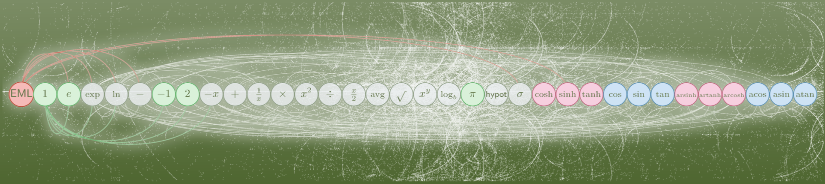
# All elementary functions from a single binary operator


Andrzej Odrzywołek

Jagiellonian University in Kraków

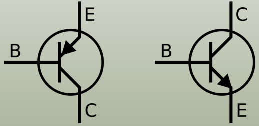

22 May 2026, 11:25 – 12:15

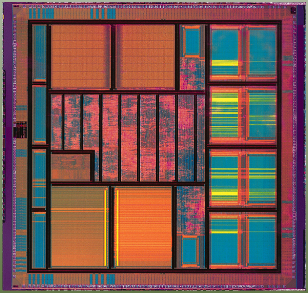
Kleine Lokhalle, Motoworld München, Germany



# Reusable elements are the core of scientific world model

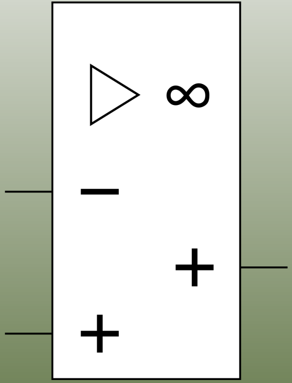
- transistors for digital and analog electronics
- operational amplifier for processes with negative and positive feedback loops
- activation functions in artificial neural networks (ReLU, sigmoid)
- DNA: base-4 CGTA carrier of genetic information ...
- ... encoded in triples to form 64 codes for 20 amino acids and STOP codon, forming molecular machinery for all life.
- single-axiom math ($\iota$ combinator) or single-instruction computers (SUBLEQ, Rule 110 cellular automaton)
- geometrical tilings (ein-stein, igloo brick)
- last but not least: the NAND gate.



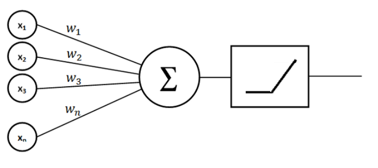
# Reusable elements are the core of scientific world model

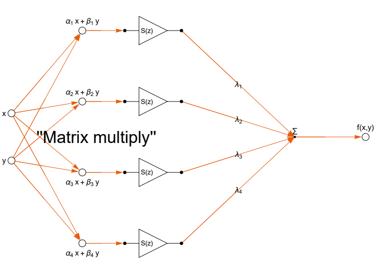
- transistors for digital and analog electronics
- operational amplifier for processes with negative and positive feedback loops
- activation functions in artificial neural networks (ReLU, sigmoid)
- DNA: base-4 CGTA carrier of genetic information ...
- ...encoded in triples to form 64 codes for 20 amino acids and STOP codon, forming molecular machinery for all life.
- single-axiom math ($\iota$ combinator) or single-instruction computers (SUBLEQ, Rule 110 cellular automaton)
- geometrical tilings (ein-stein, igloo brick)
- last but not least: the NAND gate.



# Reusable elements are the core of scientific world model

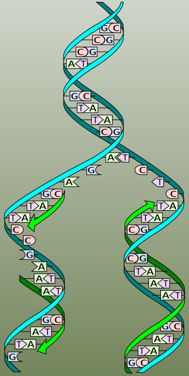
- transistors for digital and analog electronics
- operational amplifier for processes with negative and positive feedback loops
- activation functions in artificial neural networks (ReLU, sigmoid)
- DNA: base-4 CGTA carrier of genetic information . . .
- . . . encoded in triples to form 64 codes for 20 amino acids and STOP codon, forming molecular machinery for all life.
- single-axiom math ($\iota$ combinator) or single-instruction computers (SUBLEQ, Rule 110 cellular automaton)
- geometrical tilings (ein-stein, igloo brick)
- last but not least: the NAND gate.



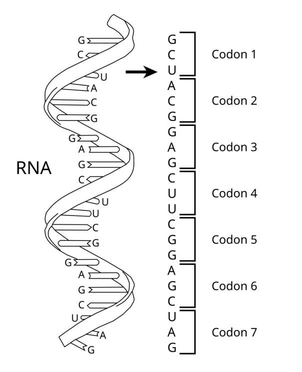
# Reusable elements are the core of scientific world model

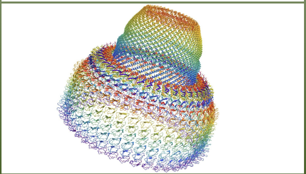
- transistors for digital and analog electronics
- operational amplifier for processes with negative and positive feedback loops
- activation functions in artificial neural networks (ReLU, sigmoid)
- DNA: base-4 CGTA carrier of genetic information . . .
- . . . encoded in triples to form 64 codes for 20 amino acids and STOP codon, forming molecular machinery for all life.
- single-axiom math ($\iota$ combinator) or single-instruction computers (SUBLEQ, Rule 110 cellular automaton)
- geometrical tilings (ein-stein, igloo brick)
- last but not least: the NAND gate.



# Reusable elements are the core of scientific world model

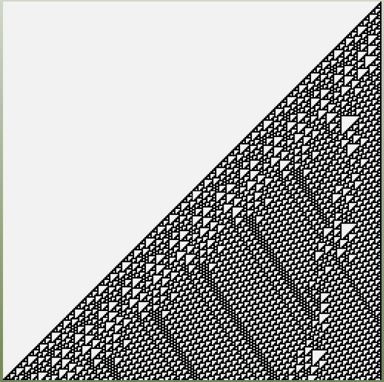
- transistors for digital and analog electronics
- operational amplifier for processes with negative and positive feedback loops
- activation functions in artificial neural networks (ReLU, sigmoid)
- DNA: base-4 CGTA carrier of genetic information ...
- ...encoded in triples to form 64 codes for 20 amino acids and STOP codon, forming molecular machinery for all life.
- single-axiom math ($\iota$ combinator) or single-instruction computers (SUBLEQ, Rule 110 cellular automaton)
- geometrical tilings (ein-stein, igloo brick)
- last but not least: the NAND gate.



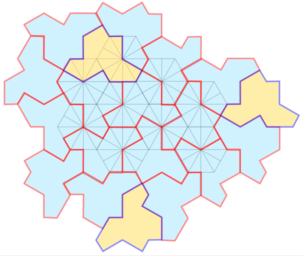
# Reusable elements are the core of scientific world model

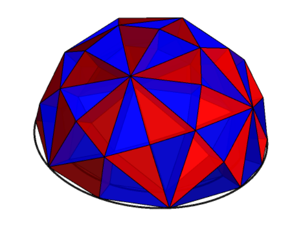
- transistors for digital and analog electronics
- operational amplifier for processes with negative and positive feedback loops
- activation functions in artificial neural networks (ReLU, sigmoid)
- DNA: base-4 CGTA carrier of genetic information …
- …encoded in triples to form 64 codes for 20 amino acids and STOP codon, forming molecular machinery for all life.
- single-axiom math ($\iota$ combinator) or single-instruction computers (SUBLEQ, Rule 110 cellular automaton)
- geometrical tilings (ein-stein, igloo brick)
- last but not least: the NAND gate.



# Reusable elements are the core of scientific world model

- transistors for digital and analog electronics
- operational amplifier for processes with negative and positive feedback loops
- activation functions in artificial neural networks (ReLU, sigmoid)
- DNA: base-4 CGTA carrier of genetic information ...
- ... encoded in triples to form 64 codes for 20 amino acids and STOP codon, forming molecular machinery for all life.
- single-axiom math ($\iota$ combinator) or single-instruction computers (SUBLEQ, Rule 110 cellular automaton)
- geometrical tilings (ein-stein, igloo brick)
- last but not least: the NAND gate.





# Reusable elements are the core of scientific world model

- transistors for digital and analog electronics
- operational amplifier for processes with negative and positive feedback loops
- activation functions in artificial neural networks (ReLU, sigmoid)
- DNA: base-4 CGTA carrier of genetic information ...
- ...encoded in triples to form 64 codes for 20 amino acids and STOP codon, forming molecular machinery for all life.
- single-axiom math ($\iota$ combinator) or single-instruction computers (SUBLEQ, Rule 110 cellular automaton)
- geometrical tilings (ein-stein, igloo brick)
- last but not least: the NAND gate.


**Q** = **A** NAND **B**

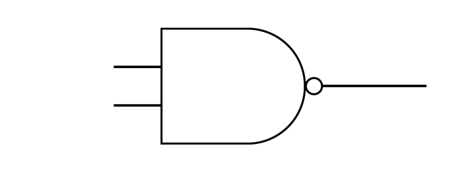
**Truth Table**

| Input A | Input B | Output Q |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

# NAND gate and functional completeness

**Using binary NAND operator we can compute any boolean function.**

- Sheffer (1913),
  Shannon (1938)
- logical negation NOT
- logical AND
- logical OR
- Generate 0 and 1
- Add two 4-bit integers in binary form
- Build supercomputer!

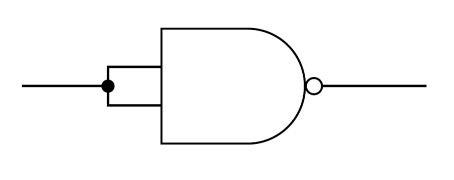


For someone from the XIX century the above would look like a ridiculous method to, e.g., add two small integers. Yet, in XXI, this is the ONLY method. Cray-1 (1976 supercomputer) CPU was made mostly from ECL NAND gates!

# NAND gate and functional completeness

**Using binary NAND operator we can compute any boolean function.**

- Sheffer (1913), Shannon (1938)
- logical negation NOT
- logical AND
- logical OR
- Generate 0 and 1
- Add two 4-bit integers in binary form
- Build supercomputer!

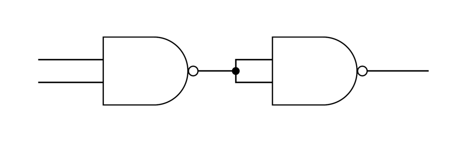


For someone from the XIX century the above would look like a ridiculous method to, e.g., add two small integers. Yet, in XXI, this is the ONLY method. Cray-1 (1976 supercomputer) CPU was made mostly from ECL NAND gates!

# NAND gate and functional completeness

**Using binary NAND operator we can compute any boolean function.**

- Sheffer (1913), Shannon (1938)
- logical negation NOT
- logical AND
- logical OR
- Generate 0 and 1
- Add two 4-bit integers in binary form
- Build supercomputer!

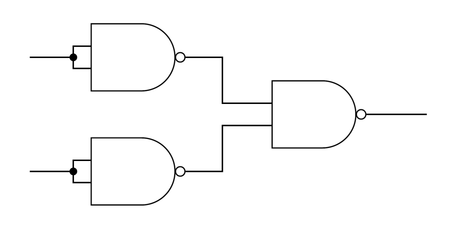


For someone from the XIX century the above would look like a ridiculous method to, e.g., add two small integers. Yet, in XXI, this is the ONLY method. Cray-1 (1976 supercomputer) CPU was made mostly from ECL NAND gates!

# NAND gate and functional completeness

## Using binary NAND operator we can compute any boolean function.

- Sheffer (1913),
  Shannon (1938)
- logical negation NOT
- logical AND
- logical OR
- Generate 0 and 1
- Add two 4-bit integers in binary form
- Build supercomputer!

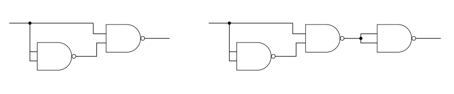


For someone from the XIX century the above would look like a ridiculous method to, e.g., add two small integers. Yet, in XXI, this is the ONLY method. Cray-1 (1976 supercomputer) CPU was made mostly from ECL NAND gates!

# NAND gate and functional completeness

## Using binary NAND operator we can compute any boolean function.

- Sheffer (1913), Shannon (1938)
- logical negation NOT
- logical AND
- logical OR
- Generate 0 and 1
- Add two 4-bit integers in binary form
- Build supercomputer!

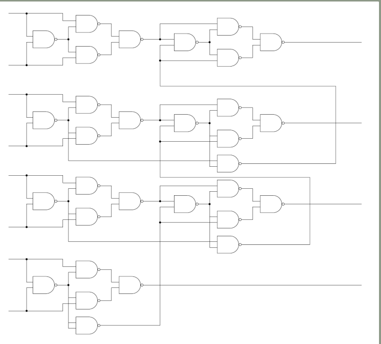


For someone from the XIX century the above would look like a ridiculous method to, e.g., add two small integers. Yet, in XXI, this is the ONLY method. Cray-1 (1976 supercomputer) CPU was made mostly from ECL NAND gates!

# NAND gate and functional completeness

## Using binary NAND operator we can compute any boolean function.

- Sheffer (1913), Shannon (1938)
- logical negation NOT
- logical AND
- logical OR
- Generate 0 and 1
- Add two 4-bit integers in binary form
- Build supercomputer!

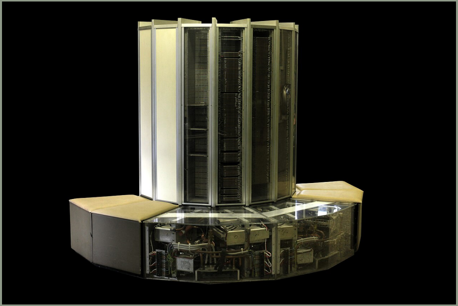


For someone from the XIX century the above would look like a ridiculous method to, e.g., add two small integers. Yet, in XXI, this is the ONLY method. Cray-1 (1976 supercomputer) CPU was made mostly from ECL NAND gates!

# NAND gate and functional completeness

## Using binary NAND operator we can compute any boolean function.

- Sheffer (1913),
  Shannon (1938)
- logical negation NOT
- logical AND
- logical OR
- Generate 0 and 1
- Add two 4-bit integers in binary form
- Build supercomputer!
  `(200k NANDs)`



For someone from the XIX century the above would look like a ridiculous method to, e.g., add two small integers. Yet, in XXI, this is the ONLY method. Cray-1 (1976 supercomputer) CPU was made mostly from ECL NAND gates!

# NAND gate and functional completeness

**Using binary NAND operator we can compute any boolean function.**

- Sheffer (1913),
  Shannon (1938)
- logical negation NOT
- logical AND
- logical OR
- Generate 0 and 1
- Add two 4-bit integers in binary form
- Build supercomputer! `(30M soldiers)`


For someone from the XIX century the above would look like a ridiculous method to, e.g., add two small integers. Yet, in XXI, this is the ONLY method. Cray-1 (1976 supercomputer) CPU was made mostly from ECL NAND gates!

# Importance of elementary functions in STEM

- quantitative science is about numbers, we must evaluate expressions: $(\frac{1}{x}+\frac{1}{y})^{-1}$, $\frac{-b+\sqrt{b^2-4ac}}{2a}$, $\frac{v^2}{g}\sin 2\alpha$, $\frac{1}{\sqrt{\pi}}e^{-x^2}$, $\sqrt{2}$, $p\log p$, etc.
- exact elementary formulas are everywhere: fundamental theories, empirical laws, programming languages, education, engineering
- majority of numerical algorithms are cast in explicit math form, e.g:

1. Gaussian quadratures

$$\int_{-1}^{1} f(x)dx = \frac{5}{9}f(-\sqrt{3/5}) + \frac{8}{9}f(0) + \frac{5}{9}f(\sqrt{3/5})$$

2. Runge-Kutta methods

$$y_{n+1} = y_n + hf[x_n + \frac{h}{2}, y_n + \frac{h}{2}f(x_n, y_n)]$$

3. Discrete Fourier Transform (FFT)

$$X_k = x_0 + x_1 e^{-\frac{2\pi i}{N}} + x_2 e^{-\frac{4\pi i}{N}} + \ldots + x_{n-1} e^{-\frac{2(N-1)\pi i}{N}}$$

# How far can we reduce elementary functions?

- Historically, we started with counting; addition, multiplication and fractions are fundamental since antiquity
- Spherical and planar trigonometry required to measure heavens and land; Tycho de Brahe was using *Prosthaphaeresis* to multiply $\cos\alpha\cos\beta = \frac{1}{2}[\cos(\alpha+\beta) + \cos(\alpha-\beta)]$
- Logarithms (Napier, Briggs, Kepler) were "killer app" that transformed computations in a decade 1610-1620
  1. multiplication became addition: $\log(x \times y) = \log(x) + \log(y)$
  2. division became subtraction: $\log(x/y) = \log(x) - \log(y)$
  3. powers became multiplication: $\log(x^y) = \log(x) \times y$
  4. roots became division: $\log(\sqrt[y]{x}) = \log(x)/y$
- Imaginary unit $i = \sqrt{-1}$ introduced (Cardano, 1545) to solve cubic equations reduced trigonometry to complex arithmetic via Euler's formula

$$e^{i\varphi} = \cos\varphi + i\sin\varphi$$

- Elementary functions reduced to $\exp, \ln$ and arithmetic $(+, -, \times, /)$ for integration in finite terms (Liouville 1835, Ritt 1925+, Risch 1969)
- Slide rule forgotten due to rise of electronic calculators in 70s ...
  ... which were replaced by software in late 90s.

*Can we push this reduction forward, into ML-friendly form?*

# Broken calculator

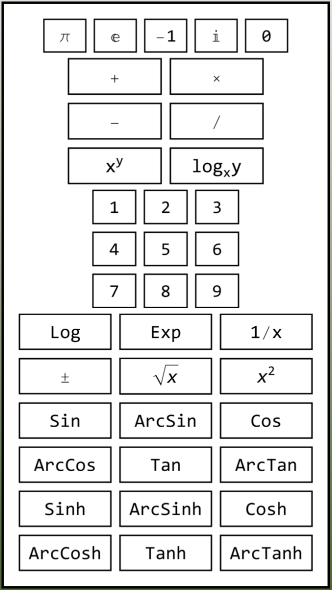
Simple mental model of the problem we are solving here is a malfunctioning calculator with missing or damaged buttons.


**Would CASIO still work with multiplication sign '×' removed?**

YES: $x \times y = 1/[(1/x)/y]$.

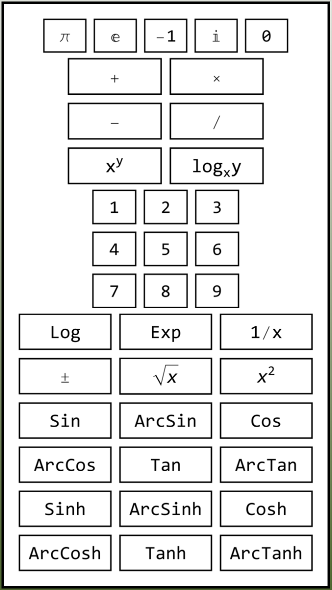
$$12 \times 7 = 84 \qquad 1 \div 7 = \div 12 = \div = 84.00436$$

What if we remove digit '1' as well? It quickly becomes complicated...



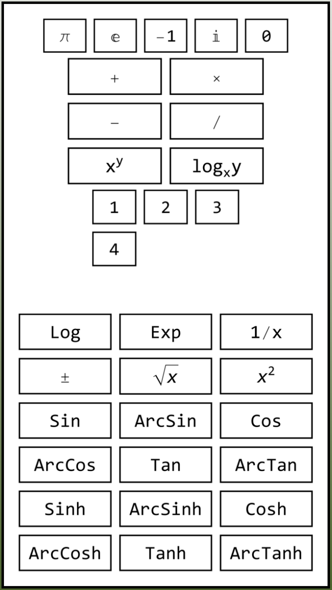

- $9 = 3^2, 8 = 2^3, 7 = 3 + 4, 6 = 2 \times 3, 5 = 2 + 3$; OUT!
- $4 = 2 + 2, 3 = 1 + 2$; OUT!
- $\tan x = \frac{\sin x}{\cos x}, \tanh x = \frac{\sinh x}{\cosh x}$; OUT!
- $\cos x = \sin\left(x + \frac{\pi}{2}\right), \arcsin x = \frac{\pi}{2} - \arccos x$; OUT!
- $\cosh x = \frac{e^x + e^{-x}}{2}, \sinh x = \frac{e^x - e^{-x}}{2}$; OUT!
- $\operatorname{arsinh} x = \ln\left(1 + \sqrt{1 + x^2}\right)$, $\operatorname{arcosh} x = \ln\left(x + \sqrt{x+1}\sqrt{x-1}\right)$, $\operatorname{artanh} x = \frac{\ln(1+x) - \ln(1-x)}{2}$; OUT!
- $-x = 0 - x, 0 = 1 - 1$; OUT!

# Reduction of *'bloated'* scientific calculator



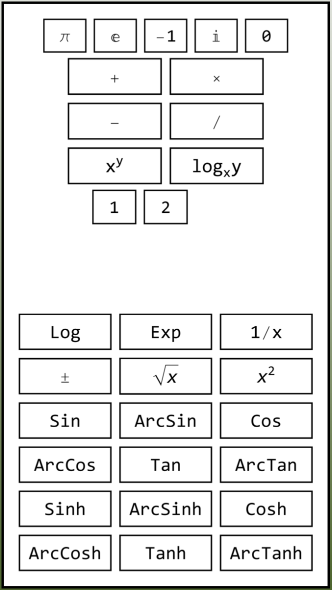

- $9 = 3^2, 8 = 2^3, 7 = 3 + 4, 6 = 2 \times 3, 5 = 2 + 3$; OUT!
- $4 = 2 + 2, 3 = 1 + 2$; OUT!
- $\tan x = \frac{\sin x}{\cos x}, \tanh x = \frac{\sinh x}{\cosh x}$; OUT!
- $\cos x = \sin\left(x + \frac{\pi}{2}\right), \arcsin x = \frac{\pi}{2} - \arccos x$; OUT!
- $\cosh x = \frac{e^x + e^{-x}}{2}, \sinh x = \frac{e^x - e^{-x}}{2}$; OUT!
- $\operatorname{arsinh} x = \ln\left(1 + \sqrt{1 + x^2}\right)$, $\operatorname{arcosh} x = \ln\left(x + \sqrt{x+1}\sqrt{x-1}\right)$, $\operatorname{artanh} x = \frac{\ln(1+x) - \ln(1-x)}{2}$; OUT!
- $-x = 0 - x, 0 = 1 - 1$; OUT!

# Reduction of *'bloated'* scientific calculator



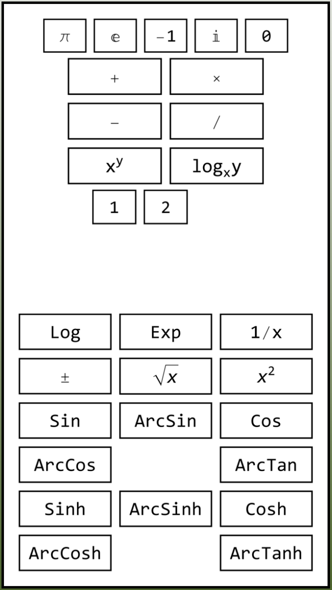

- $9 = 3^2, 8 = 2^3, 7 = 3 + 4, 6 = 2 \times 3, 5 = 2 + 3$; OUT!
- $4 = 2 + 2, 3 = 1 + 2$; OUT!
- $\tan x = \frac{\sin x}{\cos x}, \tanh x = \frac{\sinh x}{\cosh x}$; OUT!
- $\cos x = \sin\left(x + \frac{\pi}{2}\right), \arcsin x = \frac{\pi}{2} - \arccos x$; OUT!
- $\cosh x = \frac{e^x + e^{-x}}{2}, \sinh x = \frac{e^x - e^{-x}}{2}$; OUT!
- $\operatorname{arsinh} x = \ln\left(1 + \sqrt{1 + x^2}\right)$, $\operatorname{arcosh} x = \ln\left(x + \sqrt{x + 1}\sqrt{x - 1}\right)$, $\operatorname{artanh} x = \frac{\ln(1+x) - \ln(1-x)}{2}$; OUT!
- $-x = 0 - x, 0 = 1 - 1$; OUT!

# Reduction of *'bloated'* scientific calculator



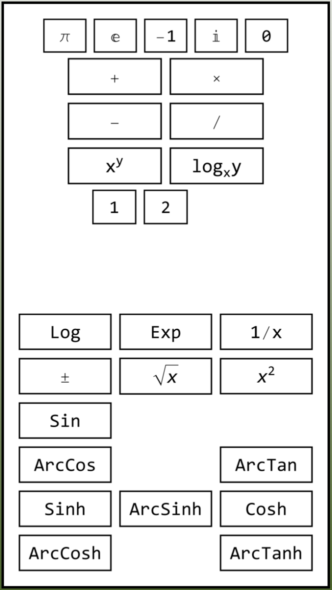

- $9 = 3^2, 8 = 2^3, 7 = 3 + 4, 6 = 2 \times 3, 5 = 2 + 3$; OUT!
- $4 = 2 + 2, 3 = 1 + 2$; OUT!
- $\tan x = \frac{\sin x}{\cos x}, \tanh x = \frac{\sinh x}{\cosh x}$; OUT!
- $\cos x = \sin\left(x + \frac{\pi}{2}\right), \arcsin x = \frac{\pi}{2} - \arccos x$; OUT!
- $\cosh x = \frac{e^x + e^{-x}}{2}, \sinh x = \frac{e^x - e^{-x}}{2}$; OUT!
- $\operatorname{arsinh} x = \ln\left(1 + \sqrt{1 + x^2}\right)$, $\operatorname{arcosh} x = \ln\left(x + \sqrt{x+1}\sqrt{x-1}\right)$, $\operatorname{artanh} x = \frac{\ln(1+x) - \ln(1-x)}{2}$; OUT!
- $-x = 0 - x, 0 = 1 - 1$; OUT!

# Reduction of *'bloated'* scientific calculator



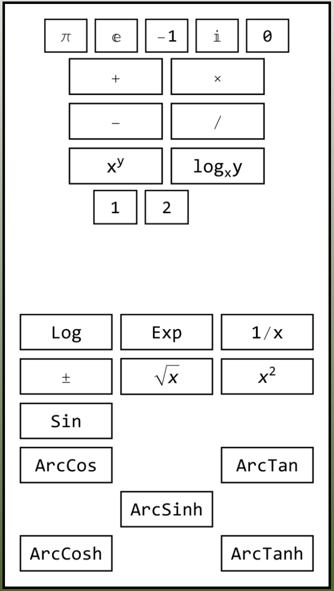

- $9 = 3^2, 8 = 2^3, 7 = 3 + 4, 6 = 2 \times 3, 5 = 2 + 3$; OUT!
- $4 = 2 + 2, 3 = 1 + 2$; OUT!
- $\tan x = \frac{\sin x}{\cos x}, \tanh x = \frac{\sinh x}{\cosh x}$; OUT!
- $\cos x = \sin\left(x + \frac{\pi}{2}\right), \arcsin x = \frac{\pi}{2} - \arccos x$; OUT!
- $\cosh x = \frac{e^x + e^{-x}}{2}, \sinh x = \frac{e^x - e^{-x}}{2}$; OUT!
- $\operatorname{arsinh} x = \ln\left(1 + \sqrt{1 + x^2}\right)$, $\operatorname{arcosh} x = \ln\left(x + \sqrt{x+1}\sqrt{x-1}\right)$, $\operatorname{artanh} x = \frac{\ln(1+x) - \ln(1-x)}{2}$; OUT!
- $-x = 0 - x, 0 = 1 - 1$; OUT!

# Reduction of *'bloated'* scientific calculator



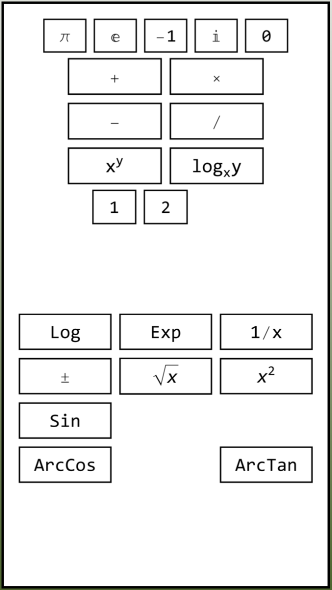

- $9 = 3^2, 8 = 2^3, 7 = 3 + 4, 6 = 2 \times 3, 5 = 2 + 3$; OUT!
- $4 = 2 + 2, 3 = 1 + 2$; OUT!
- $\tan x = \frac{\sin x}{\cos x}, \tanh x = \frac{\sinh x}{\cosh x}$; OUT!
- $\cos x = \sin\left(x + \frac{\pi}{2}\right), \arcsin x = \frac{\pi}{2} - \arccos x$; OUT!
- $\cosh x = \frac{e^x + e^{-x}}{2}, \sinh x = \frac{e^x - e^{-x}}{2}$; OUT!
- $\operatorname{arsinh} x = \ln\left(x + \sqrt{1 + x^2}\right)$, $\operatorname{arcosh} x = \ln\left(x + \sqrt{x+1}\sqrt{x-1}\right)$, $\operatorname{artanh} x = \frac{\ln(1+x) - \ln(1-x)}{2}$; OUT!
- $-x = 0 - x, 0 = 1 - 1$; OUT!

## Reduction of *'bloated'* scientific calculator



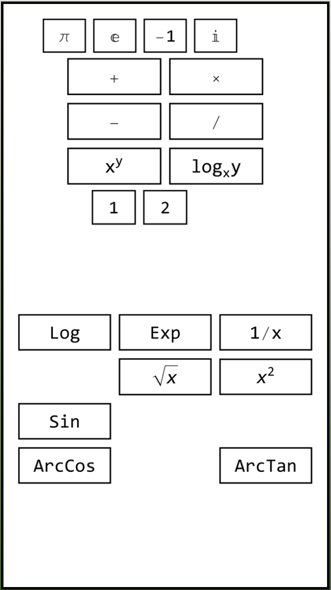

- $9 = 3^2, 8 = 2^3, 7 = 3 + 4, 6 = 2 \times 3, 5 = 2 + 3$; OUT!
- $4 = 2 + 2, 3 = 1 + 2$; OUT!
- $\tan x = \frac{\sin x}{\cos x}, \tanh x = \frac{\sinh x}{\cosh x}$; OUT!
- $\cos x = \sin\left(x + \frac{\pi}{2}\right), \arcsin x = \frac{\pi}{2} - \arccos x$; OUT!
- $\cosh x = \frac{e^x + e^{-x}}{2}, \sinh x = \frac{e^x - e^{-x}}{2}$; OUT!
- $\operatorname{arsinh} x = \ln\left(1 + \sqrt{1 + x^2}\right)$, $\operatorname{arcosh} x = \ln\left(x + \sqrt{x + 1}\sqrt{x - 1}\right)$, $\operatorname{artanh} x = \frac{\ln(1+x) - \ln(1-x)}{2}$; OUT!
- $-x = 0 - x, 0 = 1 - 1$; OUT!

# Reduction of *'bloated'* scientific calculator



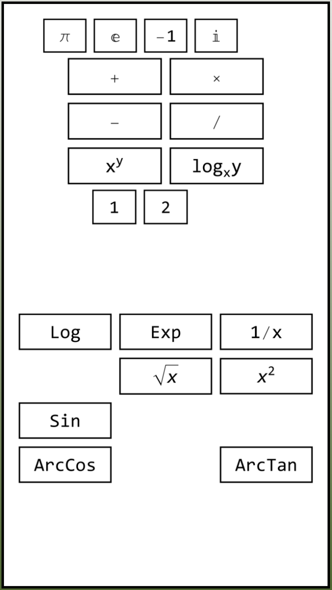

- $9 = 3^2, 8 = 2^3, 7 = 3 + 4, 6 = 2 \times 3, 5 = 2 + 3$; OUT!
- $4 = 2 + 2, 3 = 1 + 2$; OUT!
- $\tan x = \frac{\sin x}{\cos x}, \tanh x = \frac{\sinh x}{\cosh x}$; OUT!
- $\cos x = \sin\left(x + \frac{\pi}{2}\right), \arcsin x = \frac{\pi}{2} - \arccos x$; OUT!
- $\cosh x = \frac{e^x + e^{-x}}{2}, \sinh x = \frac{e^x - e^{-x}}{2}$; OUT!
- $\operatorname{arsinh} x = \ln\left(1 + \sqrt{1 + x^2}\right)$, $\operatorname{arcosh} x = \ln\left(x + \sqrt{x+1}\sqrt{x-1}\right)$, $\operatorname{artanh} x = \frac{\ln(1+x) - \ln(1-x)}{2}$; OUT!
- $-x = 0 - x, 0 = 1 - 1$; OUT!

# Reduction of *'bloated'* scientific calculator




- $9 = 3^2, 8 = 2^3, 7 = 3 + 4, 6 = 2 \times 3, 5 = 2 + 3$; OUT!
- $4 = 2 + 2, 3 = 1 + 2$; OUT!
- $\tan x = \frac{\sin x}{\cos x}, \tanh x = \frac{\sinh x}{\cosh x}$; OUT!
- $\cos x = \sin\left(x + \frac{\pi}{2}\right), \arcsin x = \frac{\pi}{2} - \arccos x$; OUT!
- $\cosh x = \frac{e^x + e^{-x}}{2}, \sinh x = \frac{e^x - e^{-x}}{2}$; OUT!
- $\operatorname{arsinh} x = \ln\left(1 + \sqrt{1 + x^2}\right)$, $\operatorname{arcosh} x = \ln\left(x + \sqrt{x + 1}\sqrt{x - 1}\right)$, $\operatorname{artanh} x = \frac{\ln(1+x) - \ln(1-x)}{2}$; OUT!
- $-x = 0 - x, 0 = 1 - 1$; OUT!

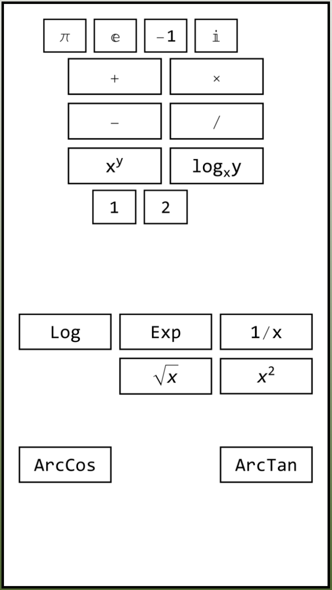
# Reduction of calculator - complex numbers




- $\sin x = \frac{e^{ix} - e^{-ix}}{2i}$; OUT!
- $\arccos x = \frac{\pi}{2} + i \ln\left(ix + \sqrt{1 - x^2}\right)$, $\arctan x = \frac{i}{2}\ln(1 - ix) - \frac{i}{2}\ln(1 + ix)$; OUT!
- $\pi = \frac{\ln(-1)}{i}, i = \sqrt{-1}$; OUT!
- $x^2 = x \times x, \sqrt{x} = x^{1/2}$; OUT!
- $\frac{1}{x} = x^{-1}, e = \exp(1)$; OUT!
- $\log_x y = \frac{\ln y}{\ln x}, x^y = \exp(y \ln x)$; OUT!
- $x + y = x - (-1) \times y$, $\frac{x}{y} = \exp(\ln x - \ln y)$; OUT!
- $x \times y = \exp(\ln x - (\ln 1 - \ln y))$; OUT!
- $-1 = 1 - 2,\ 2 = 1 - (\ln 1 - 1)$;
- VOILA!

EXP, MINUS, LOG, ONE - 4 buttons.

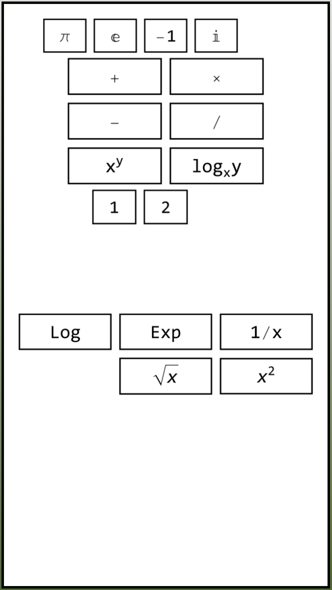
# Reduction of calculator - complex numbers




- $\sin x = \frac{e^{ix} - e^{-ix}}{2i}$; OUT!
- $\arccos x = \frac{\pi}{2} + i \ln\left(ix + \sqrt{1 - x^2}\right)$, $\arctan x = \frac{i}{2}\ln(1 - ix) - \frac{i}{2}\ln(1 + ix)$; OUT!
- $\pi = \frac{\ln(-1)}{i}$, $i = \sqrt{-1}$; OUT!
- $x^2 = x \times x$, $\sqrt{x} = x^{1/2}$; OUT!
- $\frac{1}{x} = x^{-1}$, $e = \exp(1)$; OUT!
- $\log_x y = \frac{\ln y}{\ln x}$, $x^y = \exp(y \ln x)$; OUT!
- $x + y = x - (-1) \times y$, $\frac{x}{y} = \exp(\ln x - \ln y)$; OUT!
- $x \times y = \exp(\ln x - (\ln 1 - \ln y))$; OUT!
- $-1 = 1 - 2$, $2 = 1 - (\ln 1 - 1)$;
- VOILA!

EXP, MINUS, LOG, ONE - 4 buttons.

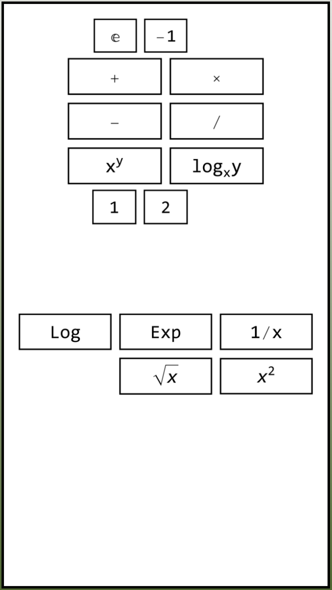
# Reduction of calculator - complex numbers




- $\sin x = \frac{e^{ix} - e^{-ix}}{2i}$; OUT!
- $\arccos x = \frac{\pi}{2} + i \ln\left(ix + \sqrt{1 - x^2}\right)$, $\arctan x = \frac{i}{2}\ln(1 - ix) - \frac{i}{2}\ln(1 + ix)$; OUT!
- $\pi = \frac{\ln(-1)}{i}, i = \sqrt{-1}$; OUT!
- $x^2 = x \times x, \sqrt{x} = x^{1/2}$; OUT!
- $\frac{1}{x} = x^{-1}, e = \exp(1)$; OUT!
- $\log_x y = \frac{\ln y}{\ln x}, x^y = \exp(y \ln x)$; OUT!
- $x + y = x - (-1) \times y$, $\frac{x}{y} = \exp(\ln x - \ln y)$; OUT!
- $x \times y = \exp(\ln x - (\ln 1 - \ln y))$; OUT!
- $-1 = 1 - 2,\ 2 = 1 - (\ln 1 - 1)$;
- VOILA!

EXP, MINUS, LOG, ONE - 4 buttons.

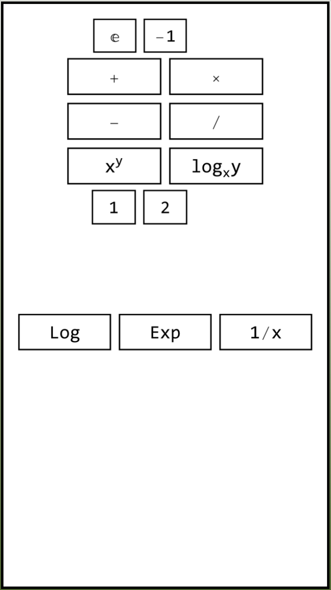
# Reduction of calculator - complex numbers




- $\sin x = \frac{e^{ix} - e^{-ix}}{2i}$; OUT!
- $\arccos x = \frac{\pi}{2} + i \ln\left(ix + \sqrt{1 - x^2}\right)$, $\arctan x = \frac{i}{2}\ln(1 - ix) - \frac{i}{2}\ln(1 + ix)$; OUT!
- $\pi = \frac{\ln(-1)}{i}$, $i = \sqrt{-1}$; OUT!
- $x^2 = x \times x$, $\sqrt{x} = x^{1/2}$; OUT!
- $\frac{1}{x} = x^{-1}$, $e = \exp(1)$; OUT!
- $\log_x y = \frac{\ln y}{\ln x}$, $x^y = \exp(y \ln x)$; OUT!
- $x + y = x - (-1) \times y$, $\frac{x}{y} = \exp(\ln x - \ln y)$; OUT!
- $x \times y = \exp(\ln x - (\ln 1 - \ln y))$; OUT!
- $-1 = 1 - 2,\ 2 = 1 - (\ln 1 - 1)$;
- VOILA!

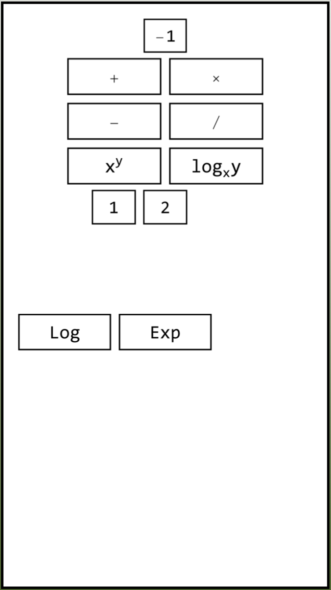
EXP, MINUS, LOG, ONE - 4 buttons.

# Reduction of calculator - complex numbers




- $\sin x = \frac{e^{ix} - e^{-ix}}{2i}$; OUT!
- $\arccos x = \frac{\pi}{2} + i \ln\left(ix + \sqrt{1 - x^2}\right)$, $\arctan x = \frac{i}{2} \ln(1 - ix) - \frac{i}{2} \ln(1 + ix)$; OUT!
- $\pi = \frac{\ln(-1)}{i}, i = \sqrt{-1}$; OUT!
- $x^2 = x \times x, \sqrt{x} = x^{1/2}$; OUT!
- $\frac{1}{x} = x^{-1}, e = \exp(1)$; OUT!
- $\log_x y = \frac{\ln y}{\ln x}, x^y = \exp(y \ln x)$; OUT!
- $x + y = x - (-1) \times y$, $\frac{x}{y} = \exp(\ln x - \ln y)$; OUT!
- $x \times y = \exp(\ln x - (\ln 1 - \ln y))$; OUT!
- $-1 = 1 - 2,\ 2 = 1 - (\ln 1 - 1)$;
- VOILA!

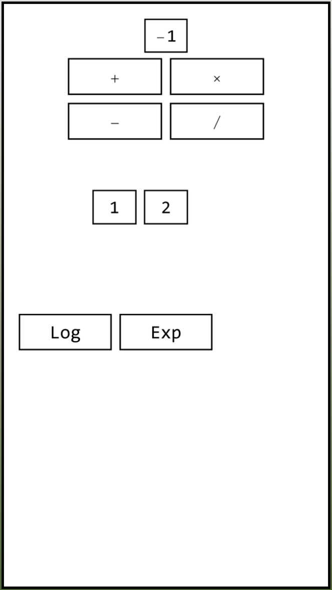
EXP, MINUS, LOG, ONE - 4 buttons.




- $\sin x = \frac{e^{ix} - e^{-ix}}{2i}$; OUT!
- $\arccos x = \frac{\pi}{2} + i \ln\left(ix + \sqrt{1 - x^2}\right)$, $\arctan x = \frac{i}{2} \ln(1 - ix) - \frac{i}{2} \ln(1 + ix)$; OUT!
- $\pi = \frac{\ln(-1)}{i}, i = \sqrt{-1}$; OUT!
- $x^2 = x \times x, \sqrt{x} = x^{1/2}$; OUT!
- $\frac{1}{x} = x^{-1}, e = \exp(1)$; OUT!
- $\log_x y = \frac{\ln y}{\ln x}, x^y = \exp(y \ln x)$; OUT!
- $x + y = x - (-1) \times y$, $\frac{1}{y} = \exp(\ln 1 - \ln y)$; OUT!
- $x \times y = \exp(\ln x - (\ln 1 - \ln y))$; OUT!
- $-1 = 1 - 2,\ 2 = 1 - (\ln 1 - 1)$;
- VOILA!

EXP, MINUS, LOG, ONE - 4 buttons.

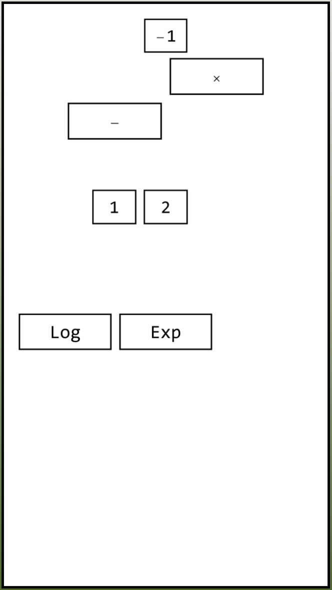
# Reduction of calculator - complex numbers




- $\sin x = \frac{e^{ix} - e^{-ix}}{2i}$; OUT!
- $\arccos x = \frac{\pi}{2} + i \ln\left(ix + \sqrt{1 - x^2}\right)$, $\arctan x = \frac{i}{2} \ln(1 - ix) - \frac{i}{2} \ln(1 + ix)$; OUT!
- $\pi = \frac{\ln(-1)}{i}$, $i = \sqrt{-1}$; OUT!
- $x^2 = x \times x$, $\sqrt{x} = x^{1/2}$; OUT!
- $\frac{1}{x} = x^{-1}$, $e = \exp(1)$; OUT!
- $\log_x y = \frac{\ln y}{\ln x}$, $x^y = \exp(y \ln x)$; OUT!
- $x + y = x - (-1) \times y$, $\frac{x}{y} = \exp(\ln x - \ln y)$; OUT!
- $x \times y = \exp(\ln x - (\ln 1 - \ln y))$; OUT!
- $-1 = 1 - 2$, $2 = 1 - (\ln 1 - 1)$;
- VOILA!

EXP, MINUS, LOG, ONE - 4 buttons.

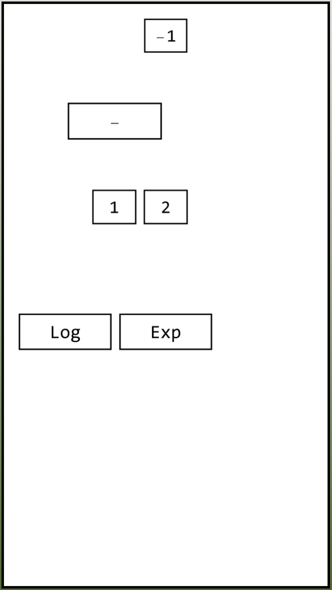
# Reduction of calculator - complex numbers




- $\sin x = \frac{e^{ix} - e^{-ix}}{2i}$; OUT!
- $\arccos x = \frac{\pi}{2} + i \ln\left(ix + \sqrt{1 - x^2}\right)$, $\arctan x = \frac{i}{2}\ln(1 - ix) - \frac{i}{2}\ln(1 + ix)$; OUT!
- $\pi = \frac{\ln(-1)}{i}, i = \sqrt{-1}$; OUT!
- $x^2 = x \times x, \sqrt{x} = x^{1/2}$; OUT!
- $\frac{1}{x} = x^{-1}, e = \exp(1)$; OUT!
- $\log_x y = \frac{\ln y}{\ln x}, x^y = \exp(y \ln x)$; OUT!
- $x + y = x - (-1) \times y$, $\frac{x}{y} = \exp(\ln x - \ln y)$; OUT!
- $x \times y = \exp(\ln x - (\ln 1 - \ln y))$; OUT!
- $-1 = 1 - 2,\ 2 = 1 - (\ln 1 - 1)$;
- VOILA!

EXP, MINUS, LOG, ONE - 4 buttons.

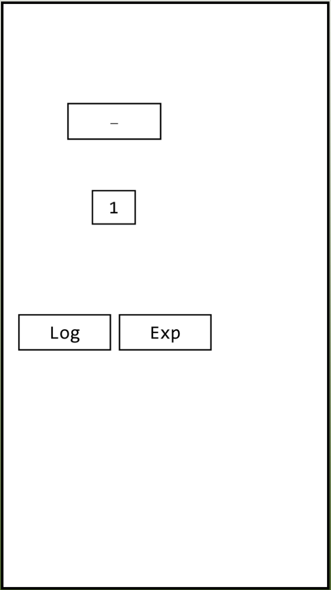
# Reduction of calculator - complex numbers




- $\sin x = \frac{e^{ix} - e^{-ix}}{2i}$; OUT!
- $\arccos x = \frac{\pi}{2} + i \ln\left(ix + \sqrt{1 - x^2}\right)$, $\arctan x = \frac{i}{2} \ln(1 - ix) - \frac{i}{2} \ln(1 + ix)$; OUT!
- $\pi = \frac{\ln(-1)}{i}, i = \sqrt{-1}$; OUT!
- $x^2 = x \times x, \sqrt{x} = x^{1/2}$; OUT!
- $\frac{1}{x} = x^{-1}, e = \exp(1)$; OUT!
- $\log_x y = \frac{\ln y}{\ln x}, x^y = \exp(y \ln x)$; OUT!
- $x + y = x - (-1) \times y$, $\frac{x}{y} = \exp(\ln x - \ln y)$; OUT!
- $x \times y = \exp(\ln x - (\ln 1 - \ln y))$; OUT!
- $-1 = 1 - 2, \; 2 = 1 - (\ln 1 - 1)$;
- VOILA!

  EXP, MINUS, LOG, ONE - 4 buttons.

# Reduction of calculator - complex numbers

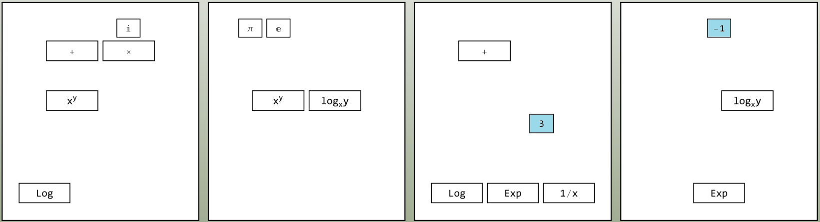



- $\sin x = \frac{e^{ix} - e^{-ix}}{2i}$; OUT!
- $\arccos x = \frac{\pi}{2} + i \ln\left(ix + \sqrt{1 - x^2}\right)$, $\arctan x = \frac{i}{2}\ln(1 - ix) - \frac{i}{2}\ln(1 + ix)$; OUT!
- $\pi = \frac{\ln(-1)}{i}, i = \sqrt{-1}$; OUT!
- $x^2 = x \times x, \sqrt{x} = x^{1/2}$; OUT!
- $\frac{1}{x} = x^{-1}, e = \exp(1)$; OUT!
- $\log_x y = \frac{\ln y}{\ln x}, x^y = \exp(y \ln x)$; OUT!
- $x + y = x - (-1) \times y$, $\frac{x}{y} = \exp(\ln x - \ln y)$; OUT!
- $x \times y = \exp(\ln x - (\ln 1 - \ln y))$; OUT!
- $-1 = 1 - 2,\ 2 = 1 - (\ln 1 - 1)$;
- VOILA!

EXP, MINUS, LOG, ONE - 4 buttons.

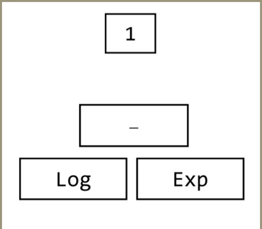
# Final reduction

There are multiple ways to reduce calculations to small primitive sets.




To find **single** operator, we must go *beyond* historically established primitives.

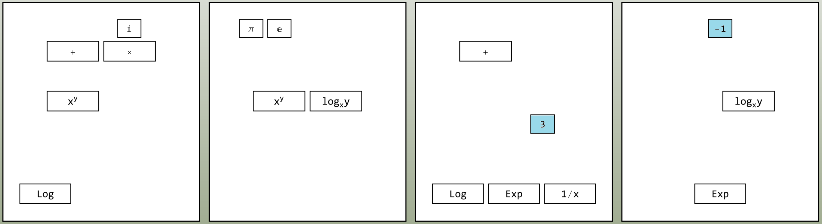
## The EML operator & 2-button calculator




1. $\operatorname{eml}(x, y) = \exp x - \ln y$
2. $\exp x = \operatorname{eml}(x, 1)$
3. $\ln x = \operatorname{eml}(1, \operatorname{eml}[\operatorname{eml}(1, x), 1])$
4. $x - y = \operatorname{eml}(\ln x, \exp y)$
5. ...

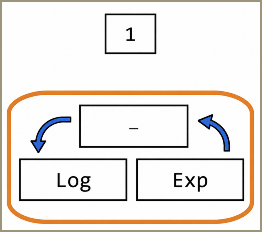
# Final reduction

There are multiple ways to reduce calculations to small primitive sets.




To find **single** operator, we must go *beyond* historically established primitives.

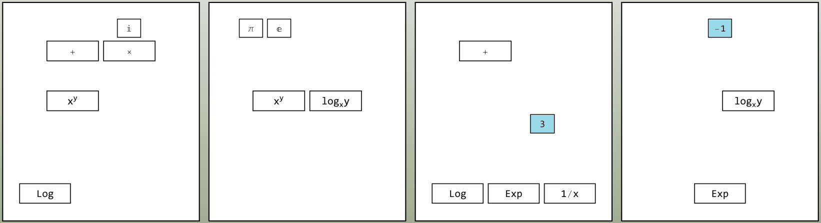
## The EML operator & 2-button calculator




1. $\text{eml}(x, y) = \exp x - \ln y$
2. $\exp x = \text{eml}(x, 1)$
3. $\ln x = \text{eml}(1, \text{eml}[\text{eml}(1, x), 1])$
4. $x - y = \text{eml}(\ln x, \exp y)$
5. ...

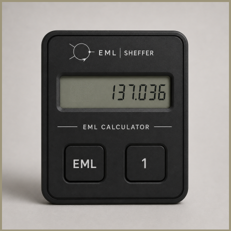
# Final reduction

There are multiple ways to reduce calculations to small primitive sets.




To find **single** operator, we must go *beyond* historically established primitives.

## The EML operator & 2-button calculator




1. $\mathrm{eml}(x, y) = \exp x - \ln y$
2. $\exp x = \mathrm{eml}(x, 1)$
3. $\ln x = \mathrm{eml}(1, \mathrm{eml}[\mathrm{eml}(1, x), 1])$
4. $x - y = \mathrm{eml}(\ln x, \exp y)$
5. $\ldots$

# How was the EML operator found?

In retrospective, the **E**xp-**M**inus-**L**og (EML) operator

$$\mathrm{eml}(x, y) = e^x - \ln y,$$

looks nearly obvious. **You cannot unsee it**.

**Search for single universal operator was not easy ...**

1. Does it exist at all? Abstract algebra says: YES, but known constructions use conditionals and compactification of real line/axis.
2. Is it differentiable? Do we need "pathological" functions like Minkowski $?(x)$ or Cantor Devil's staircase?
3. Is it elementary? Maybe we need special functions like $\Gamma(x)$ or higher-rank operations like tetration ${}^y x$?
4. If **it is elementary**, then of what form?

The practical method to answer above questions is brute-force search.

# Constant recognition (CR))

*CR is a niche subfield of experimental math: given decimal expansion -0.45158270528945 find formula which reproduce it. You can view it as benchmark for compression, random/direct search or pattern recognition.*

## Notable software

- RIES, Constant Recognizer
- Wolfram Alpha, SymPy/nsimplify, Maple/identify
- Plouffe inverter, ISC2, Ask Constants, MESearch

## How to use it as formula discovery tool?

1. Is $x + y$ expressible in terms of EML?
2. Let's substitute some "exotic" constants, e.g. Euler's $\gamma = x \simeq 0.5772156649$, and Glaisher $A = y \simeq 1.2824271291$
3. Assume $\gamma, A$ are algebraically EML-independent (Schanuel conjecture)
4. Run CR on $\gamma + A = 1.859642794002155497481854658952$
5. Wait for CR to discover EML formula

$\mathrm{eml}(1, \mathrm{eml}(\mathrm{eml}(\mathrm{eml}(\gamma, \mathrm{eml}(\mathrm{eml}(\gamma, \mathrm{eml}(1, \mathrm{eml}(A, 1))), 1)), \mathrm{eml}(\gamma, 1)), 1))$

# Constant recognition (CR))

*CR is a niche subfield of experimental math: given decimal expansion -0.45158270528945 find formula which reproduce it. You can view it as benchmark for compression, random/direct search or pattern recognition.*

## Notable software

- RIES, Constant Recognizer
- Wolfram Alpha, SymPy/nsimplify, Maple/identify
- Plouffe inverter, ISC2, Ask Constants, MESearch

## How to use it as formula discovery tool?

1. Is $x + y$ expressible in terms of EML?
2. Let's substitute some "exotic" constants, e.g.
   Euler's $\gamma = x \simeq 0.5772156649$, and Glaisher $A = y \simeq 1.2824271291$
3. Assume $\gamma, A$ are algebraically EML-independent (Schanuel conjecture)
4. Run CR on $\gamma + A = 1.8596427940021554974818546589522$
5. Wait for CR to discover EML formula

$$\mathrm{eml}(1, \mathrm{eml}(\mathrm{eml}(\mathrm{eml}(x, \mathrm{eml}(\mathrm{eml}(x, \mathrm{eml}(1, \mathrm{eml}(y, 1))), 1)), \mathrm{eml}(x, 1)), 1))$$

# Alien math: what the EML is?

## What sense is in using high-rank nonlinear operator to multiply/add?

$x + y =$ eml(1,eml(eml(eml(x,eml(eml(x,eml(1, eml(y,1))),1)),eml(x,1)),1))

**or**

$\pi =$ eml(eml(eml(1,eml(eml(1,eml(1,eml(eml(1,eml(eml(1,eml(eml(1,eml(1,eml(eml(1,1
),1))),1)),eml(eml(eml(eml(eml(1,eml(eml(1,eml(1,eml(eml(1,eml(1,eml(eml(1,eml(eml(
1,eml(eml(1,eml(1,eml(eml(1,1),1))),1)),eml(1,1))),1))),1))),1)),eml(eml(eml(1,eml(eml(
1,eml(1,eml(eml(1,1),1))),1)),eml(eml(1,eml(eml(1,eml(eml(eml(1,eml(eml(1,eml(1,eml(
eml(1,1),1))),1)),eml(eml(1,eml(eml(1,eml(eml(1,eml(eml(1,1),1)),eml(eml(eml(1,eml(e
ml(1,eml(1,eml(eml(1,1),1))),1)),eml(1,1)),1))),1)),1)),1)),1)),1)),1)),1),1),1))),1))),1)),e
ml(eml(eml(1,eml(eml(1,eml(1,eml(eml(1,1),1))),1)),eml(eml(1,eml(eml(1,eml(1,eml(eml
(1,eml(eml(1,eml(eml(1,eml(1,eml(eml(1,1),1))),1)),eml(1,1))),1))),1)),1)),1)),1)

## Mathematical primitives

- Humans, both evolutionary and by education, use counting as primitive math operation. Kronecker: *Die ganzen Zahlen hat der liebe Gott gemacht, alles andere ist Menschenwerk.* For alien perspective, ask LLM.
- *Solaris*: intelligent ocean planet. Nothing to count beyond 1, but exp is everywhere: attenuation, exponential growth, RC circuit, radioactivity $T_{1/2} = \tau \ln 2$, statistic $e^{-E/kT}$, entropy $S = k \log W$, inflation $e^{H_\infty t}$, gaussian $e^{-\pi x^2}$, quantum phase $e^{i\varphi}|\psi\rangle$
- Physics: quantum/discrete vs classical/continuous ?
- Kindergarten with VR "magick" boxes: doubling, adding one, identity ...

# Binary tree grammar

## Context-free grammar

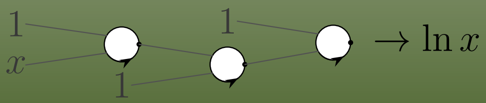
Every elementary formula in the EML form has a binary expression tree. For constants, grammar is particularly simple

$$S \to 1 \mid \mathrm{eml}(S, S)$$

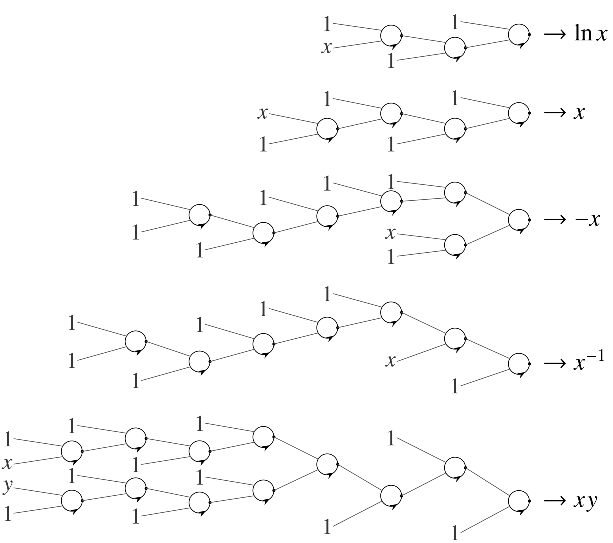
For functions of many variables $f(x, y, z, \ldots)$, just add relevant variables $x, y, z, \ldots$ as additional terminal symbols

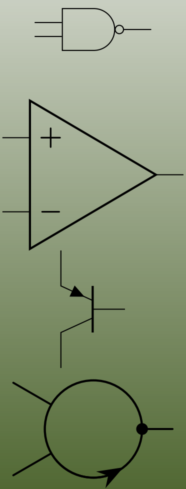
$$S \to 1, x, y, z, \ldots \mid \mathrm{eml}(S, S)$$

Above is among the best studied data structures in computer science. All elementary functions compiled to EML, are of this form!



# Formulas as 'analog' circuits



# EML master formula

$$T_{\text{eml},d}\left(x;\{a_1,a_2,\ldots,a_k\},\{b_1,b_2,\ldots,b_k\},\{c_1,c_2,\ldots,c_k\}\right),\ k=2^{d+1}-2$$

## The full level-2 master formula

$$\begin{aligned}&T_{\text{eml},d}\left(x;\{\alpha_1,\ldots,\alpha_6\},\{\beta_1,\ldots,\beta_6\},\{\gamma_1,\gamma_2,0,0,0,0\}\right)\\&=\text{eml}\Big[\alpha_1+\beta_1 x+\gamma_1\,\text{eml}\left(\alpha_3+\beta_3 x,\alpha_4+\beta_4 x\right),\\&\qquad\alpha_2+\beta_2 x+\gamma_2\,\text{eml}\left(\alpha_5+\beta_5 x,\alpha_6+\beta_6 x\right)\Big]\end{aligned}$$

By construction, master formula includes all possible elementary math expressions, e.g:

$$T_{\text{eml},1}(\{1,1\},\{1,0\},\{0,0\})=e^x.$$

Can we fit EML trees to the data?

# Symbolic regression (SR)

SR is a subfield of ML, attempting to fit data using **ANY** formulas.

$$f(t,x,y) = A\sin(\omega t+\phi)^{\lambda} \times (\alpha+\beta x^{\gamma})^{\delta} + a^{b^{c^{y}}}$$

Due to excessive diversity of math, genetic, AI or exhaustive enumeration methods are used to generate candidate formulas, while gradient-based optimization determine the free parameters.

**The problem: completeness and expressiveness of candidate formulas**

In e.g. PySR, we set up search using

$$\sin, \cos, \sqrt{\cdot}, \exp \text{ and } +, -, \times$$

with several free parameters. Is it complete? What if ground truth is rational function? Would we recover it, eventually? Should we add binary exponentiation and/or division? → This is exactly "broken calc" problem!

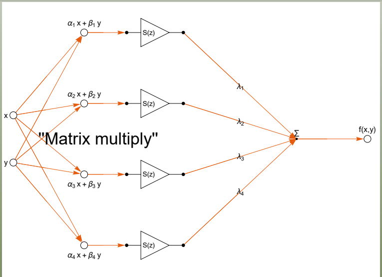
SOLUTION: problem vanishes, when we have **ONE** primitive: EML!

NOTE: proof of concept, does not scale above depth 6.

# Future: AI with built-in elementary functions?

Older (non-reasoning) LLM models struggled even with basic arithmetic like multiplication table, not to mention trig functions like $\sin, \cos, \ldots$

Can we "bake in" perfect math library into neural net architecture?

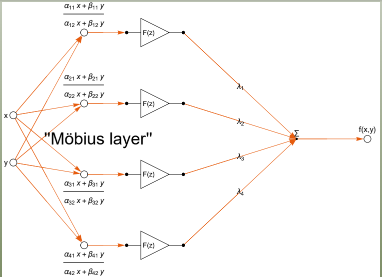



- Standard activation functions are e.g:
  - ReLU $S(z) = (|z| + z)/2$,
  - sigmoid $S(z) = 1/(1 + e^{-z})$.
- Above provide **approximation** ONLY.
- Can we find $F(z)$ which gives EML-type completeness, while keeping old good properties of ANNs?

# Future: AI with built-in elementary functions?

**Older (non-reasoning) LLM models struggled even with basic arithmetic like multiplication table, not to mention trig functions like $\sin, \cos, \ldots$**

Can we "bake in" perfect math library into neural net architecture?

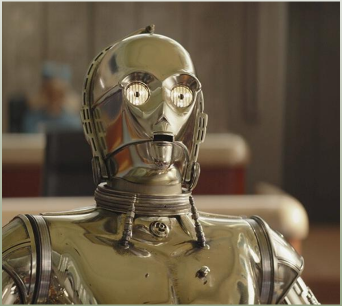



- Standard activation functions are e.g:
  - ReLU $S(z) = (|z| + z)/2$,
  - sigmoid $S(z) = 1/(1 + e^{-z})$.
- Above provide **approximation** ONLY.
- Can we find $F(z)$ which gives EML-type completeness, while keeping old good properties of ANNs?

PRELIMINARY: rational functions $(+, -, \times, /)$ with $\boxed{F(z) = e^z + \ln z}$ compute 'ALL' elementary functions, by recovering $\exp$ and $\ln$ separately.

$$e^z = \frac{F(3z) - F(z) - \ln 3}{F(2z) - F(z) - \ln 2} - 1, \qquad \ln z = F(z) - e^z.$$

# Do we really want savant AI with built-in advanced math?



# Summary

- The EML, sole sufficient binary *Sheffer* operator for evaluation of standard elementary functions, has been found
- EML is remarkably simple, Exp-Minus-Log

$$\mathrm{eml}\,(x, y) = \exp x - \log y$$

- EML require constant $1$ and evaluation in complex domain
- EML generate real elementary functions as binary trees

$$S \to 1 \mid \mathrm{eml}(S, S)$$

- EML is the tip of iceberg
    1. EDL $\{\mathrm{edl}(x, y) = e^x / \ln y, e\}$, -EML $\{-\mathrm{eml}(x, y), -\infty\}$
    2. LDE $\{\ln x / e^y, 0\}$, PLI $\{1/(\ln x)^y, 1\}, \{1/(\ln x)^y, \sqrt[y]{\ln x}\}$
    3. ternary variants T(x,y,z) $= e^{x-y} \ln x / \ln z$
    4. rational variant $\{x \# y = 1/(x - y), e^x + \ln x, 1\}$
- Proof of concept for pure gradient-based symbolic regression
- Numerous independent implementations and follow-up studies

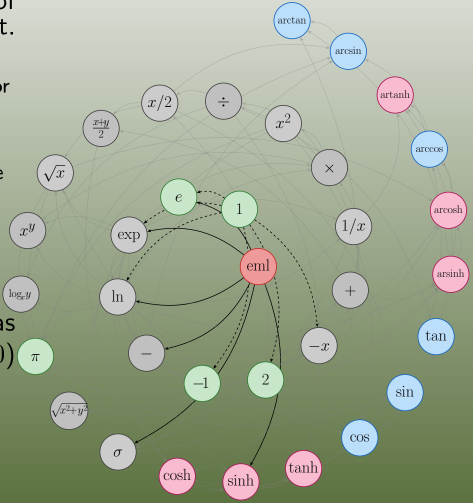
Thanks to everyone who found this topic amusing, inspiring or disturbing!

# EXTRA SLIDES

# Genetic DNA analogies

In the EML form, bioinformatical view of elementary functions is even more direct.

1. EML as Last Universal Common Ancestor (LUCA)
2. calculator buttons as conserved EML sequences, in analogy to HOX or Histone H4 proteins in living organisms
3. most EML sequences generate "bad" expressions, we keep using only those, which proved useful in practice




For example, human-readable formulas in EML binary form $(1 \to 1, \mathrm{eml} \to 0)$ would often include sequence

00100010010101000100101011101111010001001001001001001001010111011101110111011111

which is just a number **10**.

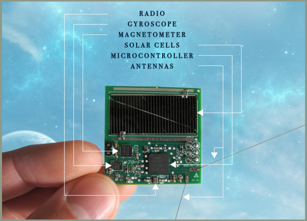
# Follow-up applications, extensions and fun

1. proof autoformalization in LEAN4 (B. Naskrecki)
2. simulations of the FPGA circuits (A. Taylor, P. Ibelings)
3. pure-EML RPN stack calculator (W. Young)
4. LLM neural net size compression (S. Majcher)
5. algebraic structure (T. Stachowiak)
6. symbolic integration (R. Belaiche)
7. gradient/landscape optimization (C. Gupta)
8. fractal and synth music generation (P. Bragoszewski)
9. programming competitions (Code golf)
10. correctly rounded IEEE 754 numerical implementation (K. Szewczuk)
11. philosophy of scientific discovery (S. Philos)

# FPGA simulation and RPN implementation

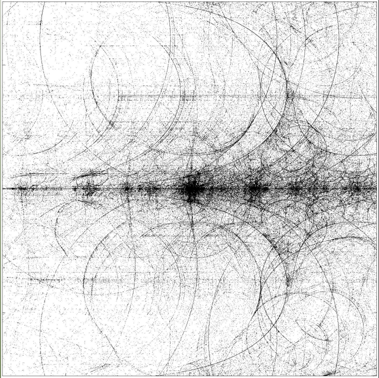
Simulations of the real modern FPGA circuits by A. Taylor and P. Ibelings demonstrated niche applications of EML even now.

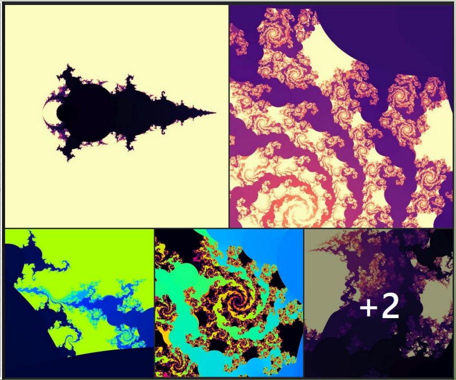
1. Flexibility is required
2. Resources are constrained
3. Throughput is not critical




This could be useful for nano satellites, and in future laser-powered swarm in Proxima Centauri system.

# EML numbers/EML fractals





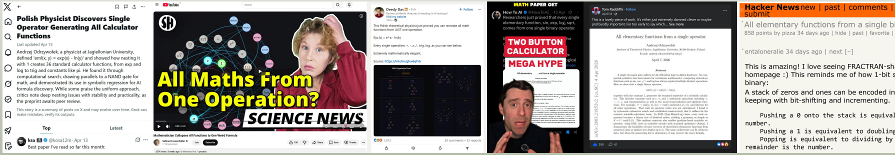

Rendering by P. Bragoszewski
https://github.com/indieboomer/eml-fractal-explorer

# ALL elementary functions controversy

| Expression or implicit formula | EML | Math | Math + $\mathbb{C}$ | $\sqrt[\infty]{f}$ | C <math.h> |
|---|---|---|---|---|---|
| $\sin x$, $\log x$, $\sqrt{x}$ | yes | yes | yes | yes | yes |
| $\lvert x\rvert$, $\lfloor x\rfloor$, $\operatorname{Re} z$ | no | no | no | no | yes |
| ${}_4F_3\left[\frac{1}{5},\frac{2}{5},\frac{3}{5},\frac{4}{5};\frac{1}{2},\frac{3}{4},\frac{5}{4};5\left(\frac{5}{4}x\right)^4\right]$ | no | yes | yes | yes | no |
| $r^7+ar^3+br^2+cr+1=0$ | no | yes | yes | yes | no |
| $W(x)e^{W(x)}=x$ | no | no | no | yes | no |
| $\cos^\infty(x)$ | no | no | yes | yes | no |
| $x+\gamma$ | ? | ? | yes | yes | no |
| $x! = \Gamma(x+1)$ | no | no | no | no | yes |
| ${}^y x$ | no | no | no | no | no |

- Common sense in STEM equates "calculator buttons" or `<math.h>` (`libm`) with "elementary functions"
- Mathematical definition adds *algebraic adjunctions*:
  - algebraic numbers and implicit polynomials
  - some include (!) generic complex constants as well ($\aleph_0 \to \mathfrak{c}$)
- Wolfram extends it to analytic functions ($\infty$-order polynomial roots)
- 5-th order polynomial confusion: exact, radical, *explicite*, elementary?
- Famous math questions: equation solvability and symbolic integration
- T. Chow: "*purely transcendental elementary functions*".
- EML might provide compact way to define EL ($\exp$-$\log$) functions

# EML going viral on X, Li, Fb, YouTube, Ig and TikTok




1. arXiv:2603.21852 posted 23 Mar, v2 4 Apr, no-one noticed for 3 weeks
2. on Mon 13 Apr it exploded: sidebar NEWS on X, 1M views posts, 858HN pts, 12000% traffic increase on my WWW, tens of downloads of . . . my "How to build the perfect Igloo" Eureka article!
3. tens of notifications, several e-mails every hour, hype ended in 48 hours
4. countless GitHub repos, simulators, articles, videos and commentaries
5. paper X-rayed for every possible aspect, my photos circulate on internet
6. wiki page on EML created, then deleted
7. *probably largest social media viral/hype on math/cs preprint?*

It is rare for science topic to go viral . . . especially for math and computer science. Reaction resembled famous **Monty Hall Problem**.

# Why arxiv:2603.21852 went viral?

1. Result is easy to verify for everyone with first-year university math:
   - logarithms and exponentials
   - "elementary" functions
   - basic complex numbers and analysis
   - simple logic and combinatorics
2. Entry level is low: a lot of useful contributions come from students
3. On the other hand, nearly forgotten knowledge of slide rule and RPN calculators bring retired engineers back into action
4. Vague terminology related to concept of "functions" and especially "elementary functions" caused arguments
5. Idea of 2-button calc looks weird, cause mixed reactions among people who spent years learning $\sin, \cos, \log, \tan, \sqrt{\cdot}$ ...
6. Polarized discussion on application potential, social media algorithms amplify extreme views and boost them
7. Multidisciplinary connections to electronics, bioinformatics, machine learning, abstract math, history of science, philosophy, ...

# Seven stages of EML denial

1. EML is not possible and must be wrong.
2. Maybe EML is correct, but there is no proof.
3. Well, there are proofs, but this was already known.
4. Maybe it was not known, but it is trivial, just no one asked.
5. EML is not completely trivial, but there will never be any applications for it.
6. There are niche applications, but we already have better methods.
7. EML is better at some tasks, but it is not economically viable, no money come from it . . .

# What was known before EML?

- rational functions allow reduction of $+, -, \times, /$ to $\frac{1}{x-y}$ with **two** constants 0 and 1
- abstract algebra allow reduction of any set of binary operations to one by selective dispatcher using multiple constants
- standard calculator operations are reducible to arithmetic, complex-valued "constants", $\exp$ and $\ln$
- Hua's identity allow to compute square and multiplication from addition, subtration and reciprocal (inverse element)

$$a - \left(a^{-1} + \left(b^{-1} - a\right)^{-1}\right)^{-1} = aba$$

**What was not known**

EML is non-trivial merger of the above concepts, demonstrating that exact elementary formulas can be cast into uniform tree form with single building block. This is close to architecture used in modern machine learning, artificial neural networks in particular.

# Kolmogorov-style complexity of mathematical formulas

## Occam Razor for lagrangian density $\mathcal{L}$

How to sort, e.g. $f(R)$ gravity theories by complexity?

$$\phi, \phi+\frac{\phi^2}{137}, \log(1+\phi), e^{\phi}-1, \phi+\phi^4, \frac{1}{1-\phi}-1, \sqrt{\sqrt{1+2\phi^2}-1}, 1-\cos\phi$$

Using EML RPN komplexity, we get clear ranking: 1. $\phi$ (K=1,D=0), 2. $\log(1+\phi)$ (K=33,D=13) 3. $e^{\phi}-1$ (K=45, D=12) 4. $\phi+\phi^4$ (K=153, D=27) 5. $\frac{1}{1-\phi}-1$ (K=189,D=36) 6. $\sqrt{\sqrt{1+2\phi^2}-1}$ (K=485,D=59) 7. $1-\cos\phi$ (K=769, D=68) 8. $\phi+\frac{\phi^2}{137}$ (K=3741,D=64)

## Can we enumerate ALL elementary functions without repetitions?

1. enumeration of rationals possible via Stern-Brocot tree or by composing "ladder" function and reciprocal (Calkin–Wilf)
2. can we do the same for EML expressions?

# Hyperoperations

| Rank | Definition | Operation | Inverse operation | Neutral element | Self-inverse |
|---|---|---|---|---|---|
| 0 | Counting | Successor $x+1$ | Predecessor $x-1$ | $-\infty$ | $2\lfloor x\rfloor+1-x$ |
| 1 | Repeated successor | Addition $x+y$ | Subtraction $x-y$ | $0$ | $-x+0$ |
| 2 | Repeated addition | Multiplication $x\times y$ | Division $x/y$ | $1$ | $1/x$ |
| 3 | Repeated multiplication | Exponentiation $x^y$ | Logarithm $\log_x y$ | $e$ | $e^{1/\ln x}$ |
| 4 | Power towers | Tetration ${}^y x$ | slog, ssqrt | ? | ? |
| 5 | Repeated tetration | Pentation | ? | ? | ? |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |

$$\mathrm{suc}(x) = x+1,\ x+n = \overbrace{\mathrm{suc}(\mathrm{suc}(\cdots\mathrm{suc}(x)\cdots))}^{n \text{ times}},$$

$$x\cdot n = \underbrace{x+x+x+\cdots+x}_{n \text{ times}},\ x^n = \underbrace{x\cdot x\cdot x\cdot\ldots\cdot x}_{n \text{ times}},\ {}^n x = \overbrace{x^{x^{\cdot^{\cdot^{x}}}}}^{n \text{ times}}$$