

$$B = \begin{bmatrix} 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \quad \begin{bmatrix} 30 & 24 & 18 \\ 84 & 69 & 54 \\ 138 & 114 & 90 \end{bmatrix} = AB$$

$$84 = 4 \times 9 + 5 \times 6 + 6 \times 3$$

In order to multiply two 2×2 matrices, one needs to perform how many multiplications?

6	0%
7	0%
8	0%
9	0%

In order to multiply two $n \times n$ matrices, one needs to perform how many multiplications?

n^2

0%

n

0%

n^3

0%

It is unknown

0%

Matrix Multiplication

Strassen Algorithm

Start with two matrices $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \quad B = \begin{bmatrix} e & f \\ g & h \end{bmatrix}.$

Define the quantities

$$
\begin{aligned}
M_1 &= (a+d)(e+h) \\
M_2 &= (c+d)e \\
M_3 &= a(f-h) \\
M_4 &= d(g-e) \\
M_5 &= (a+b)h \\
M_6 &= (c-a)(e+f) \\
M_7 &= (b-d)(g+h)
\end{aligned}
$$

The result is thus:

$$C = \begin{bmatrix} M_1 + M_4 - M_5 + M_7 & M_3 + M_5 \\ M_2 + M_4 & M_1 - M_2 + M_3 + M_6 \end{bmatrix}$$

This requires “only” 7 multiplications!!!

Matrix Multiplication

New Frontiers

Year	Reference	#matrix multiplications per step	#matrix additions per step	total arithmetic operations
1969	Strassen[12]	7	18	$7n^{\log_2 7} - 6n^2$
1971	Winograd[13]	7	15	$6n^{\log_2 7} - 5n^2$
2017	Karstadt, Schwartz[14]	7	12	$5n^{\log_2 7} - 4n^2 + 3n^2 \log_2 n$
2023	Schwartz, Vaknin[15]	7	12	$5n^{\log_2 7} - 4n^2 + 1.5n^2 \log_2 n$

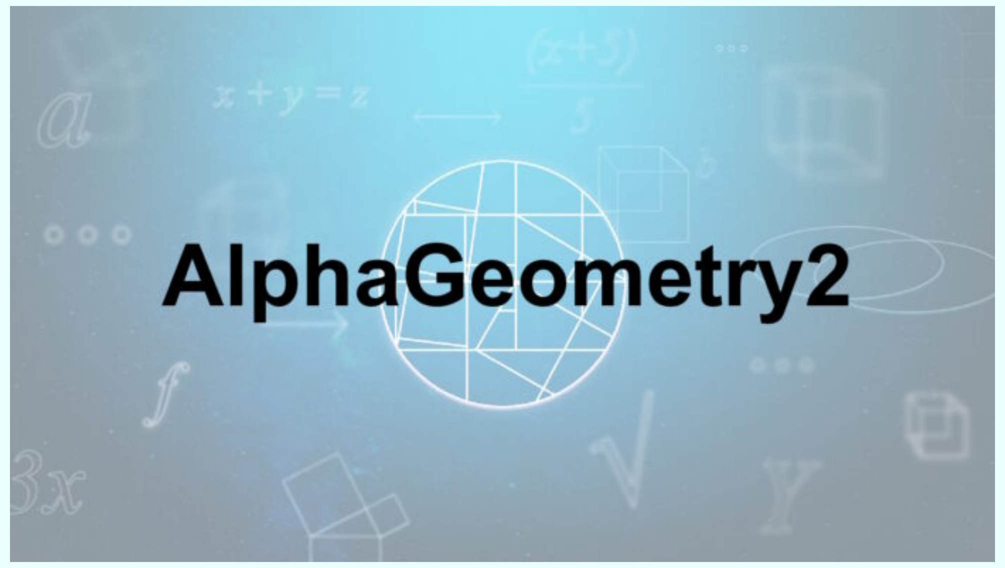
Matrix Multiplication

New Developments

- DeepMind's AlphaTensor (2022): On 4×4 matrices, AlphaTensor unexpectedly discovered a solution with 47 multiplication steps, an improvement over the 49 required with Strassen's algorithm of 1969, albeit restricted to mod 2 arithmetic. Similarly, AlphaTensor solved 5×5 matrices with 96 rather than Strassen's 98 steps. Based on the surprising discovery that such improvements exist, other researchers were quickly able to find a similar independent 4×4 algorithm, and separately tweaked Deepmind's 96-step 5×5 algorithm down to 95 steps in mod 2 arithmetic and to 97 [24] in normal arithmetic.

Other Examples

AlphaGeometry2



Here We present AlphaGeometry2 [...] which has now surpassed an average gold medalist in solving Olympiad geometry problems. To achieve this, we first extend the original AlphaGeometry language to tackle harder problems involving movements of objects, and problems containing linear equations of angles, ratios, and distances. This, together with support for non-constructive problems, has markedly improved the coverage rate of the AlphaGeometry language on International Math Olympiads (IMO) 2000-2024 geometry problems from 66% to 88%. The search process of AlphaGeometry2 has also been greatly improved through the use of Gemini architecture for better language modeling, and a novel knowledge-sharing mechanism that enables effective communication between search trees. Together with further enhancements to the symbolic engine and synthetic data generation,[...]

Other Examples

FunSearch

“Here we introduce FunSearch (short for searching in the function space), an evolutionary procedure based on pairing a pretrained LLM with a systematic evaluator. We demonstrate the effectiveness of this approach to surpass the best-known results in important problems, pushing the boundary of existing LLM-based approaches”

Can Generative AI Create New Math?

The End

A Summary

1. **Generative AI models, including large language models (LLMs) like GPT, can:**
   - Formulate new conjectures by recognizing patterns in mathematical data or literature.
   - Suggest plausible statements or generalizations that resemble known theorems.
2. **AI can assist in proof construction, especially in domains that are well-specified (such as logic, group theory, or real analysis) and where formal languages (like Lean, Coq, or Isabelle) are used.**

- Lean and Meta AI's GPT-f (2023) showed that LLMs trained on formal proofs can generate new formal derivations, verified by proof assistants.
- Benchmarks like MiniF2F test LLMs on mathematical proof tasks and have shown that AI can solve nontrivial problems. However, autonomously proving new theorems is still rare. AI typically relies on human input to define the context, guide strategies, and validate rigor. Proofs generated in natural language are often incomplete or flawed unless formalized rigorously.

3. **Limitations**

- It lacks deep mathematical understanding and can produce logically invalid proofs.
- Maintaining formal rigor is challenging without integration with symbolic systems.
- Inventing entirely new mathematical frameworks or techniques remains a human domain.

4. **What AI Is Already Doing in Mathematics**

- Conjecture discovery via empirical testing and pattern recognition.
- Generating lemmas or useful intermediate results.
- Filling in routine steps of formal proofs.
- Translating informal math to formal code and vice versa.
- Visualizing mathematical structures to aid exploration.