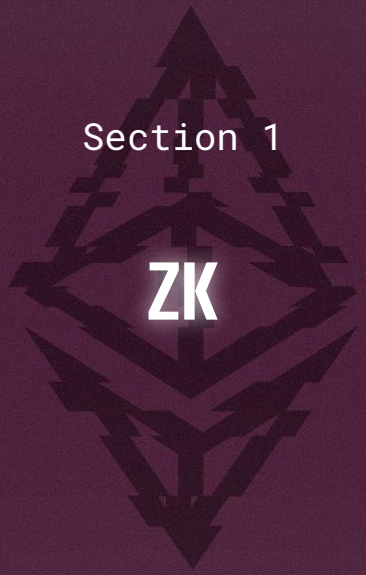
The Supreme Ruler of the World

zk math rules
for developers

Don Beaver

researcher, builder @ fierce logic

Section 1



"ZK/VK" arguments

I know a sudoku solution.

All 400,000 signatures are good.

orly?

The buzz around scaling

- One person does some work, everybody else piggybacks
  - because you can just check the results

- What did the consensus say?
  - *Are you sure?* Got CPU?
- What's the new state?
  - *Are you sure?* Got gas?
- What's the state of a rollup, of another chain?
  - *Are you sure?* Got gas?

Checking and Discreetly Delegating

Send a message - append a **checksum**

Prove a calculation - perform a **SumCheck**

Delegate a calculation - perform a **SumCalc**

Supreme Ruler

Exponential explosion for delegating/checking:

- delegate/check **2^n** steps at a cost of **O(n)**
- delegate/check **n** steps at a cost of **log n**

The **Million for 20 rule**, or **Industrial ZK**:

- check a **million** CPU for a price of **20** gas

brought to you by

the number

Supreme Ruler

ruler

2

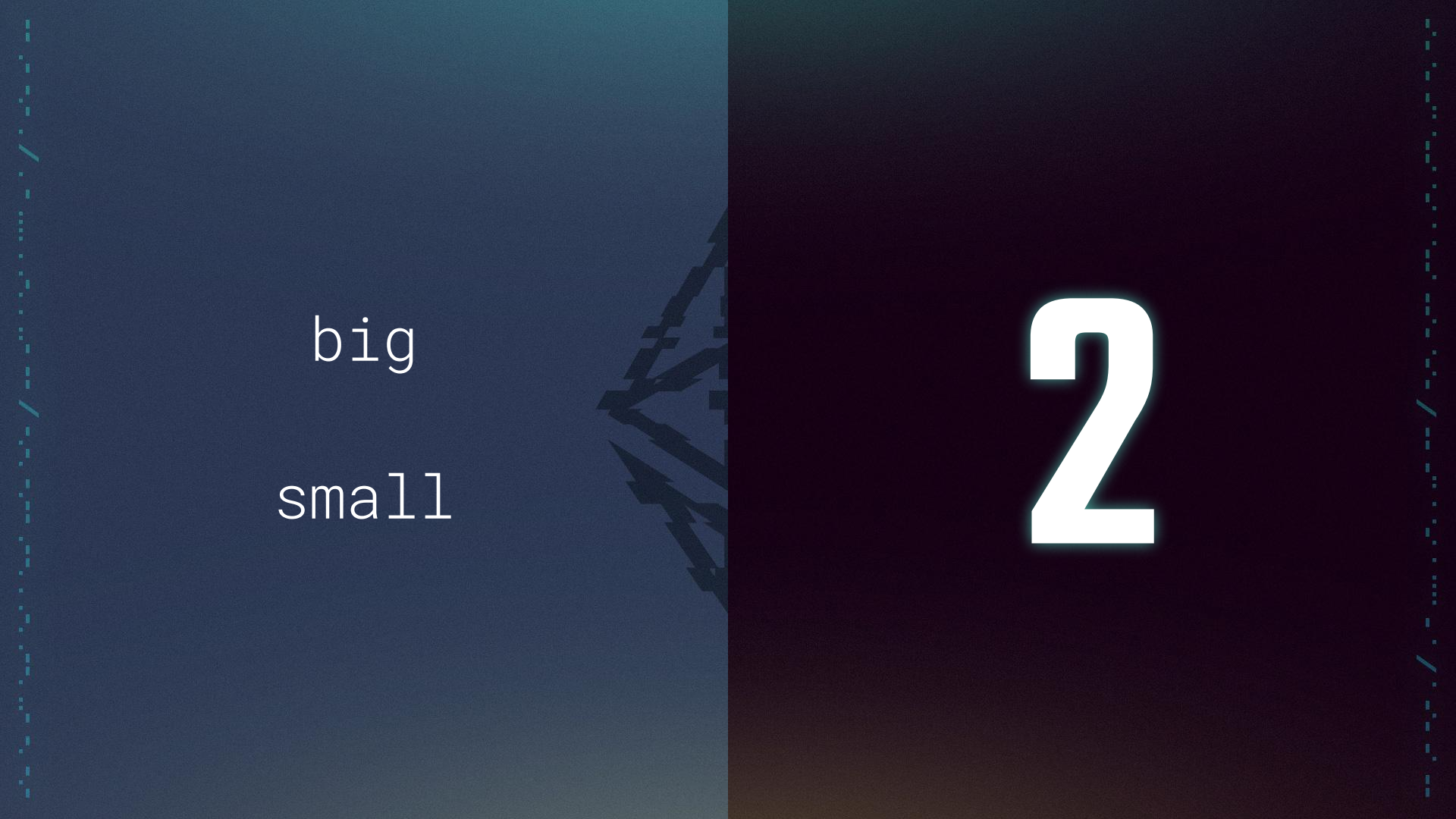
delegating

checking

2

big

small



swift

slow

2

cpu powered

gas powered

2

~~tls cryptographer~~

tls adopter

2

~~zk cryptographer~~

zk/vk adopter

2

~~moon math~~

earth math

2

multilinear

line

2

industrial age

theory age

2

2024

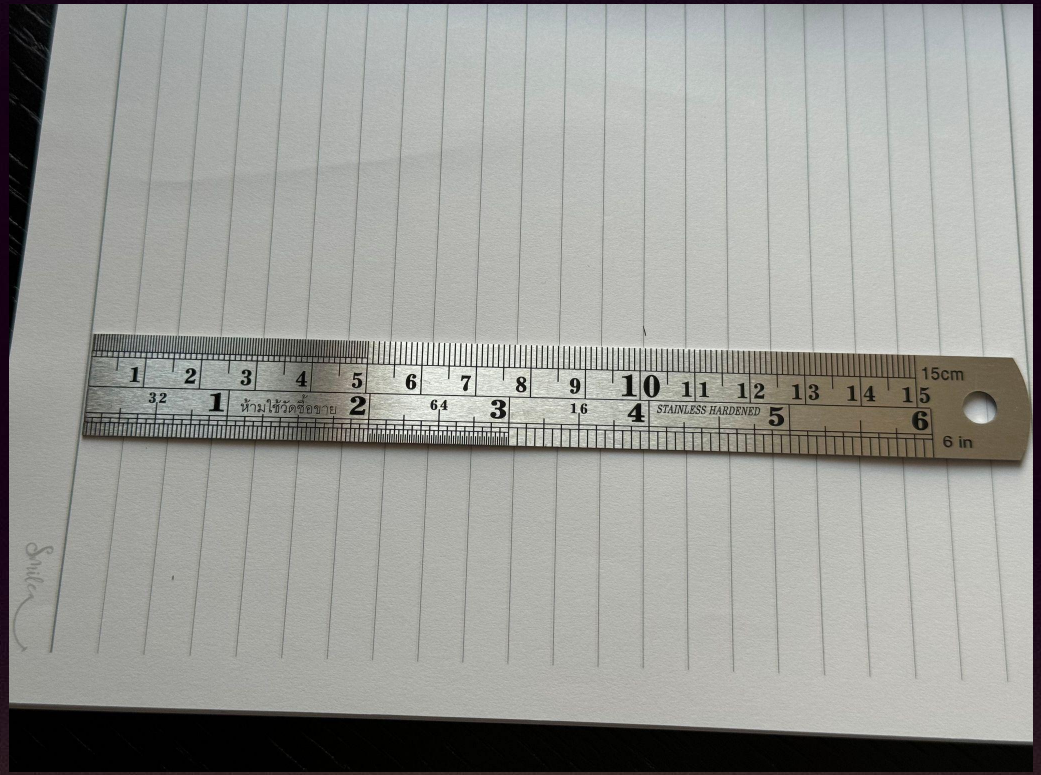
1989

2

Section 2

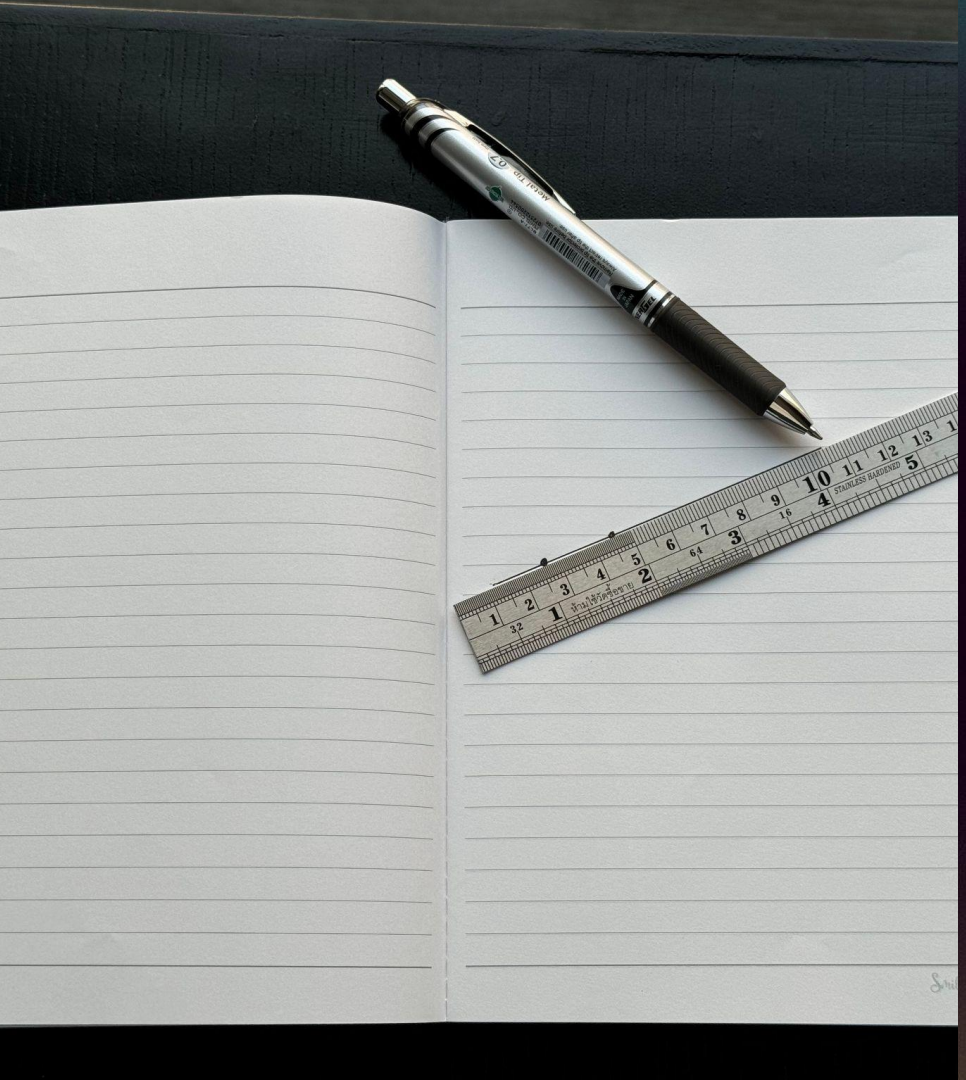
An Everyday Ruler

An Everyday Ruler



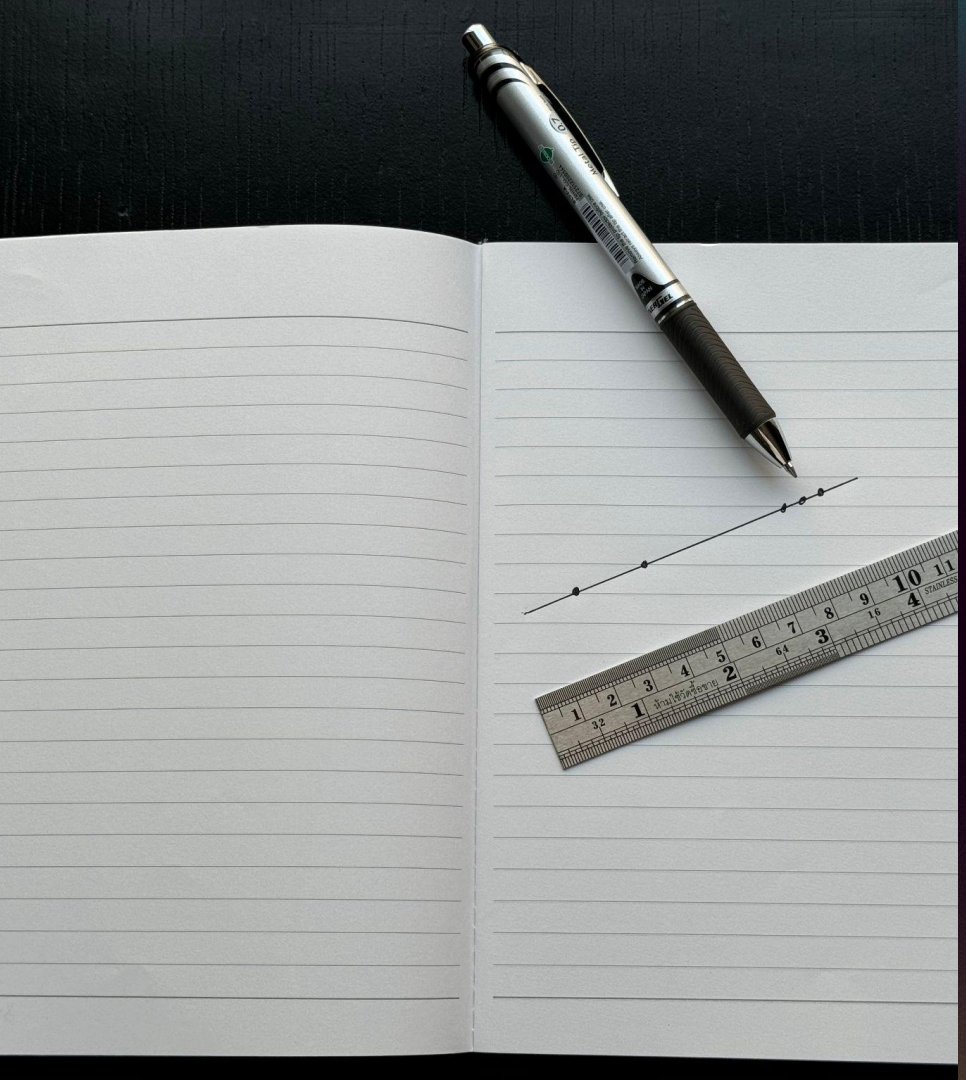
A Ruler Extends

A ruler draws
local order out
to the farthest reaches



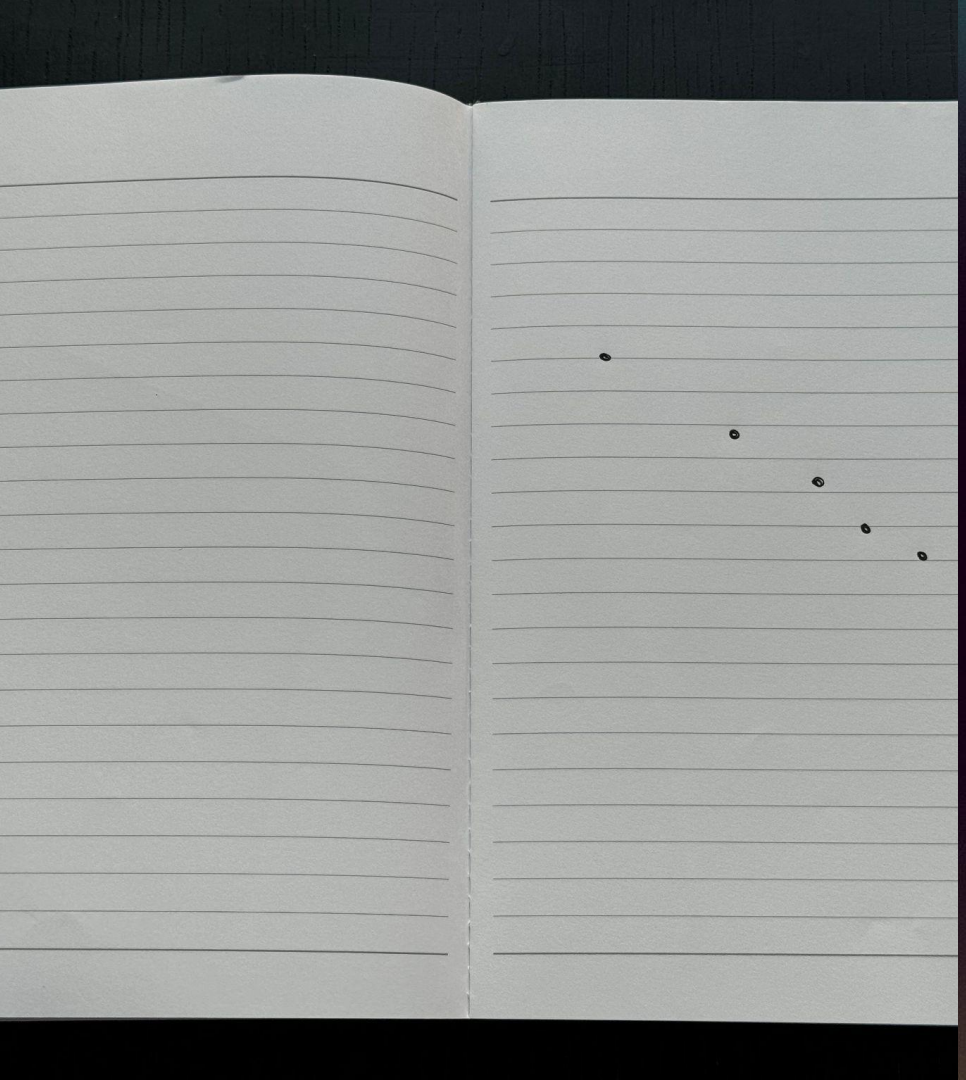
A Ruler Extends

A ruler draws
local order out
to the farthest reaches



A Ruler Extends

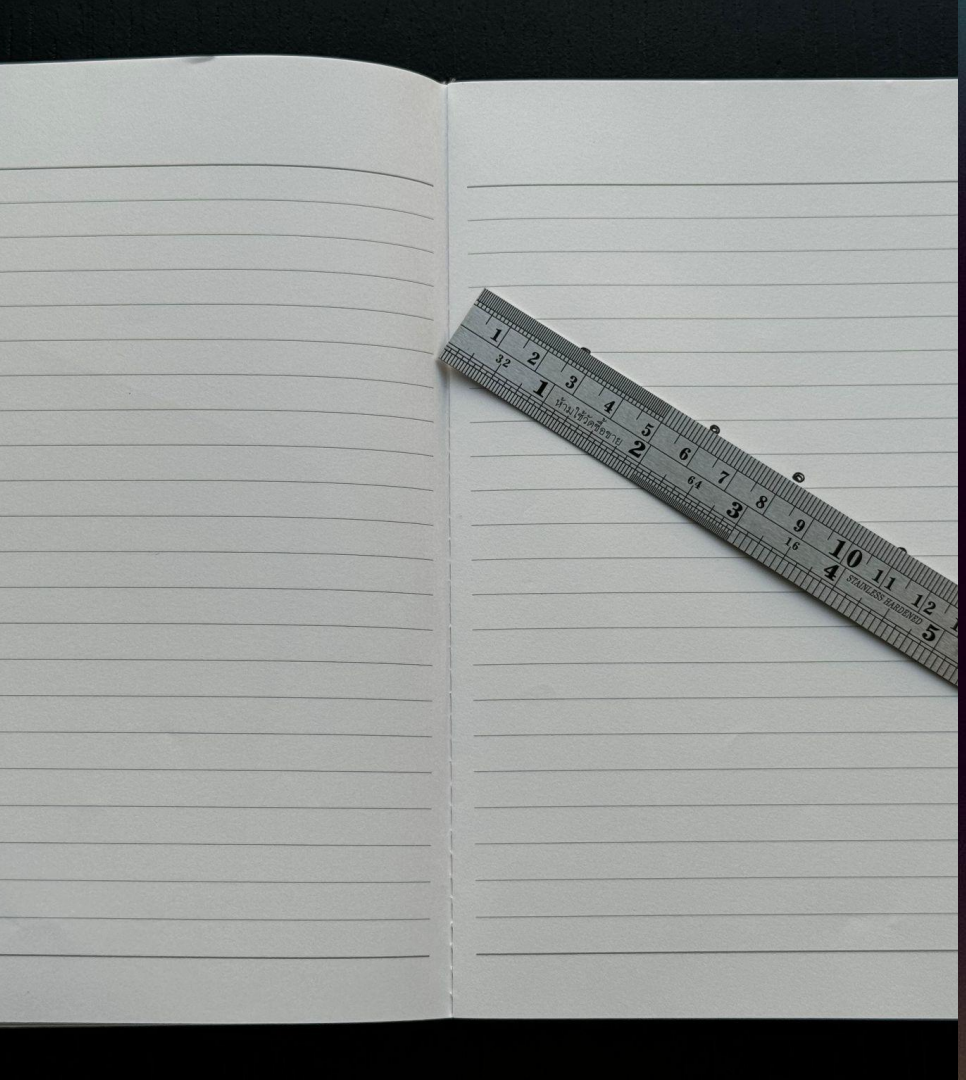
A ruler draws
local order out
to the farthest reaches



A Ruler Aligns

A ruler keeps everything in line.

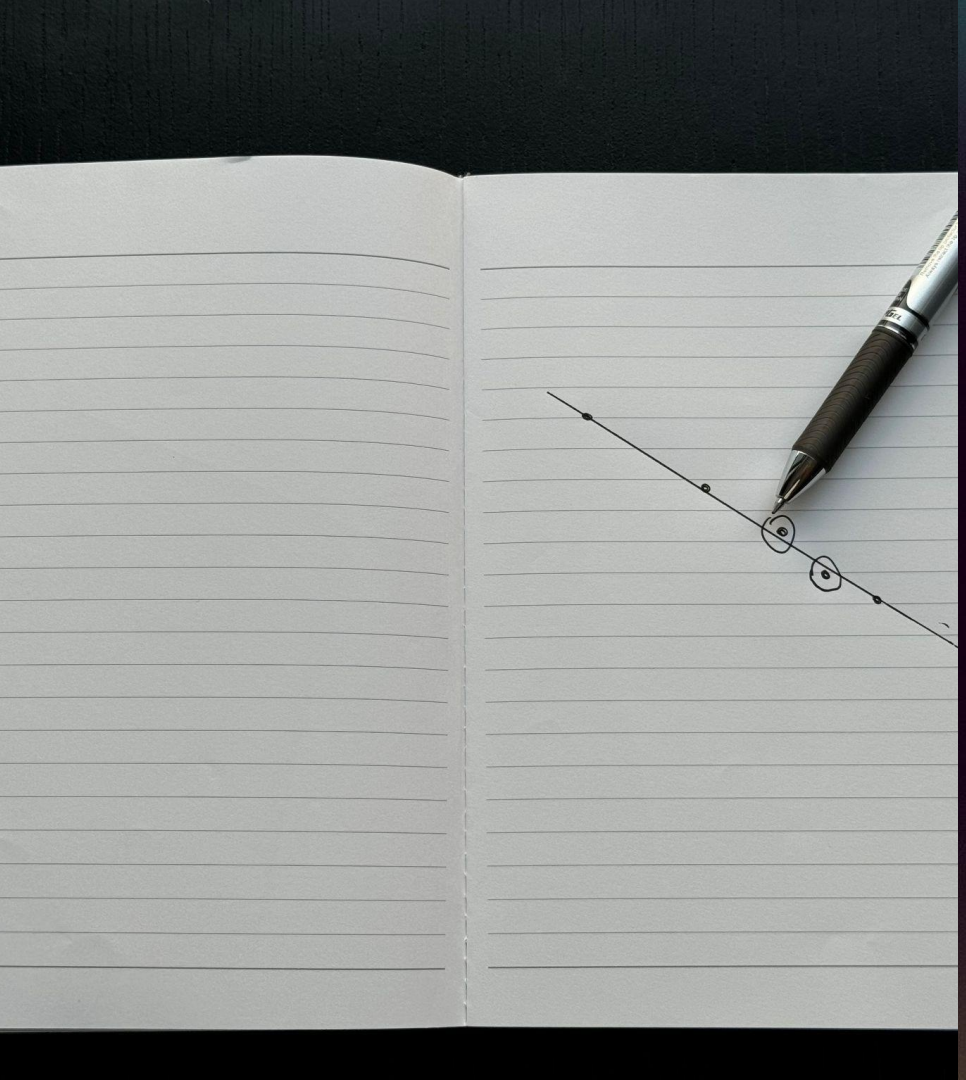
A ruler ensures claims are compliant.



A Ruler Aligns

A ruler keeps everything in line.

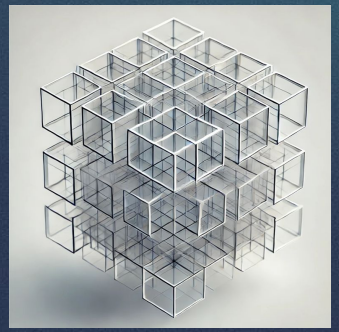
A ruler ensures claims are compliant.



A Ruler Aligns

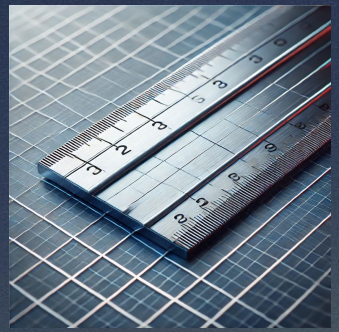
A ruler keeps everything in line.

A ruler ensures claims are compliant.



Supreme Ruler

Ruler



2

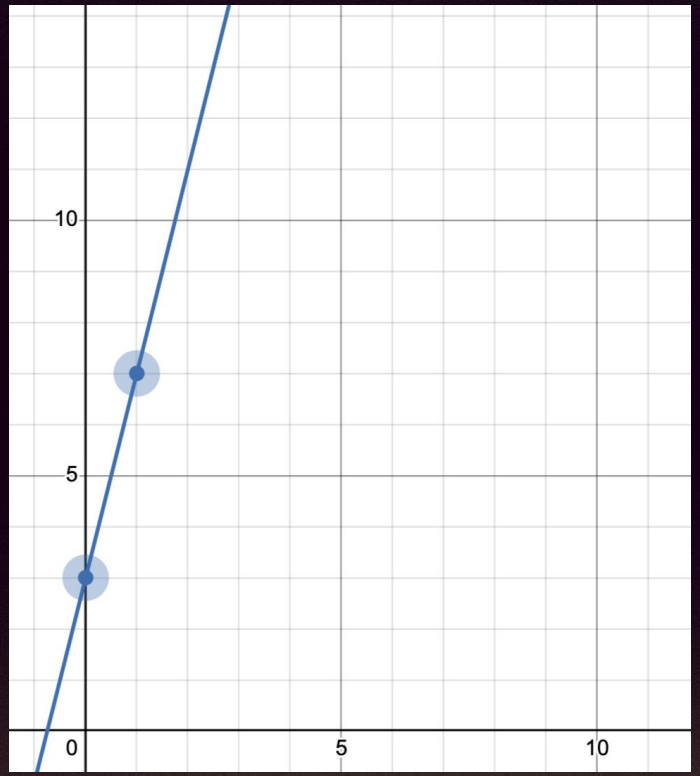
Section 3

Tables and Linear Extensions

Table of a 1-bit function

x	$f(x)$
0	3
1	7

Table of a 1-bit function - as a line



One-hot helpers

eq(0,x) = 1-x **“is x equal to 0?”**

```
eq(0,0) = 1
eq(0,1) = 0
```

eq(1,x) = x **“is x equal to 1?”**

```
eq(1,0) = 0
eq(1,1) = 1
```

Linear extension of the 2-entry table f(x)

$$F(x) = eq(0, x)3 + eq(1, x)7$$

Linear extension of the 2-entry table f(x)

$$F(x) = eq(0, x)3 + eq(1, x)7$$

$$= (1 - x)3 + x7$$

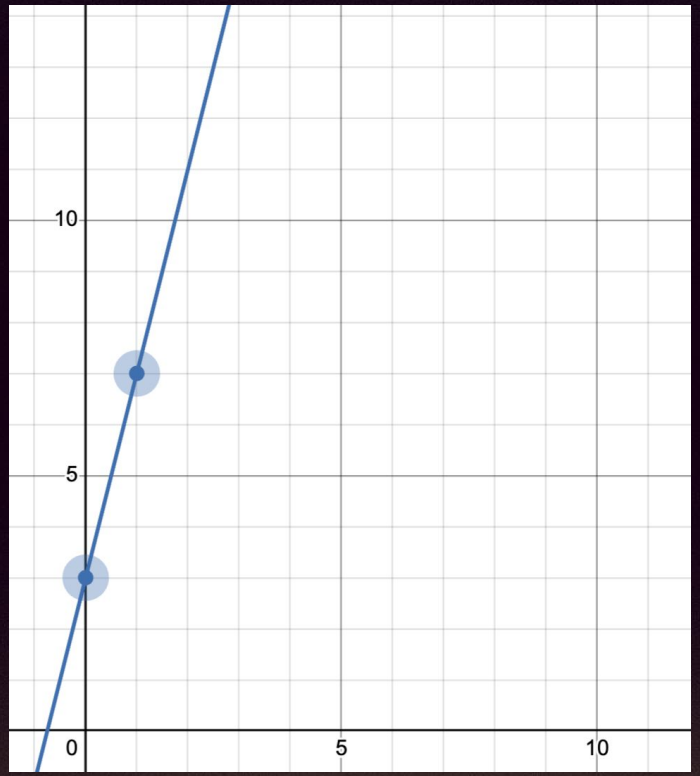
Linear extension of the 2-entry table f(x)

$$F(x) = eq(0, x)3 + eq(1, x)7$$

$$= (1 - x)3 + x7$$

$$= 4x + 3$$

Table of a 1-bit function - as a line



Earth Maths are Easy

Discreet Delegation

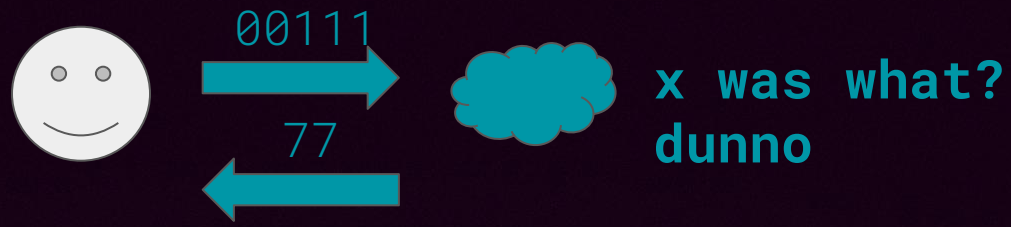
[AFK87]

f(0110)?

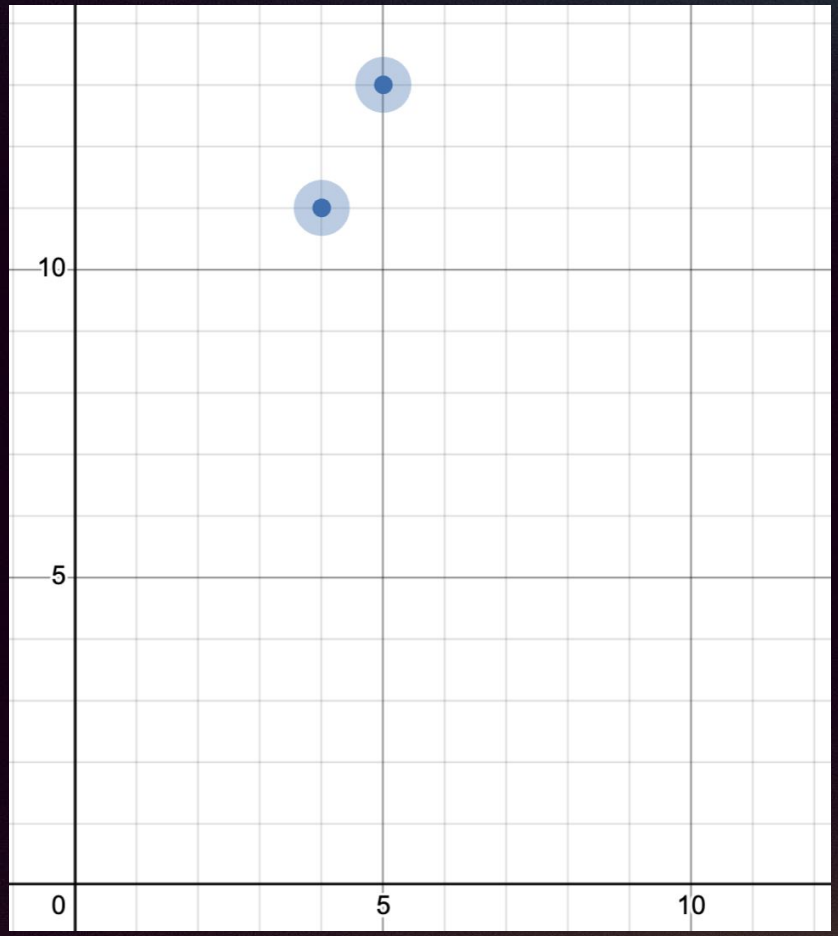
Discreet Delegation

[AFK87]

f(0110)?



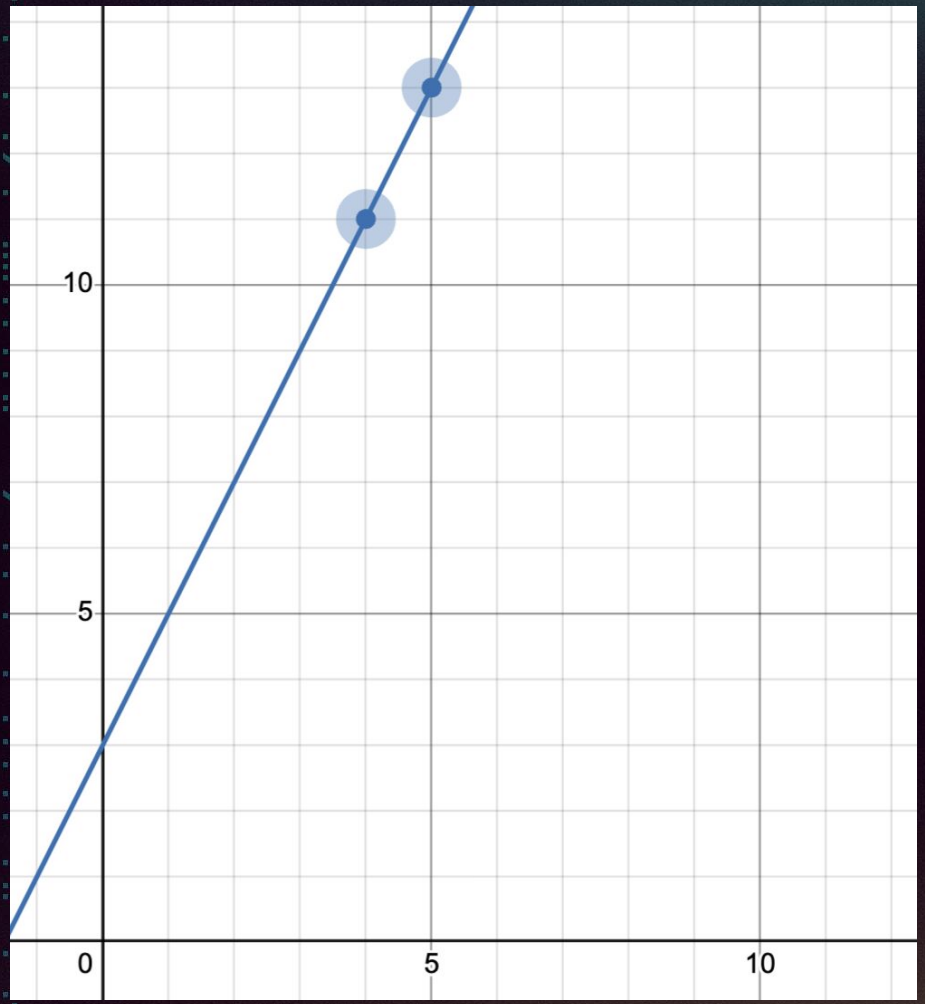
=5



Delegation to find f(1)

Hey Server, tell me F(4)?

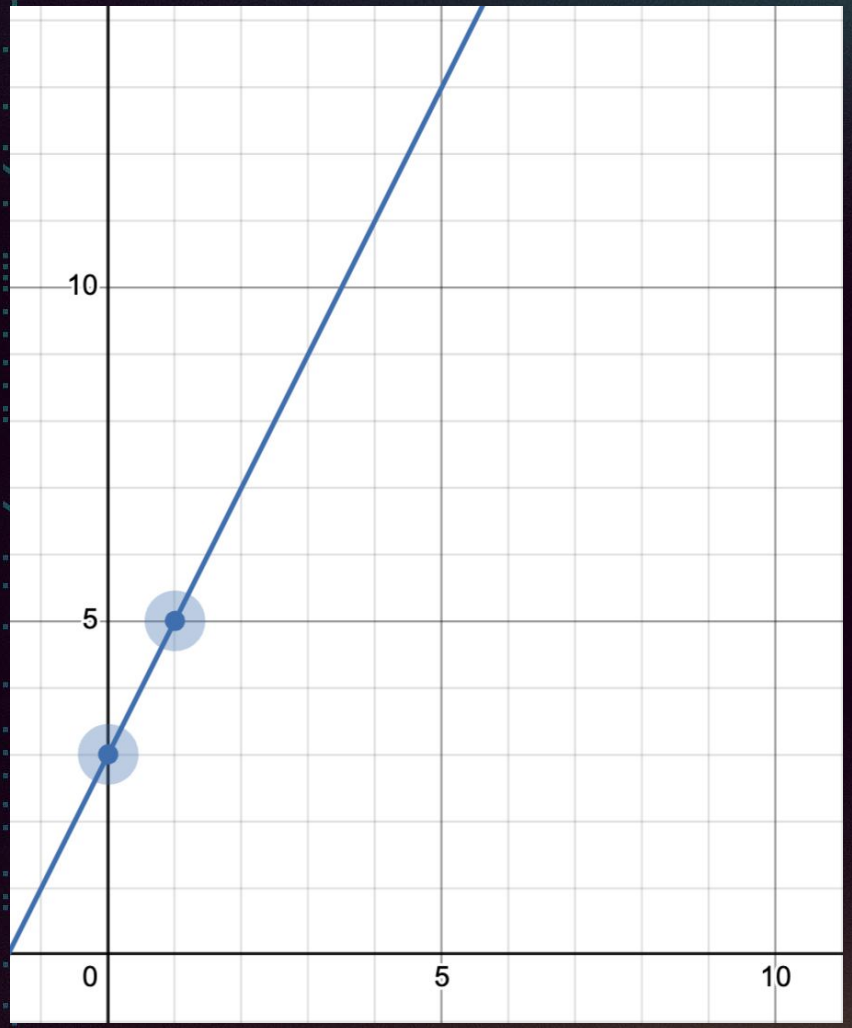
Hey Server, tell me F(5)?



Delegation to find f(1)

Thanks, Server!

Drawing a line...



Delegation to find f(1)

Thanks, Server!

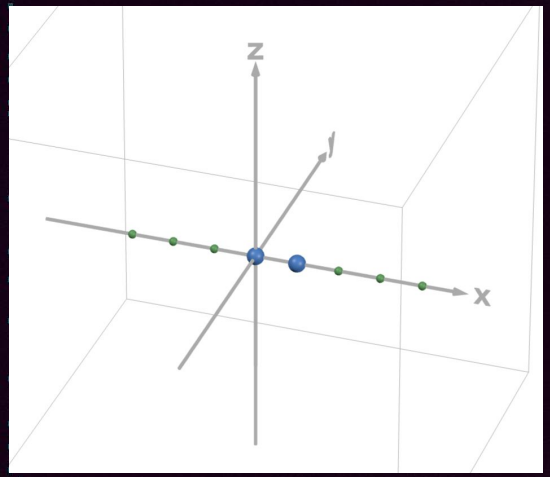
Drawing a line…

Aha, F(1) = 5.

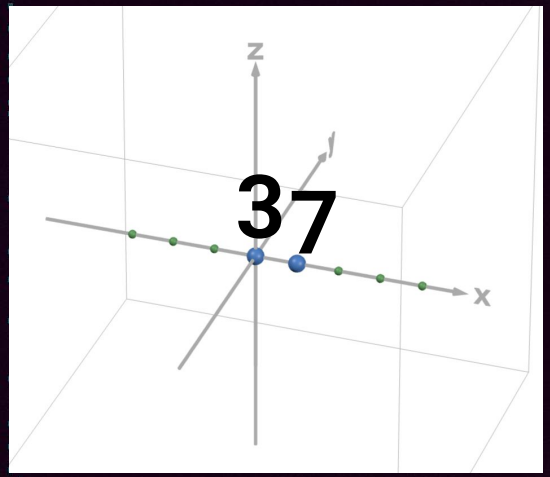
Rulers can “help”?

- Why bother asking the server?
- 2 inputs, 2 queries
- With no tricks, you gain nothing
  - naive: 1000000 inputs, 1000000 queries
- With tricks, it will get better
  - better: 1000000 inputs, 20 queries

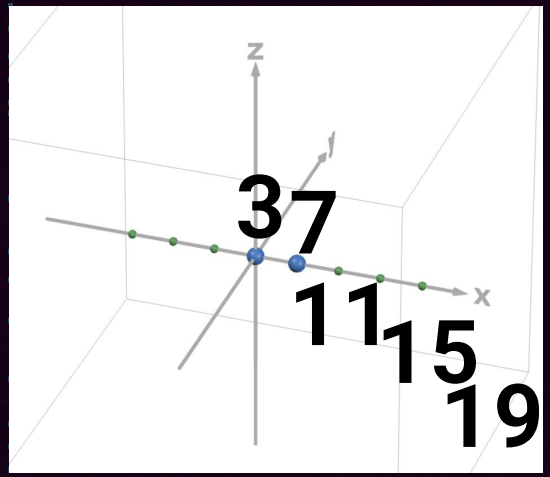
Supreme Rulers - Multilinear Extensions



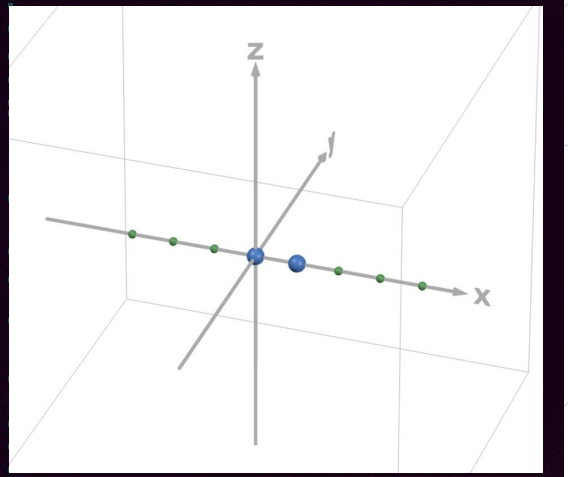
Supreme Rulers - Multilinear Extensions

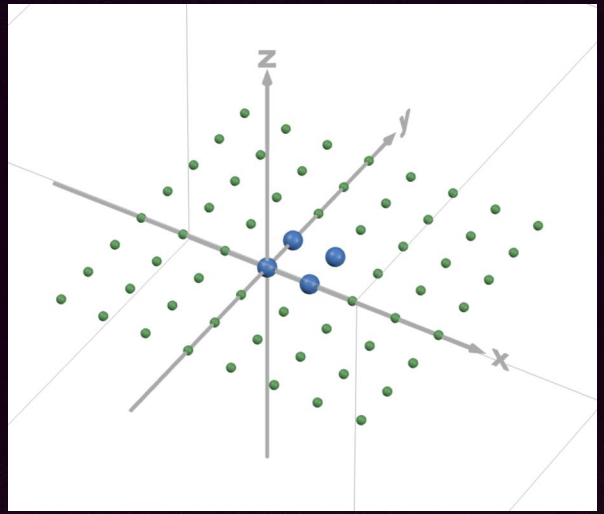


Supreme Rulers - Multilinear Extensions

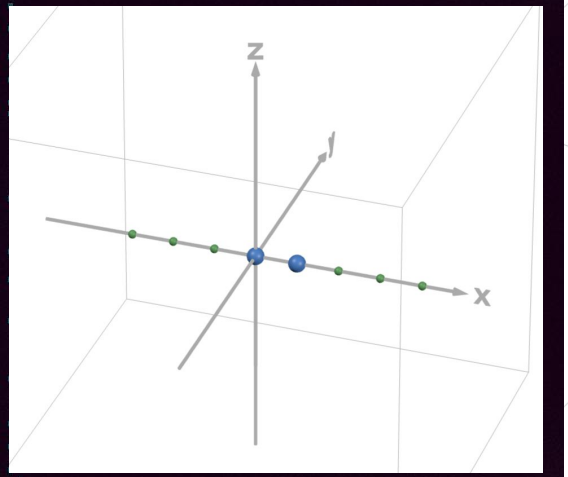


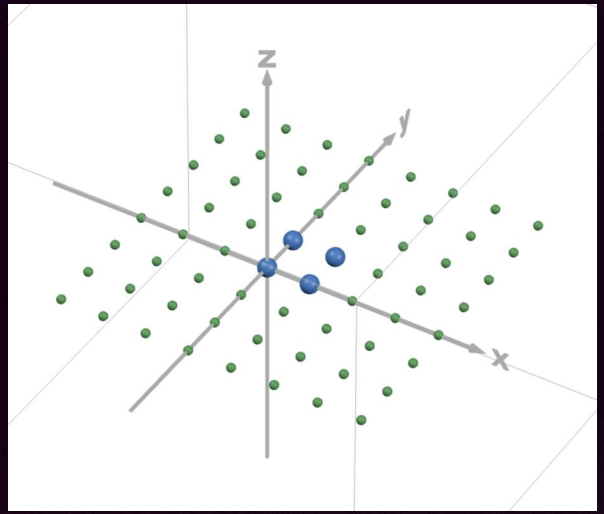
Supreme Rulers – Multilinear Extensions

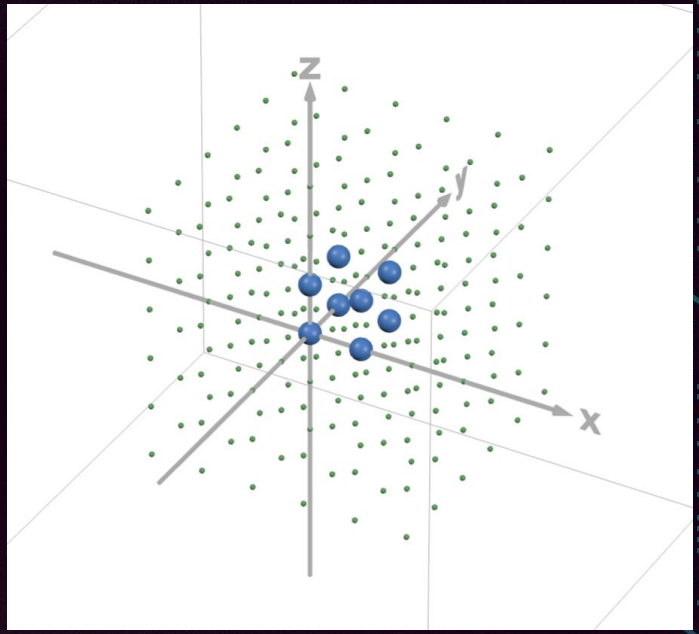




Supreme Rulers – Multilinear Extensions







Section 3

1989 - 2024

Discreet Delegation

[AFK87]

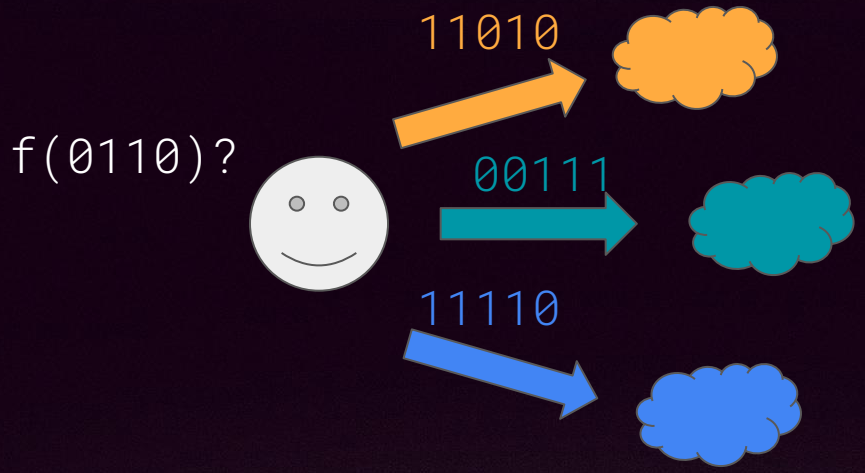
Hide x for moderate f(x)

- **(NP ∩ coNP)**

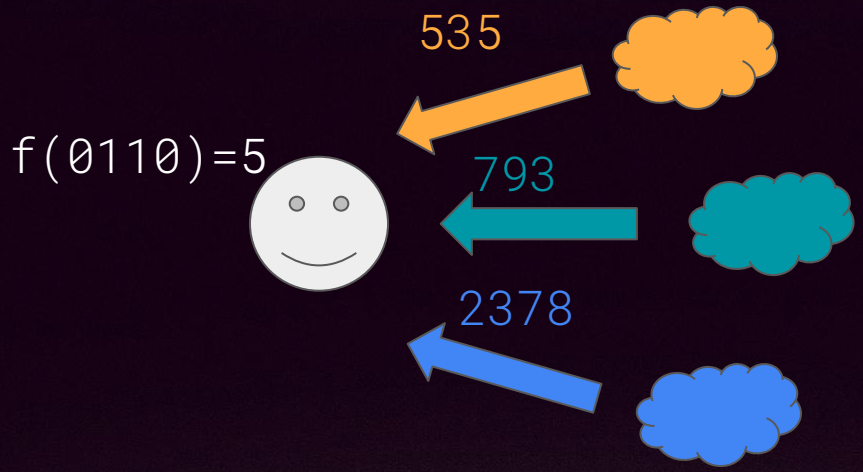
Examples:

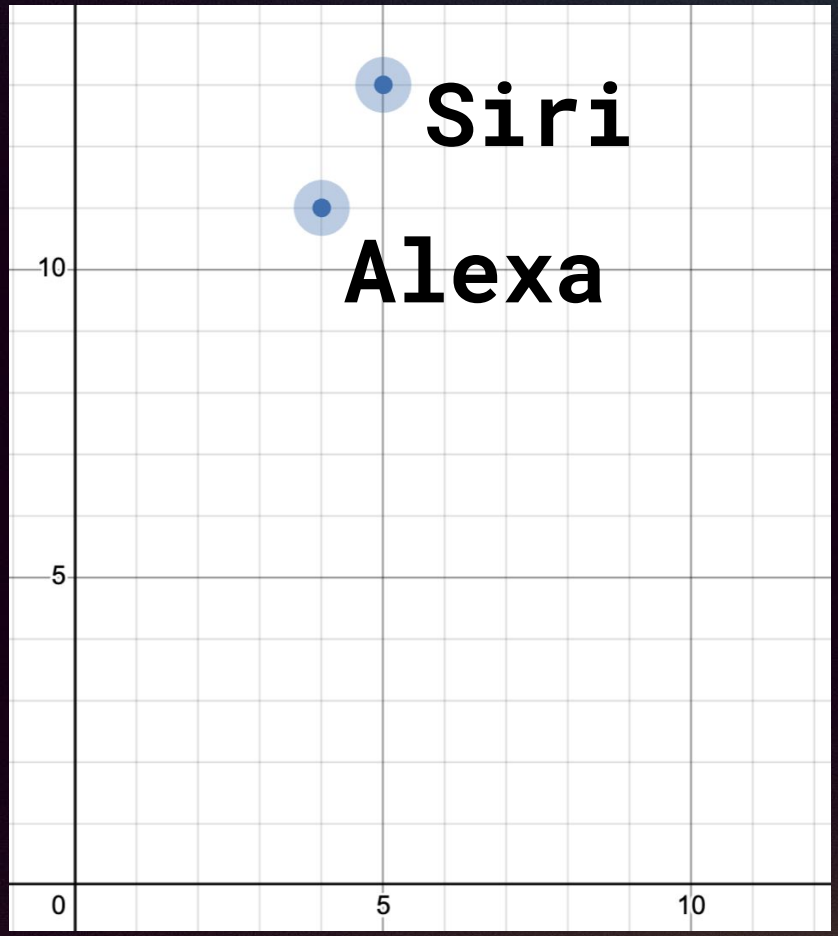
- **private inference, private off-chain calculation**

Discreet Delegation to Many



Discreet Delegation to Many

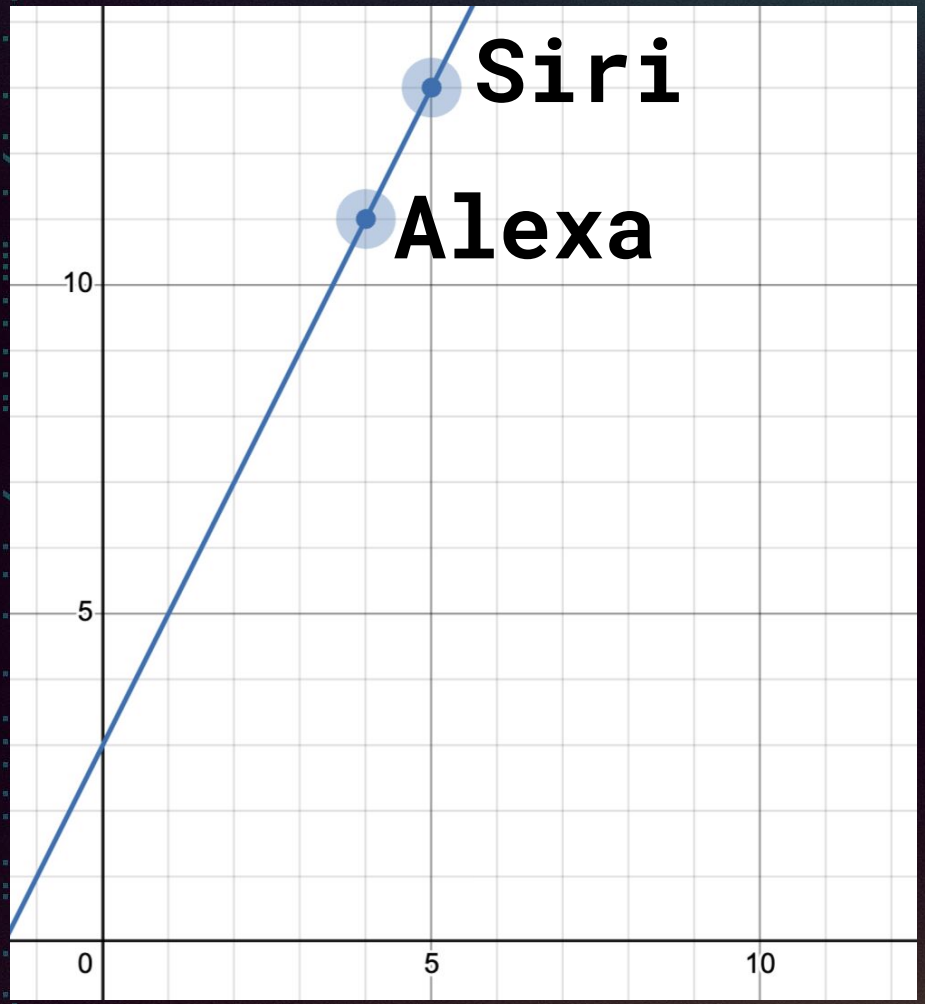




Delegation to find f(1)

Hey Alexa, tell me F(4)?

Hey Siri, tell me F(5)?

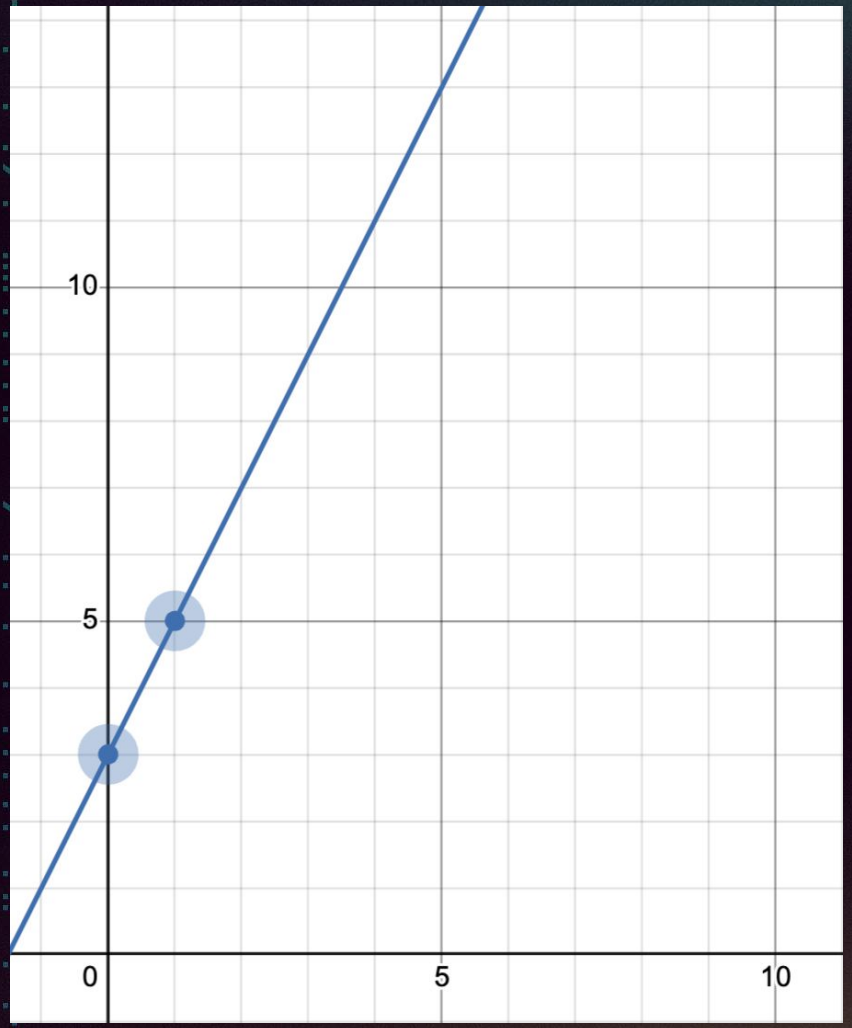


Delegation to find f(1)

Thanks, Alexa!

Thanks, Siri!

Drawing a line...



Delegation to find f(1)

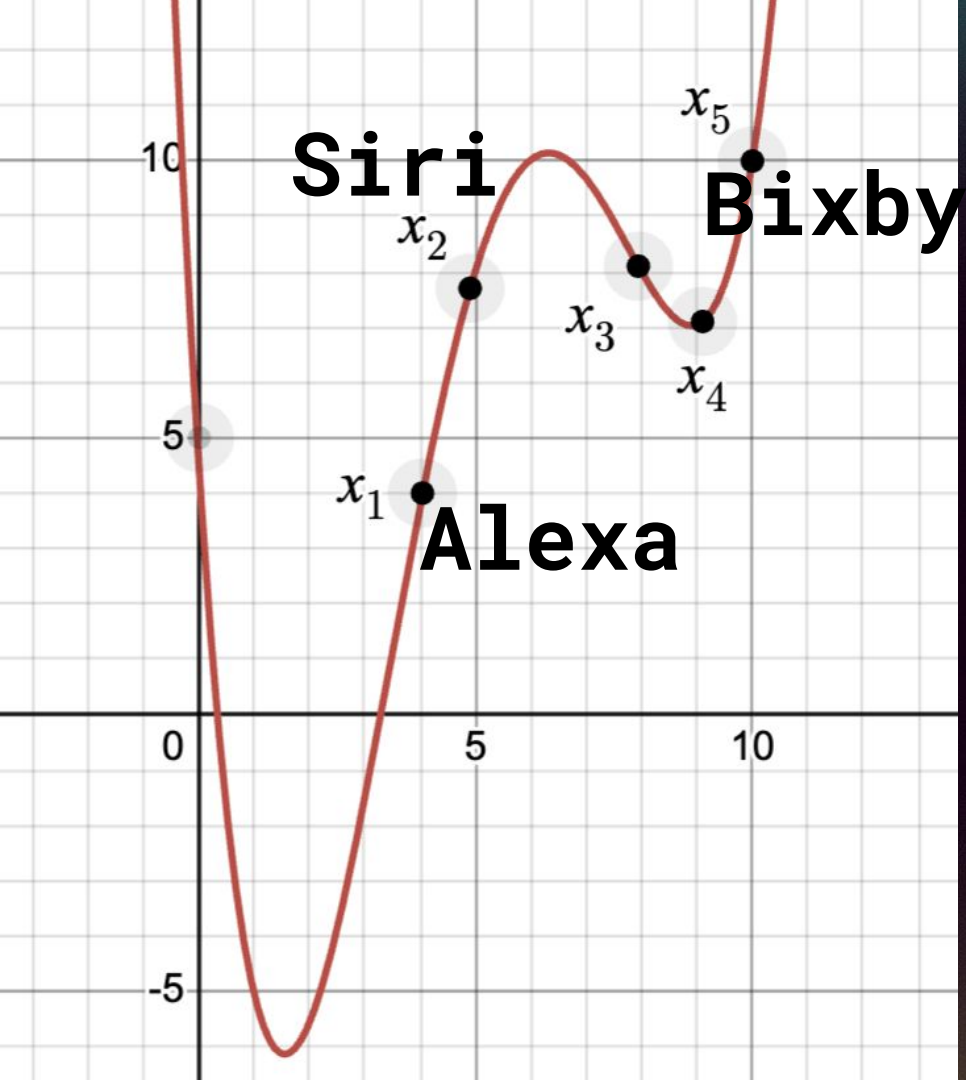
Thanks, Alexa!

Thanks, Siri!

Drawing a line…

Aha, F(1) = 5.

can ignore



Aside: secret share input bits

```
Alexa: SecSh(x), SecSh(y), SecSh(z)
Bixby: SecSh(x), SecSh(y), SecSh(z)
Siri:  SecSh(x), SecSh(y), SecSh(z)
. . . . .
```

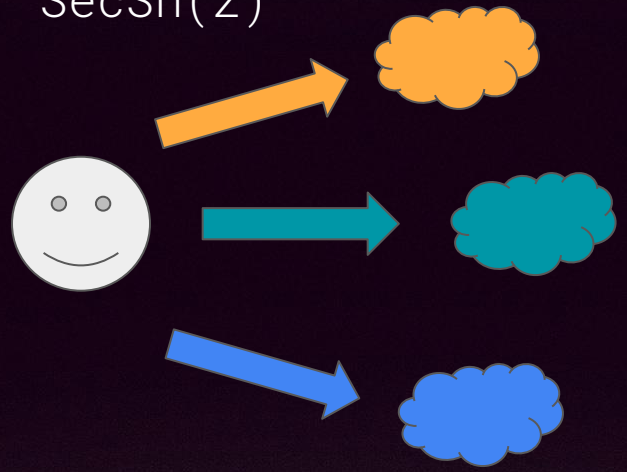
drawing a polynomial
of degree n
to learn F(boolean inputs)

back to the show

1989 (May): SumCalc

[BF89]

F(xyz)=Sum of 2^n “things”
SecSh(x), SecSh(y), SecSh(z)



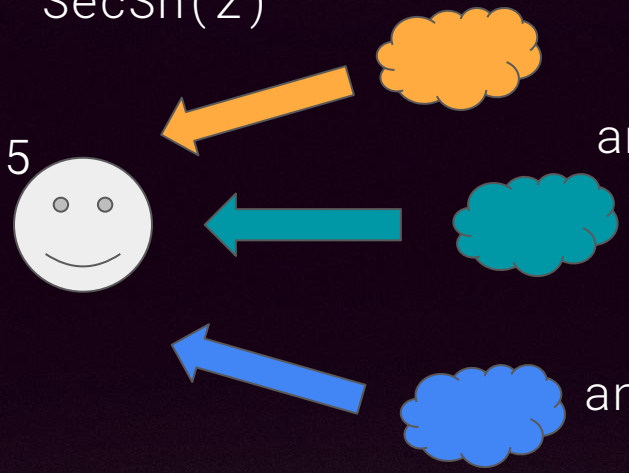
1989 (May): SumCalc

[BF89]

F(xyz)=Sum of 2^n “things”
SecSh(x), SecSh(y), SecSh(z)

ans(xyz)=Sum 2^n shares

f(0110)=5



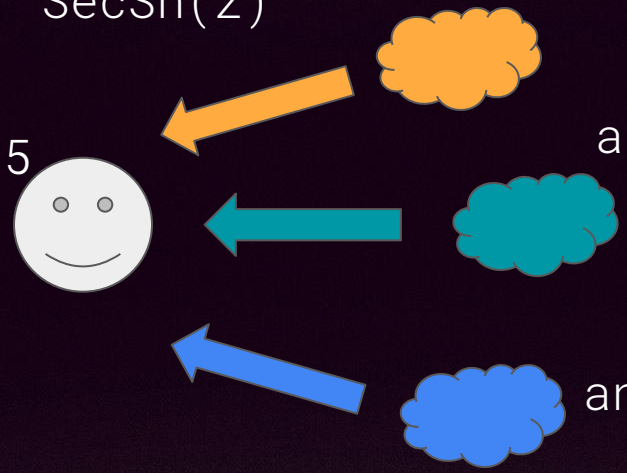
ans(xyz)=Sum 2^n shares

ans(xyz)=Sum 2^n shares

1989 (May): SumCalc

[BF89]

F(xyz)=Sum 2^n things
SecSh(x), SecSh(y), SecSh(z)



Calc polynomial of degree n=20

Calc sum of 1,000,000

MapReduce(n) of MapReduce(2^n)

1989 (May): SumCalc introduces Supreme Ruler [BF89]

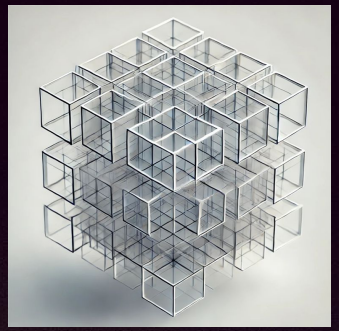
- delegate **any enormously complicated** f(x), using **tiny** communication complexity

- use a **multilinear** extension
- and a low-degree polynomial

1989 (May): SumCalc introduces Supreme Ruler [BF89]

- delegate **any enormously complicated** f(x), using **tiny** communication complexity

- use a **multilinear** extension
- and a low-degree polynomial



1989 (Summer)

Lipton

- helps for discreetly delegating Permanent / #P
  - “how many traveling salespaths?”
  - “how many sudoku solutions?”
  - way harder than NP

Nisan

- program checking: random-self-reduce #P to smaller #P

1989 (December)

1989 (December):

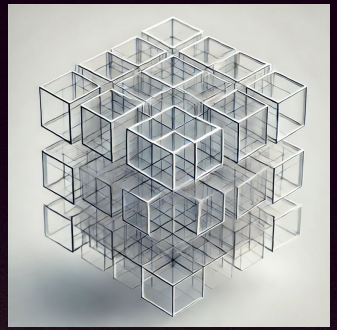
1989 (December) also: SumCheck [LFKN]

Can we **check** whether some **enormously complicated claim** f(x)=34567 is true?

Yes! When f() is in an amazingly huge complexity class (PSPACE, NEXP)

Imagine:

- solving sudoku (NP)
- counting sudoku solutions (#P)
- counting traveling salespaths (#P)
- detecting winning Go positions (PSPACE)



SumCalc and SumCheck and done

SumCalc

- Use a **multilinear** extension on a boolean hypercube
- Use a low-degree polynomial to share it
- Repeatedly interpolate lines n-times

Discreetly delegate enormously complex f(x) at low cost!

SumCheck

- Use a **multilinear** extension on a boolean hypercube
- Commit values
- Repeatedly flatten the hypercube n-times using simple queries and interpolations

Check claim for enormously complex f(x) at low cost!

Oops, no moon math

Oops, no moon math

In a Frenzy, Math Enters Age of Electronic Mail

The New York Times

By Gina Kolata

June 26, 1990

Perspectives

1989

In a Frenzy, Math Enters Age of Electronic Mail

2024

In a Frenzy, Ethereum Enters Age of Zero Knowledge

How is it that, with just **20** questions, you can check **all** the steps of a **million**-step calculation?

Section 4

Big Tables and Multilinear Extensions

Table of a 2-bit function

xy	$f(xy)$
00	5
01	9
10	2
11	17

Table of a 2-bit function - as a 2-variable multilinear

xy	$f(xy)$
00	5
01	9
10	2
11	17

$$F(xy) = \mathrm{eq}(00, xy)5 + \mathrm{eq}(01, xy)9 + \mathrm{eq}(10, xy)2 + \mathrm{eq}(11, xy)17$$

Table of a 2-bit function - as a 2-variable multilinear

$$
\begin{aligned}
F(xy) = \quad & \mathrm{eq}(00, xy)\, f(00) \\
& + \mathrm{eq}(01, xy)\, f(01) \\
& + \mathrm{eq}(10, xy)\, f(10) \\
& + \mathrm{eq}(11, xy)\, f(11)
\end{aligned}
$$

One-hot EQ helpers for 2-bit table

$$
\begin{aligned}
\mathrm{eq}(00, xy) &= (1-x)(1-y) &&= xy - x - y + 1 \\
\mathrm{eq}(01, xy) &= (1-x)y &&= -xy + y \\
\mathrm{eq}(10, xy) &= x(1-y) &&= -xy + x \\
\mathrm{eq}(11, xy) &= (1-x)y &&= -xy + 1
\end{aligned}
$$

Multilinear, with up to 4 monomials, max “degree” 2

Table of a 2-bit function - as a 2-variable multilinear

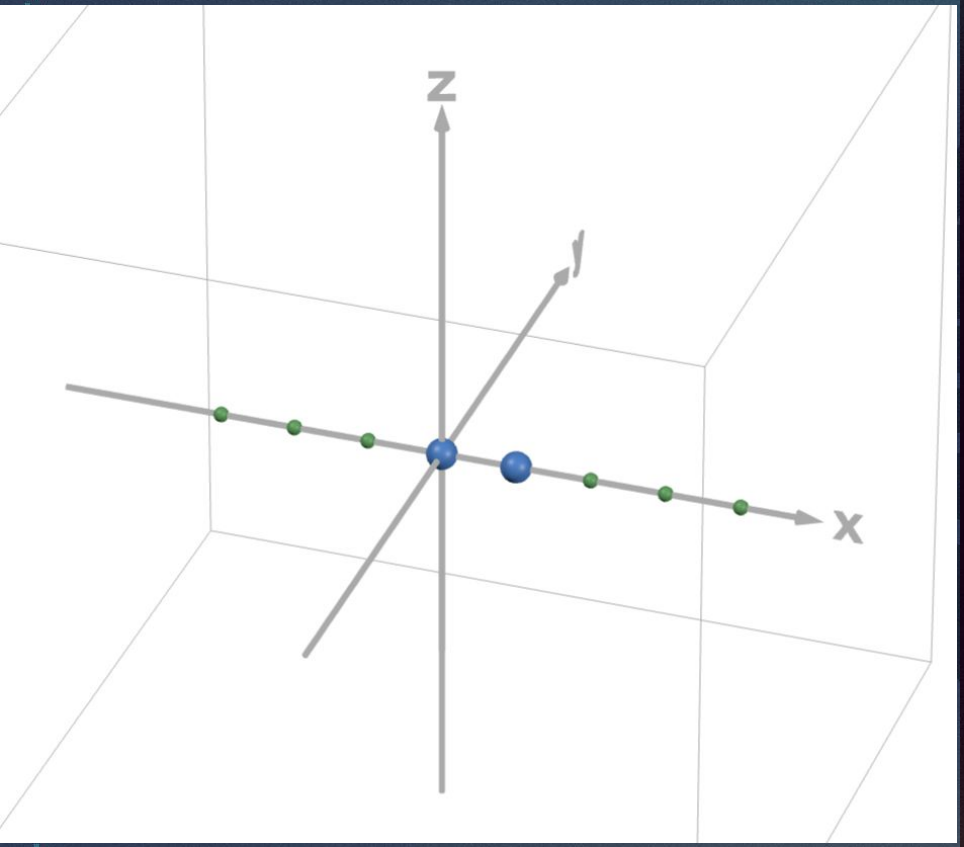
xy	$f(xy)$
00	5
01	9
10	2
11	17

$$
\begin{aligned}
F(xy) = \quad & (xy - y - y + 1)5 \\
& + (-xy + y)9 \\
& + (-xy + x)2 \\
& + (-xy + 1)17 \\
= & -23xy - 3x + 4y + 22
\end{aligned}
$$

At larger scale

$$
\begin{aligned}
F(x_1 \cdots x_n) = \quad & \mathrm{eq}(00\cdots00, x_1x_2\cdots x_{n-1}x_n) f(00..00) \\
+ \quad & \mathrm{eq}(00\cdots01, x_1x_2\cdots x_{n-1}x_n) f(00..01) \\
+ \quad & \ldots \\
+ \quad & \mathrm{eq}(11\cdots11, x_1x_2\cdots x_{n-1}x_n) f(11..11)
\end{aligned}
$$

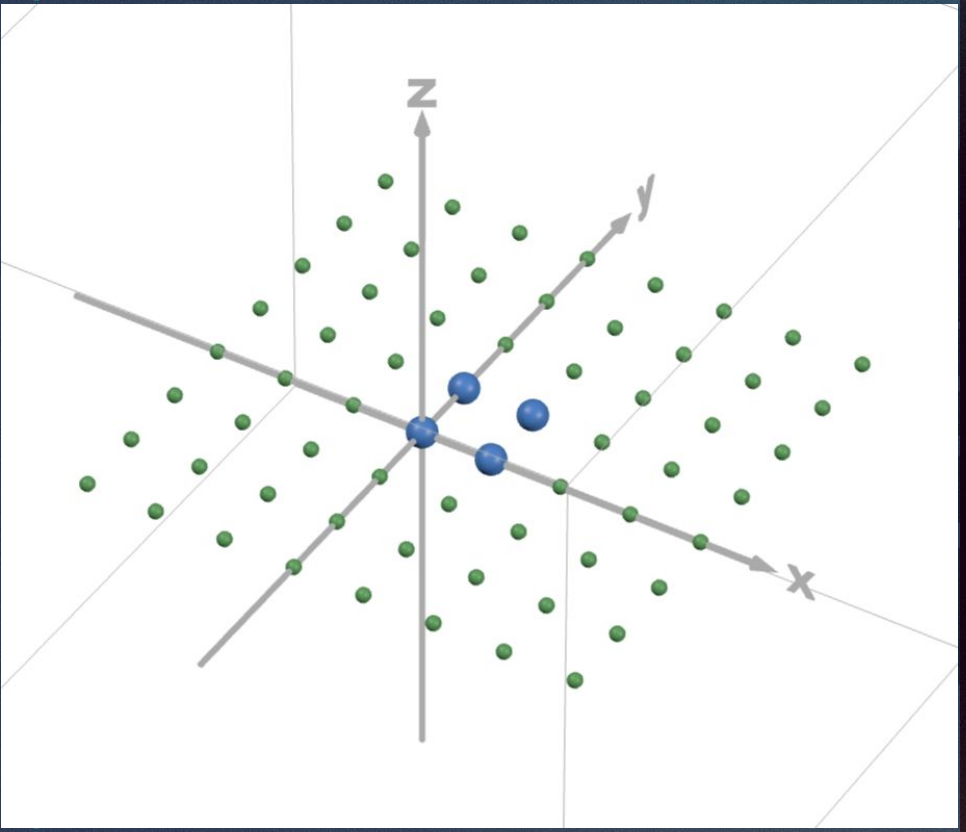
Multilinear, with up to 2^n monomials, max “degree” n



1-input MLE

2 values

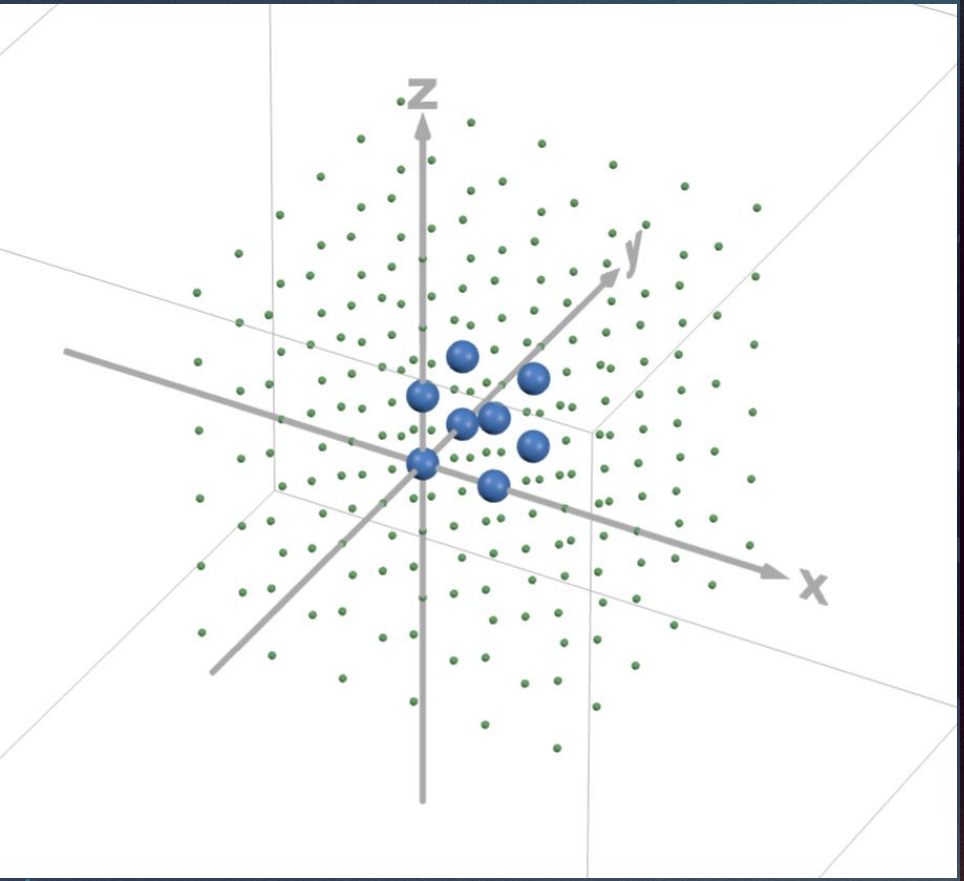
1+1 probes



2-input MLE

4 values

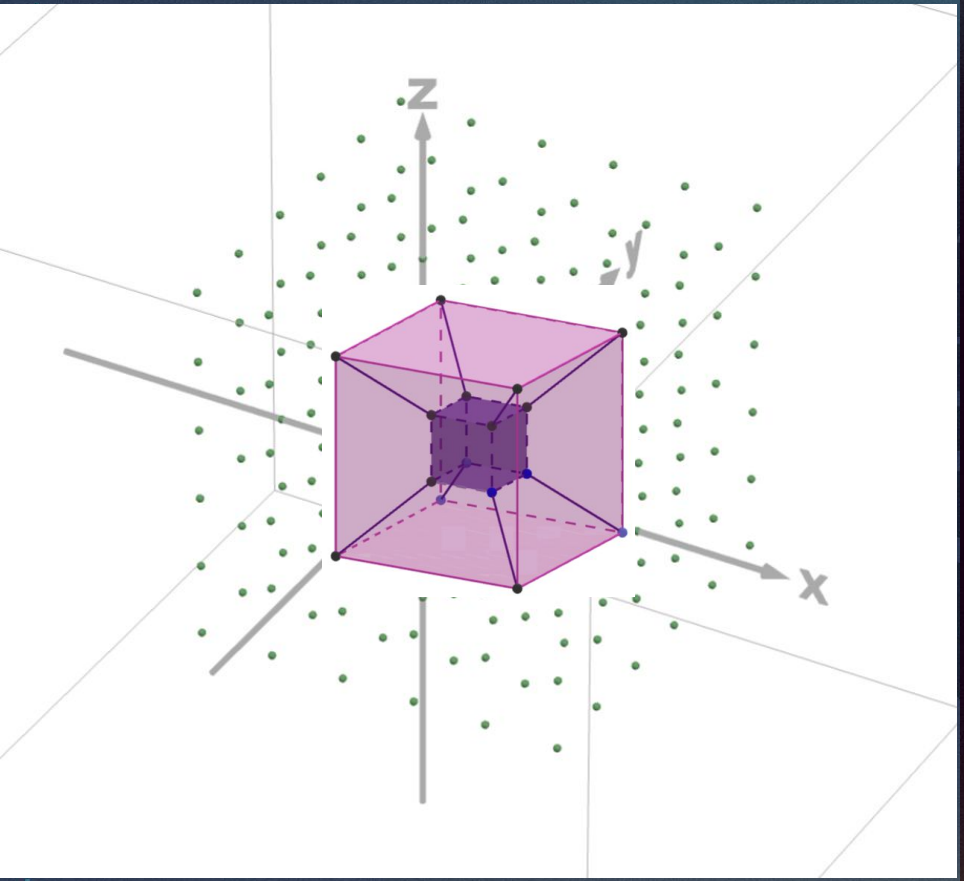
2+1 probes



3-input MLE

8 values

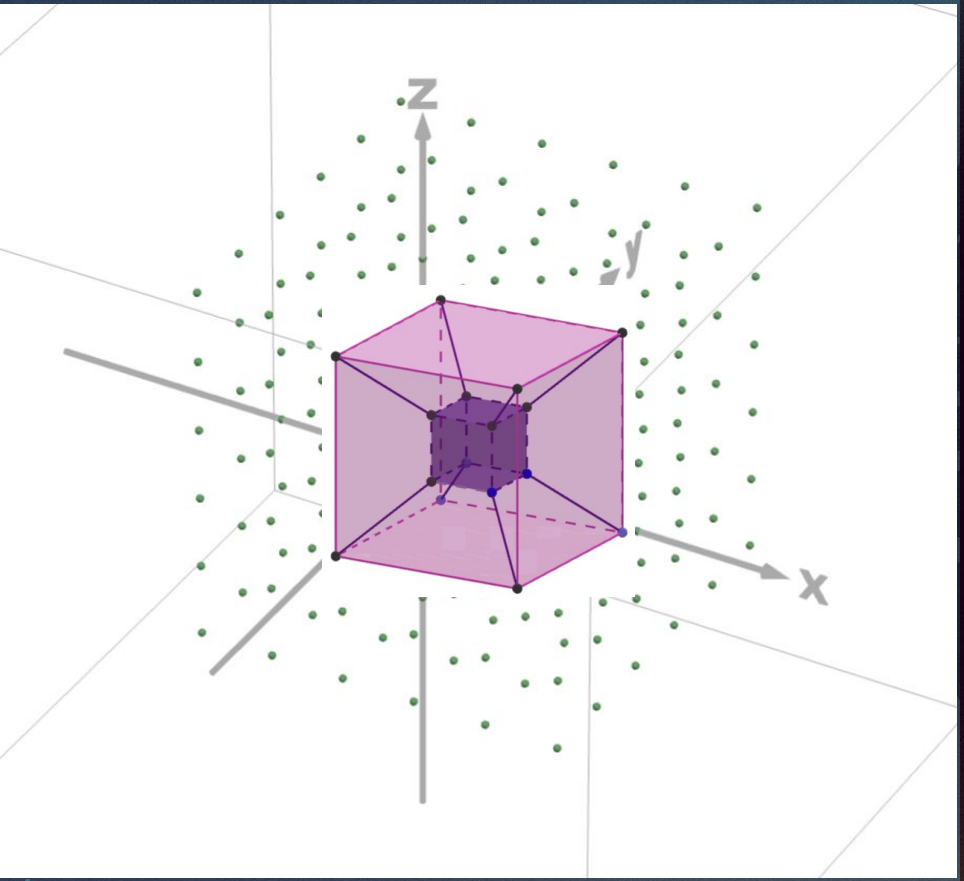
3+1 probes



n-input MLE

2^n values

n+1 probes



20-input MLE

1000000 values

20+1 probes

ZKVM

Run a program for a million steps.

Record the steps in a million-entry table, f(t).

Build the MLE on a 20-dimensional hypercube.

Moon and beyond

Elliptic curves

Commitments

Polynomial commitment schemes

FRI testing

Circuits

Algebraic representations

GKR, Plonky, Jolt, . . .

Binius, Goldilocks, Jabber, BabyBear . . .

Lots of LaTeX . . .

Section 5

Wrapping up

History of Science

1985-88

Interactive proof systems
Zero Knowledge

Discreet delegation

1989

Supreme Ruler
MLE boolean cube

Discreet n-delegation
SumCalc

IP=PSPACE
SumCheck

Swift sound computing

2013

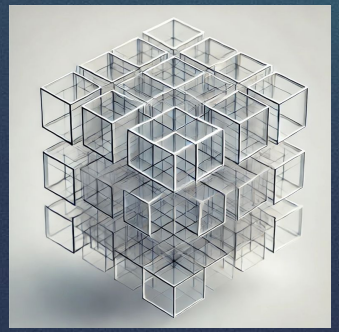
Ethereum

SNARKs, PCD

2024

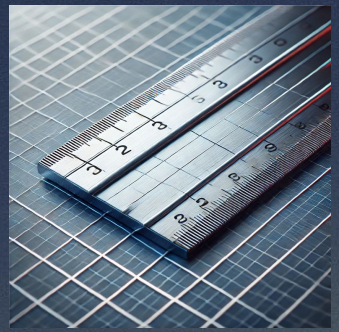
Industrial ZK

Supreme Ruler:
MLE
SumCheck-LFKN
GKR
Jolt
Lookup Singularity



Supreme Ruler

Ruler



2

million for 20

using the supreme ruler

Supreme Ruler of the World

Don Beaver

Researcher, Builder
@dcntrlzr
dcntrlzr@gmail.com